TERM PAPER REPORT

ON

Scraping Relevant Images from Web Pages Without Download

Submitted in partial fulfilment of requirements to

CS 363 - Term Paper

BY

Batch No. 22

D. Uday Sriram (Y21CS031)
D. Girish Sai (Y21CS030)
D. Lokesh Kumar (Y21CS028)

JANUARY 2024

R.V.R. & J.C. COLLEGE OF ENGINEERING (Autonomous)
(NAAC 'A+' Grade)

(Approved by **AICTE**, Affiliated to Acharya Nagarjuna University)

Chandramoulipuram::Chowdavaram,

GUNTUR – 522 019

R.V. R & J.C. COLLEGE OF ENGINEERING (Autonomous)

DEPARTMENT OF COMPUTER SCIENCE and ENGINEERING

CERTIFICATE

This is to certify that this Term Paper Report titled "**Scraping Relevant Images from Web Pages Without Download**" is the study conducted by **Dutta Uday Sriram (Y21CS031), Dhulipala Girish Sai (Y21CS030), Dasari Lokesh Kumar (Y21CS028)** and submitted in partial fulfilment of the requirements to CS 363 - Term Paper during the Academic Year 2023-2024.

Mr.S.Karthik	**Dr.Ch.Aparna**	**Dr.M.Sreelatha**
Term Paper Guide	Term Paper In-charge	Prof. &Head, CSE

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without a proper suggestions, guidance and support. Combination of these factors acts like backbone to our TERM PAPER titled "**Scraping Relevant Images from Web Pages Without Download**".

We are deeply grateful to our guide, **Mr.S.Karthik**, for his unwavering guidance, insightful feedback, and encouragement. His expertise and dedication have been instrumental in shaping the direction and quality of this research.

We would like to express our sincere gratitude to **Dr.Ch. Aparna**, In-charge for our term paper. Her expertise, guidance, and support were instrumental in the successful completion of this research.

We express our sincere thanks to **Dr.M. Sreelatha**, Head of the Department of Computer Science and Engineering for her encouragement, support, commitment to enhance research experience.

We are very much thankful to **Dr. Kolla Srinivas**, Principal of R.V.R &J.C College of Engineering, Guntur for providing this supportive Environment and to engage in research activities.

Finally, we submit our reserves thanks to lab staff in Department of Computer Science and Engineering for their cooperation, support, for providing administrative support and technical assistance during selection.

D.Uday Sriram (Y21CS031)
D.Girish Sai (Y21CS030)
D.Lokesh Kumar (Y21CS028)

ABSTRACT

Our method makes web image extraction simpler by finding the right balance between speed and accuracy, all without needing experts to step in. We group similar web pages together and then guide non-experts in selecting the relevant images. Rather than relying on extensive image datasets for training, we use textual data to educate our system. Through rigorous testing on 200 news websites and a vast corpus of over 600,000 images, our approach consistently outperformed existing automatic methods, boasting an impressive average f-Measure of 0.958. Remarkably, this level of performance was achieved with just six annotated pages per website. Thus, for 200 websites, only 1,200 pages need to be examined to identify the relevant images. Additionally, by sidestepping the need for image downloads and leveraging textual data, our approach not only saves time and storage resources but also seamlessly integrates with existing web scraping tools. Overall, our method offers a streamlined solution for web image extraction, combining efficiency, accuracy, and ease of use for non-experts.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBRIVATIONS

1.TC	-	Text-based Classification
2.DOM	-	Document object model
3.SVM	-	Support Vector Machines
4.DBSCAN	-	Density-based spatial clustering of applications with noise
5.LCS	-	Longest Common Subsequence

1. INTRODUCTION

1.1 Background

Web image extraction poses significant challenges in data retrieval, especially when considering the vast and dynamic nature of online content. Traditional methods often rely on manual preparation of extraction patterns or extensive training datasets, which can be error-prone and time-consuming. Automatic approaches, while promising, often require large training datasets and numerous features, making them difficult to implement without specialized knowledge.

1.2 Problem Statement

The challenge lies in effectively extracting relevant images from diverse web page layouts while differentiating them from irrelevant images like advertisements and headers. This task is complicated by the varying sizes and placements of images within different layouts, as well as the need to navigate the hierarchical structure of HTML elements to access them. Manual extraction methods rely on expert analysis to develop extraction patterns, a laborious process prone to errors and requiring extensive pattern testing across multiple web pages. Automated extraction approaches offer potential solutions but must contend with the complexities of accurately identifying relevant images amidst the multitude of elements on a web page.

1.3 Objectives

The proposed research aims to develop a semi-automatic approach for scraping relevant images from web pages, with a primary focus on leveraging textual data to enhance performance in terms of accuracy and computation time. By integrating advanced techniques, this approach seeks to significantly reduce reliance on manual extraction methods commonly employed in existing web scraping tools. The goal is to streamline the extraction process by minimizing the need for expert intervention, thus making it accessible to a broader range of users. Central to this approach is the construction of accurate learning models tailored to website-specific image extraction tasks, achieved with minimal features and training data.

1.4 Limitations of the Existing Techniques

- **Expert Dependency in Manual Approaches:**

  Manual extraction methods heavily rely on expert users to prepare extraction patterns, such as regular expressions, CSS selectors, or XPath expressions. This reliance introduces a bottleneck in the extraction process, as preparing these patterns is a monotonous and error-prone task, leading to potential inaccuracies and inefficiencies.

- **Challenges in Automatic Approaches:**

  While automatic extraction approaches aim to eliminate the need for expert intervention, they still face limitations in effectively scraping desired data from web pages. Unsupervised methods, relying on tree-based techniques like tree edit distance matching, are strictly tied to website structure, limiting their applicability. Supervised approaches, though more flexible, require well-prepared training datasets tailored to specific domains, posing challenges in dataset creation and feature selection.

- **Limited Focus on Image Prediction:**

  Existing studies on data extraction primarily focus on text-based data, with relatively few addressing the prediction of relevant images. Unsupervised methods for image prediction often rely on heuristic measures or simple techniques based on image size, aspect ratio, and tags, leading to suboptimal performance. While supervised approaches show promise in improving accuracy, they require extensive parameter tuning and feature engineering to achieve satisfactory results.

- **Lack of Integration:**

  Existing techniques often operate in isolation, lacking seamless integration with web scraping tools or broader data extraction frameworks. This hinders their practical utility and adoption in real-world applications where streamlined data extraction pipelines are essential.

2. Literature Review

- **N Aslam, B Tahir, HM Shafiq, MA Mehmood:**

  N. Aslam, B. Tahir, H.M. Shafiq, and M.A. Mehmood are prominent researchers in the field of web data processing and analysis. Their collective expertise spans various aspects of information retrieval, data mining, and algorithm development. In their latest research endeavor, they address the ubiquitous challenge of noise within web pages, a hindrance to effective data mining and information retrieval tasks. Building upon existing algorithms such as Boilerpipe and JustText, the authors introduce a novel approach named Web-AM to tackle this issue. By extending the capabilities of Boilerpipe and leveraging the HTML tree structure, Web-AM aims to detect and remove noise from web articles, enhancing the precision of content extraction processes. Through rigorous evaluation on benchmark datasets and the creation of their own corpus, the authors demonstrate the efficacy of Web-AM in significantly improving precision compared to existing methods.

  Reference: Web-AM: An efficient boilerplate removal algorithm for Web articles

- **HV Agun, E Uzun:**

  Luo HV Agun and E. Uzun introduce a novel approach in their study to address the challenges associated with automatically extracting relevant images from web pages. Traditional methods for this task are error-prone and time-consuming, often requiring extensive manual effort. To overcome these limitations, the authors propose a fully-automated approach based on the alignment of regular expressions, marking the first application of this technique to image extraction. The approach involves a multi-stage inference process to generate regular expressions from attribute values of relevant and irrelevant image elements on web pages, reducing complexity through the application of a constraint on the Levenshtein distance algorithm

  Reference: An efficient regular expression inference approach for relevant image extraction

- **Ziv Bar-Yossef, Sridhar Rajagopalan:**

  The formulation and proposition of the template detection problem signify a critical step towards understanding and addressing the pervasive use of templates across the web. Through our research, we have identified templates as fundamental structures that underpin a significant portion of online content. Our proposed solution, rooted in the methodology of counting frequent item sets, offers a practical approach to tackle this challenge.We observe that templates, while serving as efficient means for structuring information, often pose challenges to hypertext information retrieval (IR) and data mining (DM) systems. These challenges stem from the inherent violation of three key principles governing these systems. Firstly, templates introduce a level of uniformity that can mislead traditional search algorithms, undermining their ability to accurately retrieve relevant information. Secondly, the rigid structure imposed by templates inhibits the exploration of diverse content, limiting the effectiveness of data mining techniques in uncovering valuable insights. Lastly, templates can obscure the natural variability of content, leading to a distortion of the underlying data distribution and impeding the precision-recall trade-off.

  Reference: Template detection via data mining and its applications

- **Rodrigo Barbado, Carlos A. Iglesias:**

  The impact of online reviews on businesses has grown significantly during last years, being crucial to determine business success in a wide array of sectors, ranging from restaurants, hotels to e-commerce. Unfortunately, some users use unethical means to improve their online reputation by writing fake reviews of their businesses or competitors. Previous research has addressed fake review detection in a number of domains, such as product or business reviews in restaurants and hotels. However, in spite of its economical interest, the domain of consumer electronics businesses has not yet been thoroughly studied. This article proposes a feature framework for detecting fake reviews that has been evaluated in the consumer electronics domain. The contributions are fourfold: (i) Construction of a dataset for classifying fake reviews in the consumer electronics domain in four different cities based on scraping techniques; (ii) definition of a feature framework for fake review detection; (iii) development of a

fake review classification method based on the proposed framework and (iv) evaluation and analysis of the results for each of the cities under study. We have reached an 82% F-Score on the classification task and the Ada Boost classifier has been proven to be the best one by statistical means according to the Friedman test.

Reference: A framework for fake review detection in online consumer electronics retailers

- **Aanshi Bhardwaj, Veenu Mangat:**

World Wide Web (WWW) is now a famous medium by which people all around the world can spread and gather information of all kind. However, there is large amount of irrelevant redundant and information on web pages also. Such information makes various web mining tasks web page crawling, web page classification, link based ranking and topic distillation complex. Previously, the relevant content was extracted only from textual part of web pages. But now-a-days the content on web page is not only in the text form but also as an image, video or audio. This paper proposes an improved algorithm for extracting informative content from web pages i.e. it extracts the relevant content not only as text but also as images, videos, audios, adobe flash files and online games. Experiments were conducted on different real websites show that precision and recall values of our approach is superior to the previous Word to Leaf Ratio approach. In response to this evolving landscape, this paper introduces an improved algorithm designed to extract informative content from web pages in a manner that encompasses diverse media types beyond text alone. By extending the scope of content extraction to encompass images, videos, audios, and other multimedia elements, our approach aims to provide a more holistic representation of the information available on web pages.

Reference: An Improvised Algorithm for Relevant Content Extraction from Web Pages

- **Lidong Bing, Tak-Lam Wong, and Wai Lam:**

  We develop an unsupervised learning framework for extracting popular product attributes from product description pages originated from different E-commerce Web sites. Unlike existing information extraction methods that do not consider the popularity of product attributes, our proposed framework is able to not only detect popular product features from a collection of customer reviews but also map these popular features to the related product attributes. One novelty of our framework is that it can bridge the vocabulary gap between the text in product description pages and the text in customer reviews. Technically, we develop a discriminative graphical model based on hidden Conditional Random Fields. As an unsupervised model, our framework can be easily applied to a variety of new domains and Web sites without the need of labeling training samples. Extensive experiments have been conducted to demonstrate the effectiveness and robustness of our framework.

  Reference: Unsupervised Extraction of Popular Product Attributes from E-Commerce Web Sites by Considering Customer Reviews

- **Fadwa Estuka, James Miller:**

  Database-driven websites and the amount of data stored in their databases are growing enormously. Web databases retrieve relevant information in response to users' queries; the retrieved information is encoded in dynamically generated web pages as structured data records. Identifying and extracting retrieved data records is a fundamental task for many applications, such as competitive intelligence and comparison shopping. This task is challenging due to the complex underlying structure of such web pages and the existence of irrelevant information. Numerous approaches have been introduced to address this problem, but most of them are HTML-dependent solutions that may no longer be functional with the continuous development of HTML. Although a few vision-based techniques have been introduced, various issues exist that inhibit their performance. To overcome this, we propose a novel visual approach, i.e., programming-language-independent, for automatically extracting structured web data. The proposed approach makes full use of the natural human tendency of visual object perception and the Gestalt laws of grouping.

Reference: A Pure Visual Approach for Automatically Extracting and Aligning Structured Web Data

- **Nancy Fazal, Khue Nguyen, and Pasi Fränti:**

The purpose of this study was to find the efficiency of a web crawler for finding geotagged photos on the internet. We consider two alternatives: (1) extracting geolocation directly from the metadata of the image, and (2) geo-parsing the location from the content of the web page, which contains an image. We compare the performance of simple depth-first, breadth-first search, and a selective search using a simple guiding heuristic. The selective search starts from a given seed web page and then chooses the next link to visit based on relevance calculation of all the available links to the web pages they contain in. Our experiments show that the crawling will find images all over the world, but the results are rather sparse. Only a fraction of 6845 retrieved images (<0.1%) contained geotag, and among them only 5 percent were able to be attached to geolocation.

Reference: Efficiency of Web Crawling for Geotagged Image Retrieval

- **Leandro Neiva Lopes Figueiredo, Guilherme Tavares de Assis, and Anderson A. Ferreira:**

Extracting data from web pages is an important task for several applications such as comparison shopping and data mining. Ordinarily, the data in web pages represent records from a database and are obtained using a web search. One of the most important steps for extracting records from a web page is identifying out of the different data regions, the one containing the records to be extracted. An incorrect identification of this region may lead to an extraction of incorrect records. This process is followed by the equally important step of detecting and correctly splitting the necessary records and their attributes from the main data region. In this study, we propose a method for data extraction based on rendering information and an n-gram model (DERIN) that aims to improve wrapper performance by automatically selecting the main data region from a search results page and extracting its records and attributes based on rendering information. The proposed DERIN method can detect

different record structures using techniques based on an n-gram model. Moreover, DERIN does not require examples to learn how to extract the data, performs a given domain independently and can detect records that are not children of the same parent element in the DOM tree. Experimental results using web pages from several domains show that DERIN is highly effective and performs well when compared with other methods.

Reference: A data extraction method based on rendering information and n-gram

- **Najlah Gali, Andrei Tabarcea, and Pasi Fränti:**

  A web page typically contains a blend of information. For a particular user, only informative data such as main content and representative images are considered useful, while non-informative data such as advertisements and navigational banners are not. In this work, we focus on selecting a representative image that would best represent the content of a web page. Existing techniques rely on prior knowledge of website specific templates and on text body. We extract all images, analyze and rank them according to their features and functionality in the web page. We select the highest scored image as the representative image. Our method is fully automated, template independent, and not limited to a certain type of web pages.

  Reference: Extracting Representative Image from Web Page

- **Waqar Haider, Yeliz Yesilada:**

  Table mining on the web is an open problem, and none of the previously proposed techniques provides a complete solution. Most research focuses on the structure of the HTML document, but because of the nature and structure of the web, it is still a challenging problem to detect relational tables. Web Content Accessibility Guidelines (WCAG) also cover a wide range of recommendations for making tables accessible, but our previous work shows that these recommendations are also not followed; therefore, tables are still inaccessible to disabled people and automated processing. We propose a new approach to table mining by not looking at the HTML structure, but rather, the rendered pages by the browser. The first task in table mining on the web is to classify relational vs. layout tables, and here, we propose two alternative

approaches for that task. We first introduce our dataset, which includes 725 web pages with 9,957 extracted tables. Our first approach extracts features from a page after being rendered by the browser, then applies several machine learning algorithms in classifying the layout vs. relational tables.

Reference: Classification of Layout vs. Relational Tables on the Web: Machine Learning with Rendered Pages

- **Wook-Shin Han, Wooseong Kwak, Hwanjo Yu, Jeong-Hoon Lee:**

  Extracting tuples from HTML pages has been an important issue in various web applications. Commercial tuple extraction systems have enjoyed some success to extract tuples by regarding HTML pages as tree structures and exploiting XPath queries to find attributes of tuples in the HTML pages. However, such systems would be vulnerable to small changes on the web pages. In this paper, we propose a robust tuple extraction system which utilizes spatial relationships among elements rather than the XPath queries. Spatial information (e.g., 2-D coordinates) of elements are maintained in the DOM tree when a web page is rendered in a browser. Our system regards elements in the rendered page as spatial objects in the 2-D space and executes spatial joins to extract target elements. Since humans also identify an element in a web page by its relative spatial location, our system extracting elements by their spatial relationships could possibly be as robust as manual extraction. To specify and execute spatial joins, we propose a new query language, RAQuery, based on topological relationships between any spatial objects in the 2-D space. We then propose spatial join algorithms that efficiently process the RAQuery using novel notions of group match and prunable relation group. We next propose a tuple construction algorithm to build tuples from the extracted elements obtained by the spatial joins, which can construct tuples even when there are no boundary HTML elements specified for the tuples in the web page.

  Reference: Leveraging spatial join for robust tuple extraction from web pages

- **Jonathan I. Helfman, James D. Hollan:**

  The web is enormous and constantly growing. User-interfaces for web-based applications need to make it easy for people to access relevant information without becoming overwhelmed or disoriented. Today's interfaces employ textual representations almost exclusively, typically organized in lists and hierarchies of web-page titles or URL taxonomies. Given the ability of images to assist memory and our frequent exploitation of space in everyday problem solving to simplify choice, perception, and mental computation, it is surprising that so little use is made of images and spatial organizations in accessing and organizing web information. The work we summarize in this paper suggest that spatial and temporal organization of selectable images may offer multiple advantages over textual lists of titles and URLs. We describe several image-based applications, detail basic image representation techniques, and discuss spatial and temporal strategies for organization.

  Reference: Image representations for accessing and organizing Web information

- **Chun-Nan Hsu, Ming-Tzung Dung:**

  Integrating a large number of Web information sources may significantly increase the utility of the World-Wide Web. A promising solution to the integration is through the use of a Web Information mediator that provides seamless, transparent access for the clients. Information mediators need wrappers to access a Web source as a structured database, but building wrappers by hand is impractical. Previous work on wrapper induction is too restrictive to handle a large number of Web pages that contain tuples with missing attributes, multiple values, variant attribute permutations, exceptions and typos. This paper presents SoftMealy, a novel wrapper representation formalism. This representation is based on a finite-state transducer (FST) and contextual rules. This approach can wrap a wide range of semistructured Web pages because FSTs can encode each different attribute permutation as a path

  Reference: Generating finite-state transducers for semi-structured data extraction from the Web

- **Patricia Jiménez, Juan C. Roldán, Rafael Corchuelo:**

  HTML tables have become pervasive on the Web. Extracting their data automatically is difficult because finding the relationships between their cells is not trivial due to the many different layouts, encodings, and formats available. In this article, we introduce Melva, which is an unsupervised domain-agnostic proposal to extract data from HTML tables without requiring any external knowledge bases. It relies on a clustering approach that helps make label cells apart from value cells and establish their relationships. We compared Melva to four competitors on more than 3000 HTML tables from the Wikipedia and the Dresden Web Table Corpus. The conclusion is that our proposal is 21.70% better than the best unsupervised competitor and equals the best supervised competitor regarding effectiveness, but it is 99.14% better regarding efficiency.

  Reference: A clustering approach to extract data from HTML tables

- **Yeonjung Kim, Jeahyun Park, Taehwan Kim, Joongmin Choi:**

  The main issue for effective Web information extraction is how to recognize similar patterns in a Web page. Traditionally, it has been shown that pattern matching by using the HTML DOM tree is more efficient than the simple string matching approach. Nonetheless, previous tree-based pattern matching methods have problems by assuming that all HTML tags have the same values, assigning the same weight to each node in HTML trees. This paper proposes an enhanced tree matching algorithm that improves the tree edit distance method by considering the characteristics of HTML features. We assign different values to different HTML tree nodes according to their weights for displaying the corresponding data objects in the browser. Pattern matching of HTML patterns is done by obtaining the maximum mapping values of two HTML trees that are constructed with weighted node values from HTML data objects. Experiments are done over several Web commerce sites to evaluate the effectiveness of the proposed HTML tree matching algorithm.

  Reference: Web Information Extraction by HTML Tree Edit Distance Matching

- **Christian Kohlschütter, Peter Fankhauser, Wolfgang Nejdl:**

  In addition to the actual content Web pages consist of navigational elements, templates, and advertisements. This boilerplate text typically is not related to the main content, may deteriorate search precision and thus needs to be detected properly. In this paper, we analyze a small set of shallow text features for classifying the individual text elements in a Web page. We compare the approach to complex, state-of-the-art techniques and show that competitive accuracy can be achieved, at almost no cost. Moreover, we derive a simple and plausible stochastic model for describing the boilerplate creation process. With the help of our model, we also quantify the impact of boilerplate removal to retrieval performance and show significant improvements over the baseline. Finally, we extend the principled approach by straight-forward heuristics, achieving a remarkable detection accuracy.

  Reference: Boilerplate detection using shallow text features

- **Bing Liu:**

  Covers all key tasks and techniques of Web search and Web mining, i.e., structure mining, content mining, and usage mining. Includes major algorithms from data mining, machine learning, information retrieval and text processing, which are crucial for many Web mining tasks. Contains a rich blend of theory and practice, addressing seminal research ideas and also looking at the technology from a practical point of view. Second edition includes new/revised sections on supervised learning, opinion mining and sentiment analysis, recommender systems and collaborative filtering, and query log mining. Ideally suited for classes on data mining, Web mining, Web search, and knowledge discovery in data bases. Provides internet support with lecture slides and project problems.

  Reference: Web Data Mining Exploring Hyperlinks, Contents, and Usage Data

- **Qingtang Liu, Mingbo Shao, Linjing Wu, Gang Zhao, Guilin Fan:**

  Main content extraction of web pages is widely used in search engines, web content aggregation and mobile Internet browsing. However, a mass of irrelevant information such as advertisement, irrelevant navigation and trash information is included in web pages. Such irrelevant information reduces the efficiency of web content processing in content-based applications. The purpose of this paper is to propose an automatic main content extraction method of web pages. In this method, we use two indicators to describe characteristics of web pages: text density and hyperlink density. According to continuous distribution of similar content on a page, we use an estimation algorithm to judge if a node is a content node or a noisy node based on characteristics of the node and neighboring nodes. This algorithm enables us to filter advertisement nodes and irrelevant navigation. Experimental results on 10 news websites revealed that our algorithm could achieve a 96.34% average acceptable rate.

  Reference: Main content extraction from webpages based on node characteristics

- **Pedro Lopez-Garcia, Antonio D. Masegosa, Eneko Osaba:**

  One of the most challenging issues when facing a classification problem is to deal with imbalanced datasets. Recently, ensemble classification techniques have proven to be very successful in addressing this problem. We present an ensemble classification approach based on feature space partitioning for imbalanced classification. A hybrid metaheuristic called GACE is used to optimize the different parameters related to the feature space partitioning. To assess the performance of the proposal, an extensive experimentation over imbalanced and real-world datasets compares different configurations and base classifiers. Its performance is competitive with that of reference techniques in the literature.

  Reference: Ensemble classification for imbalanced data based on feature space partitioning and hybrid metaheuristics

- **Mehul Mahrishi, Sudha Morwal, Nidhi Dahiya, Hanisha Nankani:**

  For content based indexing of videos, numerous tools and techniques are pipe-lined. The major challenge that these techniques face is the accuracy of index points generated. This paper presents an efficient way to extract text from video frames along with its timestamps. Text extraction takes place in a three-step method which combines pre-processing of extracted Video Frames, similarity measurement for removing ambiguous frames and finally text extraction using PyTesseract Optical Character Recognition. The educational videos with presentations are prioritised. Text extraction is applied upon the headings of that presentation. These extracted keywords are referred to as Index Points through out the article.

  Reference: A framework for index point detection using effective title extraction from video thumbnails

- **Edimar Manica, Carina Friedrich Dorneles, and Renata Galante.:**

  The web is a large repository of entity-pages. An entity-page is a page that publishes data representing an entity of a particular type, for example, a page that describes a driver on a website about a car racing championship. The attribute values published in the entity-pages can be used for many data-driven companies, such as insurers, retailers, and search engines. In this article, we define a novel method, called SSUP, which discovers the entity-pages on the websites. The novelty of our method is that it combines URL and HTML features in a way that allows the URL terms to have different weights depending on their capacity to distinguish entity-pages from other pages, and thus the efficacy of the entity-page discovery task is increased. SSUP determines the similarity thresholds on each website without human intervention. We carried out experiments on a dataset with different real-world websites and a wide range of entity types. SSUP achieved a 95% rate of precision and 85% recall rate. Our method was compared with two state-of-the-art methods and outperformed them with a precision gain between 51% and 66%.

  Reference: Combining URL and HTML Features for Entity Discovery in the Web

- **D. C. Reis, P. B. Golgher, A. S. Silva, A. F. Laender:**

  The Web poses itself as the largest data repository ever available in the history of humankind. Major efforts have been made in order to provide efficient access to relevant information within this huge repository of data. Although several techniques have been developed to the problem of Web data extraction, their use is still not spread, mostly because of the need for high human intervention and the low quality of the extraction results.In this paper, we present a domain-oriented approach to Web data extraction and discuss its application to automatically extracting news from Web sites. Our approach is based on a highly efficient tree structure analysis that produces very effective results. We have tested our approach with several important Brazilian on-line news sites and achieved very precise results, correctly extracting 87.71% of the news in a set of 4088 pages distributed among 35 different sites

  Reference: Automatic web news extraction using tree edit distance

- **Andrés Soto, Héctor Mora, and Jaime A. Riascos:**

  Recently, Machine Learning algorithms have been employed to automate several processes, including software development. However, this action demands large datasets for training these algorithms. To our knowledge, there is no tool for generating synthetic datasets that contain HTML objects (interfaces, codes, wireframe.) Thus, we present the Web Generator, a software designed to mainly provide web pages, designs, and content based on the Bootstrap frontend framework. The software delivers markup code, screenshots, and labels for web elements. We aim to generate enough material for training and exploring the Machine Learning approach for automatic web design and development with this software.

  Reference: An open-source software for synthetic web-based user interface dataset generation

3. Methodologies Used

3.1 Architecture

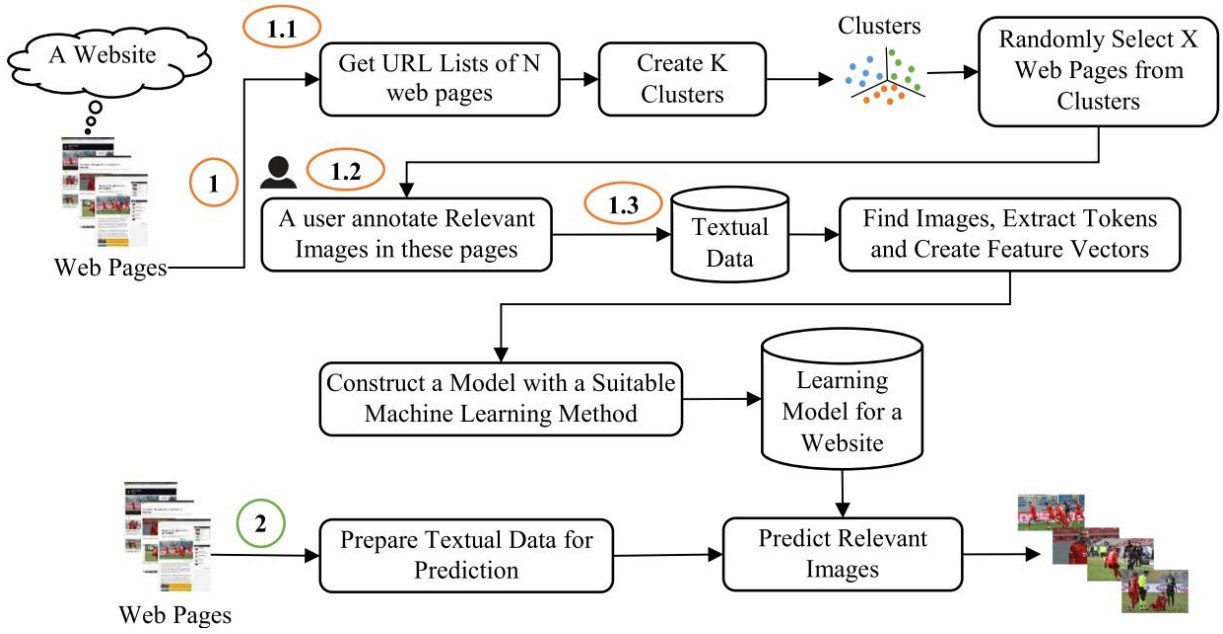

Fig.3.1.1

Initialization:

The initialization step of the TC approach involves collecting web pages from a website. This initial data collection sets the foundation for subsequent processing, including clustering, annotation, and machine learning. In the report, it's important to highlight the significance of this step as it provides the raw material for building the training dataset and ultimately training the model.

3.2 Main topics discussed in the paper

- **Clustering step**

  Once the web pages are collected, the clustering step aims to identify groups of web pages that share similar layouts or content structuresnt a wide range of possibilities. Techniques like Levenshtein distance calculation and DBSCAN (Density-based spatial clustering of applications with noise) may be utilized to cluster web pages based on textual data extracted from HTML elements. The objective is to group web pages with similar structures together, facilitating the selection of representative pages for subsequent annotation.

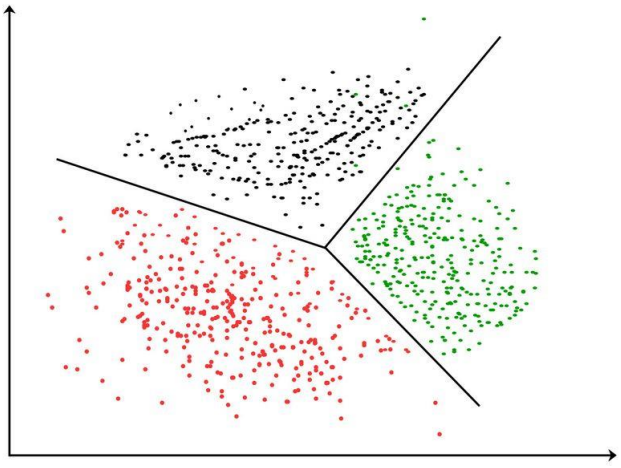

Fig. 3.2.1

- **Annotation Step:**

  In this step, the algorithm proceeds with the URLs suggested by the clustering step for further processing Web pages are downloaded from the selected URLs, and images along with their associated textual data are extracted. Initially, all images are marked as irrelevant. Users are then prompted to manually annotate relevant images by inspecting the content of the web pages. Annotations are updated in the training dataset, providing labeled data for subsequent model training.

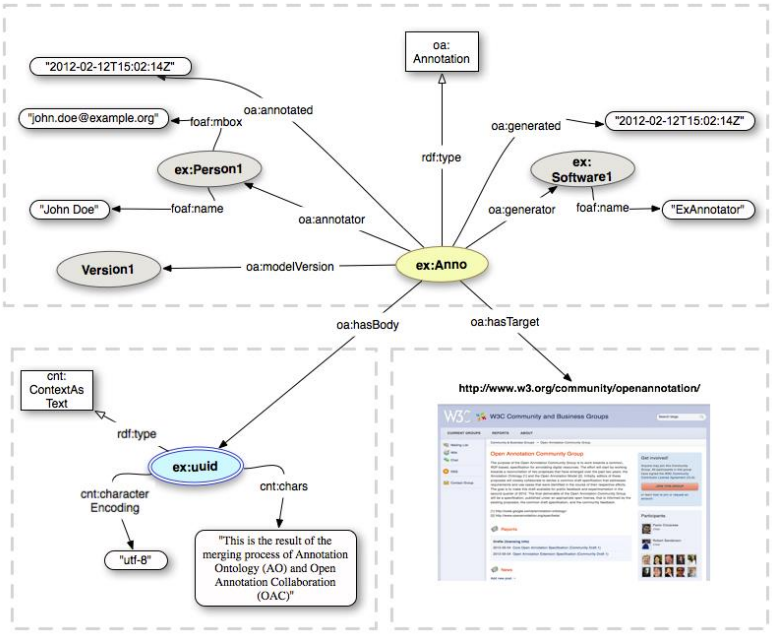

Fig. 3.2.2

- **Machine Learning Step:**

  The annotated training dataset obtained from the annotation step is transformed into feature vectors. Each image's textual data is represented numerically in the feature vectors. Machine learning methods, such as classification algorithms or neural networks, are then applied to construct a learning model. The learning model is trained on the labeled dataset to predict the relevance of images on web pages, based on their textual content.

4. Description of Algorithms

a. Clustering Algorithm

Once the web pages are collected, the clustering step aims to identify groups of web pages that share similar layouts or content structuresnt a wide range of possibilities. Techniques like Levenshtein distance calculation and DBSCAN (Density-based spatial clustering of applications with noise) may be utilized to cluster web pages based on textual data extracted from HTML elements. The objective is to group web pages with similar structures together, facilitating the selection of representative pages for subsequent annotation. The overall procedure of is provided as follows:

Step 1: Initialize an empty list to store selected URLs for training data.

Step 2: Apply the DBSCAN clustering algorithm to the input URLs

Step 3: Initialize an empty list to store the length of each cluster.

Step 4: Iterate over each cluster

- Initialize an empty list to store the length of each cluster.

Step 5: Calculate the remaining training size.

Step 6: If there is remaining training size:

- Adjust the count of URLs in each cluster.
- Select URLs from clusters until the desired training size is reached.

Step 7: Return the selected URLs for training data.

ALGORITHM 1: The Clustering Step of TC

Data:
URLs: URLs downloaded from several web pages on a website
trainingSize: desired number of URLs for the annotation step
Result: *selectedURLs*: recommended URLs for training data

```
1  selectedURLs ← [ ];
2  clusters ← DBSCAN(URLs, lev(URLs));
3  countClusters ← [ ];
4  for i ← 0 to Length(clusters) by i++ do
5      Append selectedURLs ← Select and Pop a URL from clusters_i;
6      Append countClusters ← Length(clusters_i);
7      if trainingSize == Length(selectedPages) then
8          break;
9      end
10 end
11 restTrainingSize = trainingSize - Length(clusters);
12 if restTrainingSize > 0 then
13     for i ← 0 to Length(countClusters)) by i++ do
14         countClusters_i ← countClusters_i * restTrainingSize/ Length(URLs);
15     end
16     for i ← 0 to Length(CountClusters) by i++ do
17         for j ← 0 to clusters_i by j++ do
18             Append selectedURLs ← Select and Pop a URL from clusters_i;
19         end
20     end
21 end
```

Fig. 4.1

b. Annotation Algorithm

In this step, the algorithm proceeds with the URLs suggested by the clustering step for further processing Web pages are downloaded from the selected URLs, and images along with their associated textual data are extracted. Initially, all images are marked as irrelevant. Users are then prompted to manually annotate relevant images by inspecting the content of the web pages. Annotations are updated in the training dataset, providing labeled data for subsequent model training. The overall procedure of is provided as follows:

Step 1: Initialize empty TrainingDataset.

Step 2: Loop through each URL in URLs:

- Download webpage from the URL.
- Parse images from the webpage.

Step 3: Loop through each image in Images:

- Extract textual data from the image
- Append (textual data, 0) to TrainingDataset.

Step 4: Allow the user to annotate relevant images.

Step 5: Loop through each annotation:

- Find the corresponding annotation in TrainingDataset.
- Update the relevant indicator to 1 for the annotated image.

Step 6: Return the TrainingDataset.

ALGORITHM 2: Preparing a Training Dataset and Annotating the Relevant Image

Data:
URLs: N URLs suggested by the clustering step
Images: The image elements extracted from a web page
annotations: The selected image element/s as relevant
textualData: string of Parent 1, Parent 2, and Img. See Table 1
Result: $trainingDataset = [(textualData_i, relevant_i)]$

$trainingDataset \leftarrow [\,]$;
for $i \leftarrow 0$ **to** *Len(URLs))* **by** $i{+}{+}$ **do**
    Download $WebPages_i$ from $URLs_i$;
    Parse $Images \leftarrow WebPages_i$;
    for $j \leftarrow 0$ **to** *Len(Images)* **by** $j{+}{+}$ **do**
        Extract $textualData_j \leftarrow \text{Images}_j$;
        Append $trainingDataset \leftarrow (textualData_j, 0)$;
    end
    annotations ← user annotates only relevant image/s;
    for $k \leftarrow 0$ **to** *Len(annotations)* **by** $k{+}{+}$ **do**
        $i \leftarrow$ Find $annotation_k$ in $trainingDataset$;
        Update $trainingDataset \leftarrow (textualData_j - k, relevant_j \leftarrow 1)$
    end
end

Fig. 4.2

c. Machine Learning Algorithm

The annotated training dataset obtained from the annotation step is transformed into feature vectors. Each image's textual data is represented numerically in the feature vectors. Machine learning methods, such as classification algorithms or neural networks, are then applied to construct a learning model. The learning model is trained on the labeled dataset to predict the relevance of images on web pages, based on their textual content. The overall procedure of is provided as follows:

Step 1: Initialize an empty dictionary to store tokens.

Step 2: Initialize feature Vectors with dimensions.

Step 3: Loop through each item in training dataset:

- Tokenize the textual data.
- For each token in the bag of tokens.

Step 4: Initialize all values in feature vectors to 0.

Step 5: Loop through each item in training dataset:

- For each token in the bag of tokens.

Step 6: Return the training dataset.

ALGORITHM 3: Preparing Feature Vectors

Data: *trainingDataset*: obtained from the annotation step
Result: *dictionary*: contains tokens obtained from the training dataset
$dictionary = \{token_1, token_i, \ldots, token_n\}$
featureVectors : stores the frequency of the tokens for all images in the training dataset
featureVectors = [Len(*trainingDataset*)][Len(*dictionary*)]

for $i \leftarrow 0$ **to** *Len*(*trainingDataset*) **by** $i{+}{+}$ **do**
  $bagofTokens_i \leftarrow$ Tokenization(Pre−tokenization(textualData$_i$));
  foreach *token in* $bagofTokens_i$ **do**
    if *token not in dictionary* **then**
      Append *dictionary* ← *token*
    end
  end
end
for $i \leftarrow 0$ **to** *Len(trainingDataset) - 1* **by** $i{+}{+}$ **do**
  Set all $featureVectors[i]$[Len(*dictionary* - 1] ← 0;
  foreach *token in* $bagofTokens_i$ **do**
    pos = Find the position of *token* in *Dictionary*;
    $featureVectors[i][pos] \leftarrow$ Count *token* in $bagofTokens_i$;
  end
  $featureVectors[i]$[Len(*dictionary*)] ← $trainingDataset.relevant_i$;
end

Fig.4.3

5. Discussion on Results

We encounter the problem known as the imbalanced dataset in the literature, because the number of irrelevant images is too high in our dataset. Accuracy is the ratio of both relevant and irrelevant correct image predictions to all other predictions. For example, if we annotate all images as irrelevant as the simple prediction model, then the accuracy value is 0.96. This value may seem very successful at first glance. However, it has no success in the relevant image prediction. For this reason, accuracy is generally not a very reliable metric. Instead, precision, recall, and f-measure are often used in the literature for performance evaluation. Finally, Log Loss metric is commonly used to evaluate the performance of learning models that make predictions in the form of probability values between 0 and 1. Log Loss measures the discrepancy between the predicted probabilities and the actual class labels. A learning model with a Log Loss value close to 0 is considered to have accurately predicted the class labels, while a higher Log Loss value indicates that the predictions deviate more from the true class labels.

a. Machine Learning Methods

Several machine learning methods, including Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Decision Tree (J48), Random Forests (RF), and AdaBoost, have been compared to discover a more successful learning model. SVM is a discriminative machine learning method that supports different kernel functions, including linear, polynomial, and Gaussian radial basis function (RBF). In this study, the RBF kernel was used because it is useful when the data points are not linearly separable. Besides, this kernel has been implemented and proposed by Vyas and Frasincar [45] for this task. k-NN is a non-parametric method in which weights are calculated for classes for every feature. This non-parametric method is based on the distance for classification. J48 (C4.5) is a decision tree classifier that is a map of the possible classes.

Method	Acc	Rec	Pre	f-M	Log Loss	CT(s)	PT(s)
Methods in the Unsupervised Approaches*							
Bhardwaj and Mangat [6]	0.855	0.922	0.188	0.313	5.002	–	0.782
Helfman and Hollan [17]							
Largest IS	0.958	0.384	0.407	0.395	1.451	–	0.735
Largest W	0.937	0.582	0.301	0.396	2.184	–	0.682
Largest H	0.930	0.570	0.272	0.368	2.416	–	0.702
Highest W*H	0.931	0.590	0.281	0.380	2.373	–	0.652
Gali et al. [14]	0.577	0.843	0.067	0.125	14.615	–	0.755
Fazal et al. [10]	0.840	0.921	0.173	0.292	5.551	–	0.788
Machine Learning Methods in the Supervised Approaches**							
RBF SVM [41, 45]	0.974	0.378	0.671	0.484	0.893	668.192	282.455
k-NN [41]	0.952	0.652	0.341	0.416	1.655	5.859	15.852
J48 [41]	0.964	0.656	0.516	0.572	1.408	3.588	0.068
RF [41]	0.986	0.723	0.860	0.785	0.477	55.916	2.208
AdaBoost [41]	0.987	0.787	0.824	0.805	0.459	27.738	2.936
Machine Learning Methods in the TC Approach***							
RBF SVM	0.983	0.882	0.861	0.840	0.596	**0.011**	0.214
k-NN	0.990	0.891	0.948	0.900	0.329	0.020	0.070
J48	0.992	0.953	0.956	0.947	0.260	**0.011**	**0.001**
RF	**0.995**	0.954	**0.975**	**0.958**	**0.159**	0.121	0.030
AdaBoost	**0.995**	**0.957**	0.971	**0.958**	0.183	0.028	0.012

Table 5.1

Comparison of Performance Results on Unsupervised and Supervised Approaches with the TC Approach

In this section, we compare the average performance results of various methods that can be used in both unsupervised and supervised approaches, as well as our proposed novel approach. These results are the average of five tests, and the details of these tests will be explained in Sections 5.5 and 5.6. Additionally, we explore five machine learning methodsto determine the most appropriate method for both supervised and TC approaches. Table 5 gives the performance results and average construction/prediction time results. In the unsupervised approaches, we conducted tests on the entirety of the dataset, which included all 100 web pages from each of the 200 websites. According to Table 5, the f-Measure of a simple function (width > 300px and height > 400px) proposed by Bhardwaj and Mangat[6] is 0.313. Helfman and Hollan [17] select a single image, which is a large image on a web page, for finding a representative image of a web page. In our experiments, we derive four functions, including the largest image size (IS), width (W), height (H), and W*H in a web page. The f-Measure results for the IS and the W of the image functions are 0.395 and 0.396, respectively, which are the best results among the unsupervised approaches.

	Only with Cluster	
	f-Measure ± Std.	Size of Clusters ± Std.
	0.664 ± 0.410	2.607 ± 1.615

Number of URLs *trainingSize*	**Random Selection** f-Measure ± Std.	**Clustering Step of TC** f-Measure ± Std.
1	0.483 ± 0.426	
2	**0.806 ± 0.280***	0.676 ± 0.408
3	0.889 ± 0.199	0.876 ± 0.233
4	0.915 ± 0.156	**0.927 ± 0.153***
5	0.921 ± 0.142	**0.947 ± 0.120***
6	0.941 ± 0,118	**0,958 ± 0,093***
7	0.947 ± 0,116	**0,959 ± 0,098***
8	0.950 ± 0,106	**0,965 ± 0,081***
9	0.950 ± 0,109	**0,970 ± 0,073***
10	0.958 ± 0,093	**0,970 ± 0,074***
11	0.959 ± 0,092	**0,974 ± 0,070***
12	0.967 ± 0,082	**0,975 ± 0,065***
13	0.968 ± 0,082	**0,975 ± 0,064***
14	0.967 ± 0,084	**0,977 ± 0,063***
15	0.969 ± 0,082	**0,980 ± 0,051***

Table 5.2

Size of URLs	f-Measure ± Std.
10	0.939 ± 0.143
20	0.942 ± 0.144
30	0.951 ± 0.128
40	**0.953 ± 0.126**
50	**0.955 ± 0.113**
60	**0.958 ± 0.112**
70	**0.959 ± 0.106**
80	**0.958 ± 0.109**
90	**0.958 ± 0.116**
100	**0.958 ± 0.093**

Table 5.3

Number of Websites for Training	**Dataset 1** f-Measure ± Std.	**Dataset 2** f-Measure ± Std.
20	**0.659 ± 0.105***	0.195 ± 0.046
40	**0.749 ± 0.030***	0.249 ± 0.032
60	**0.763 ± 0.008***	0.264 ± 0.042
80	**0.790 ± 0.018***	0.272 ± 0.043
100	**0.805 ± 0.019***	0.289 ± 0.036
120*	**0.805 ± 0,016***	0,304 ± 0,044
140	**0.800 ± 0,018***	0,293 ± 0,080
160	**0.797 ± 0,020***	0,271 ± 0,077

Table 5.4

6. Conclusion and Future work

In this study, we introduced a semi-automatic approach for extracting relevant images from web pages within a single website. We addressed the challenge of accurately identifying relevant images, which typically demands significant resources and expertise. By leveraging textual data from web pages, our approach eliminates the need for manual pattern generation, offering a more efficient and accessible solution. Our method demonstrates notable improvements in both performance metrics and execution time compared to generalized solutions. Furthermore, its seamless integration into existing web scraping tools underscores its practical utility for websites with extensive content. Overall, our study highlights the efficacy of semi-automatic approaches for image extraction tasks, particularly in contexts where manual efforts are impractical or time-consuming.

Future Work:

Moving forward, our research will focus on several key areas. Firstly, we aim to extend our approach to address diverse web extraction tasks beyond image extraction, exploring its adaptability and effectiveness in various scenarios. Secondly, we plan to refine our methodology by identifying optimal regular expressions for scraping data from web pages, incorporating insights gained from analyzing both positive and negative textual content within HTML elements. Thirdly, we aspire to generalize our solution by constructing learning models that can be applied across multiple websites, thereby enhancing its scalability and versatility. Lastly, we will explore enhancements to the clustering process, particularly for websites with complex layouts, to further streamline the data extraction workflow and improve overall performance. Through these future endeavors, we seek to advance the field of web data extraction and contribute towards more efficient and robust methodologies for extracting valuable insights from web content.

7. References

[1] Hayri Volkan Agun and Erdinç Uzun. 2023. An efficient regular expression inference approach for relevant image
extraction. Appl. Soft Comput. 135 (2023), 110030. DOI:https://doi.org/10.1016/j.asoc.2023.110030

[2] Julian Alarte, David Insa, Josep Silva, and Salvador Tamarit. 2018. Main content extraction from heterogeneous webpages. In Web Information Systems Engineering (WISE'18), Hakim Hacid, Wojciech Cellary, Hua Wang, Hye-Young
Paik, and Rui Zhou (Eds.). Springer International Publishing, Cham, 393–407.

[3] Naseer Aslam, Bilal Tahir, Hafiz Muhammad Shafiq, and Muhammad Amir Mehmood. 2019. Web-AM: An efficient
boilerplate removal algorithm for web articles. In International Conference on Frontiers of Information Technology
(FIT'19). IEEE, 287–2875. DOI:https://doi.org/10.1109/FIT47737.2019.00061

[4] Ziv Bar-Yossef and Sridhar Rajagopalan. 2002. Template detection via data mining and its applications. In 11th International Conference on World Wide Web (WWW'02). Association for Computing Machinery, New York, NY, 580–591.
DOI:https://doi.org/10.1145/511446.511522

[5] Rodrigo Barbado, Oscar Araque, and Carlos A. Iglesias. 2019. A framework for fake review detection in online consumer electronics retailers. Inf. Process. Manag. 56, 4 (2019), 1234–1244. DOI:https://doi.org/10.1016/j.ipm.2019.03.002

[6] Aanshi Bhardwaj and Veenu Mangat. 2014. An improvised algorithm for relevant content extraction from web pages.
J. Emerg. Technol. Web Intell. 6, 2 (May 2014), 226–230.

DOI:https://doi.org/10.4304/jetwi.6.2.226-230

[7] Lidong Bing, Tak-Lam Wong, and Wai Lam. 2016. Unsupervised extraction of popular product attributes from ecommerce web sites by considering customer reviews. ACM Trans. Internet Technol. 16, 2, Article 12 (Apr. 2016), 17

pages. DOI:https://doi.org/10.1145/2857054

[8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering

clustersin large spatial databases with noise. In 2nd International Conference on Knowledge Discovery and Data Mining

(KDD'96). AAAI Press, 226–231.

[9] Fadwa Estuka and James Miller. 2019. A pure visual approach for automatically extracting and aligning structured

web data. ACM Trans. Internet Technol. 19, 4, Article 51 (Nov. 2019), 26 pages. DOI:https://doi.org/10.1145/3365376

[10] Nancy Fazal, Khue Nguyen, and Pasi Fränti. 2019. Efficiency of web crawling for geotagged image retrieval. Webology

16 (2019), 16–39. DOI:https://doi.org/10.14704/WEB/V16I1/a177

[11] Emilio Ferrara and Robert Baumgartner. 2011. Automatic wrapper adaptation by tree edit distance matching. In

Combinations of Intelligent Methods and Applications. Springer, UK, 41–54.

[12] Leandro Neiva Lopes Figueiredo, Guilherme Tavares de Assis, and Anderson A. Ferreira. 2017. DERIN: A data extraction method based on rendering information and n-gram. Inf. Process. Manag. 53, 5 (2017), 1120–1138. DOI:https:

//doi.org/10.1016/j.ipm.2017.04.007

[13] Jeffrey E. F. Friedl and Andy Oram. 2002. Mastering Regular Expressions (2nd ed.).

O’Reilly & Associates, Inc.

[14] Najlah Gali, Andrei Tabarcea, and Pasi Fränti. 2015. Extracting representative image from web page. In 11th International Conference on Web Information Systems and Technologies (WEBIST’15). INSTICC, SciTePress, Portugal, 411–419.

DOI:https://doi.org/10.5220/0005438704110419

[15] Waqar Haider and Yeliz Yesilada. 2022. Classification of layout vs. relational tables on the web: Machine learning with

rendered pages. ACM Trans. Web 17, 1, Article 1 (Dec. 2022), 23 pages. DOI:https://doi.org/10.1145/3555349

[16] Wook-Shin Han, Wooseong Kwak, Hwanjo Yu, Jeong-Hoon Lee, and Min-Soo Kim. 2014. Leveraging spatial join for

robust tuple extraction from web pages. Inf. Sci. 261 (2014), 132–148. DOI:https://doi.org/10.1016/j.ins.2013.09.027

[17] Jonathan I. Helfman and James D. Hollan. 2000. Image representations for accessing and organizing web information.

In Internet Imaging II, Giordano B. Beretta and Raimondo Schettini (Eds.), Vol. 4311. International Society for Optics

and Photonics, SPIE, San Jose, CA, 91–101. DOI:https://doi.org/10.1117/12.411880

[18] Chun-Nan Hsu and Ming-Tzung Dung. 1998. Generating finite-state transducers for semi-structured data extraction

from the web. Inf. Syst. 23, 8 (1998), 521–538. DOI:https://doi.org/10.1016/S0306-4379(98)00027-1

[19] Imranul Islam. 2021. Representative Image Extraction from Web Page. Master’s Thesis. University of Eastern Finland,

Faculty of Science and Forestry, Joensuu School of Computing.

[20] Patricia Jiménez, Juan C. Roldán, and Rafael Corchuelo. 2021. A clustering approach to extract data from HTML

tables. Inf. Process. Manag. 58, 6 (2021), 102683. DOI:https://doi.org/10.1016/j.ipm.2021.102683

[21] Yeonjung Kim, Jeahyun Park, Taehwan Kim, and Joongmin Choi. 2007. Web information extraction by HTML tree edit

distance matching. In International Conference on Convergence Information Technology (ICCIT'07). IEEE, 2455–2460.

DOI:https://doi.org/10.1109/ICCIT.2007.19

[22] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In 3rd ACM International Conference on Web Search and Data Mining (WSDM'10). Association for Computing

Machinery, New York, NY, 441–450. DOI:https://doi.org/10.1145/1718487.1718542

[23] Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. Sov.

Phys. Dokl. 10, 8 (1966), 707–710.

[24] Bing Liu. 2011. Web Data Mining Exploring Hyperlinks, Contents, and Usage Data. Springer, Berlin. DOI:https://doi.

org/10.1007/978-3-642-19460-3

[25] Qingtang Liu, Mingbo Shao, Linjing Wu, Gang Zhao, Guilin Fan, and Jun Li. 2017. Main content extraction from web

pages based on node characteristics. J. Comput. Sci. Eng. 11 (06 2017), 39–48. DOI:https://doi.org/10.5626/JCSE.2017.

11.2.39

[26] Pedro Lopez-Garcia, Antonio D. Masegosa, Eneko Osaba, Enrique Onieva, and Asier

Perallos. 2019. Ensemble classification for imbalanced data based on feature space partitioning and hybrid metaheuristics. Appl. Intell. 49, 8 (Aug.
2019), 2807–2822. DOI:https://doi.org/10.1007/s10489-019-01423-6

[27] Mehul Mahrishi, Sudha Morwal, Nidhi Dahiya, and Hanisha Nankani. 2021. A framework for index point detection
using effective title extraction from video thumbnails. Int. J. Syst. Assur. Eng. Manag. (June 2021), 1–6. DOI:https:
//doi.org/10.1007/s13198-021-01166-z

[28] Edimar Manica, Carina Friedrich Dorneles, and Renata Galante. 2019. Combining URL and HTML features for entity
discovery in the web. ACM Trans. Web 13, 4, Article 20 (Dec. 2019), 27 pages. DOI:https://doi.org/10.1145/3365574

[29] Ion Muslea, Steve Minton, and Craig Knoblock. 1999. A hierarchical approach to wrapper induction. In 3rd Annual
Conference on Autonomous Agents (AGENTS'99). Association for Computing Machinery, New York, NY, 190–197.
DOI:https://doi.org/10.1145/301136.301191

[30] D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. 2004. Automatic web news extraction using tree edit distance.
In 13th International Conference on World Wide Web (WWW'04). Association for Computing Machinery, New York,
NY, 502–511. DOI:https://doi.org/10.1145/988672.988740

[31] Arnaud Sahuguet and Fabien Azavant. 1999. Building light-weight wrappers for legacy web data-sources using W4F.
In 25th International Conference on Very Large Data Bases (VLDB'99). Morgan Kaufmann

PublishersInc., San Francisco,

CA, 738–741.

[32] Roland Schäfer. 2017. Accurate and efficient general-purpose boilerplate detection for crawled web corpora. Lang.

Resour. Eval. 51 (2017), 873–889.

[33] Robert E. Schapire and Yoav Freund. 2012. Boosting: Foundations and Algorithms. The MIT Press, London, England.

[34] Andrés Soto, Héctor Mora, and Jaime A. Riascos. 2022. Web generator: An open-source software for synthetic webbased user interface dataset generation. SoftwareX 17 (2022), 100985. DOI:https://doi.org/10.1016/j.softx.2022.100985