A VISUAL WEBPAGE INFORMATION EXTRACTION FRAMEWORK FOR COMPETITIVE INTELLIGENCE SYSTEM

ZHIWEI ZHANG∗, WENBO QIN†, AND HAIFENG XU‡

Abstract. The extraction of webpage information is of paramount importance in the realm of competitive intelligence. This research is dedicated to the design and implementation of a visual webpage information extraction module within a competitive intelligence system, approached through the lens of research and development (R&D) technology and its practical applications. Initially, the study delineates the objectives and requirements for webpage information extraction, emphasizing the practical needs of competitive intelligence systems. By critically assessing the strengths and weaknesses of current theories and methodologies in webpage text information extraction, this paper introduces an innovative visual method for extracting webpage text information. Subsequently, the paper meticulously outlines the comprehensive architecture of the proposed module. Building upon this foundation, the study delves into the specifics of the extraction template, rule generation, optimization techniques, and the extraction algorithm pivotal to the process of visual webpage information extraction. The system’s effectiveness and practical utility are substantiated through a series of confirmatory experiments, the results of which are thoroughly analyzed. The findings affirm that the developed system adeptly fulfills the webpage information extraction needs of competitive intelligence systems, contributing significantly to the R&D efforts in which the authors are engaged.

Key words: Information extraction visualization, natural language processing, competitive intelligence, data mining

1. Introduction. With the exponential growth of the Internet and significant advancements in information technology, the digital landscape has emerged as the predominant arena for corporate and institutional information dissemination. An array of content, including corporate profiles, product details, promotional events, recruitment opportunities, and technological advancements, is extensively published online [25, 26, 14]. Recent studies underscore that a vast majority (approximately 90%) of the data requisites for competitive intelligence analyses are sourced from the Internet, underscoring the critical role of web-based information in today’s business milieu [33, 19, 28]. Competitive intelligence encompasses a comprehensive system of information gathering, analysis, and dissemination, designed to equip businesses with profound insights into competitors, market dynamics, and industry evolutions. Within this framework, webpage information extraction assumes a pivotal position, empowering entities to not only distill invaluable insights from the deluge of data but also to leverage such insights for tangible competitive edges. Particularly in the big data era, navigating the complexities of data processing and analysis presents formidable challenges [19, 18, 24].

Webpage information extraction refers to the process of automatically retrieving structured or semi-structured information from unstructured webpage content. This process involves identifying relevant pieces of data within a webpage, such as text, images, links, and other multimedia elements, and then transforming this data into a more organized format that is suitable for analysis, storage, and further processing [1, 13]. The goal of webpage information extraction is to enable computers to understand and utilize the vast amount of information available on the Internet efficiently. This technology underpins various applications, including search engines, competitive intelligence systems, market research, and content aggregation services, facilitating the automatic collection and analysis of web data [27, 17].

In the realm of competitive intelligence systems, the task of webpage information extraction encounters significant hurdles due to the sheer diversity and complexity of webpage contents. The digital landscape is a mosaic of information presented in myriad forms—ranging from text, images, videos, tables, to interactive

∗School of Informatics and Engineering, Suzhou University, Suzhou, China. Corresponding author: Zhiwei Zhang (zzwloveai@gmail.com).
†School of Informatics and Engineering, Suzhou University, Suzhou, China.
‡School of Informatics and Engineering, Suzhou University, Suzhou, China.
A Visual Webpage Information Extraction Framework for Competitive Intelligence System

elements, each differing vastly in format and structure across websites. This variability presents a formidable challenge in crafting a one-size-fits-all, efficient information extraction system [8, 6]. For example, product descriptions might be straightforward text or be supplemented with images and videos, while technical articles could include intricate diagrams or snippets of code. Moreover, the inherent complexity within HTML structures, coupled with the ever-evolving designs of webpages, complicates the accurate retrieval of information, necessitating highly flexible and adaptive strategies for extraction [21, 12]. Ensuring the comprehensiveness and precision of extracted information thus requires leveraging sophisticated text processing and image recognition technologies [9, 23].

Furthermore, competitive intelligence systems grapple with the challenges posed by the variability of noise data, webpage formats, and structures. The internet is inundated with noise—advertisements, promotional links, and irrelevant comments clutter webpages, obscuring valuable information. Effectively sifting through and excluding such noise is a crucial task in the extraction process [30, 5]. This necessitates not only the development of sophisticated algorithms capable of discerning between relevant and irrelevant content but also their ongoing refinement to keep pace with the dynamic nature of the web. The diversity in webpage design and layout demands that extraction systems be versatile enough to navigate a variety of HTML/CSS structures [20, 15]. Additionally, the rapid evolution of web technologies and the introduction of new standards for webpage design and layout further amplify the complexity, challenging information extraction systems to continuously adapt to these new formats and structures [2, 31].

The challenges of real-time updates and information overload significantly complicate the task of webpage information extraction. The dynamic nature of the Internet, with its continuous influx of new content and the potential modification or removal of existing information, presents a critical challenge for competitive intelligence systems. Timely tracking and management of these changes are crucial, as reliance on outdated or inaccurate information could result in erroneous analyses and decision-making [11, 29]. Furthermore, the exponential growth in online content has precipitated an era of information overload, posing a substantial challenge. Competitive intelligence systems are thus tasked with the efficient processing and filtering of this vast amount of data to ensure the extraction and analysis of only the most relevant and valuable information. Consequently, these systems must possess not only robust data processing capabilities but also sophisticated data screening and prioritization mechanisms. Such features are essential to navigate through the vast data landscape effectively, preventing the degradation of analysis efficiency or decision-making errors that could arise from information overload [32, 3].

Within the domain of competitive intelligence systems, visual webpage information extraction technology presents clear advantages over traditional text extraction methodologies [4, 7]. Primarily, the visual approach enhances the efficiency and accuracy of information extraction by offering an intuitive display of data structures and content. Traditional methodologies, which predominantly rely on semantic analysis and keyword matching, frequently fall short in addressing webpages characterized by intricate structures or varied formats. Conversely, visual extraction technology capitalizes on the visual layout and structural features of webpages to more precisely locate and extract pivotal information. For instance, by scrutinizing the Document Object Model (DOM) structure and visual indicators on webpages, elements such as article titles, texts, images, and tables can be more effectively identified. Furthermore, visual information extraction methods substantially bolster data understanding and analysis. This approach simplifies the process for users to comprehend the overarching structure and key components of the data. The intuitive nature of this representation not only facilitates a swift comprehension of the information’s essence but also aids analysts in uncovering potential data interconnections. Crucially, the visualization technique assumes added significance in the context of big data processing [16]. With data volumes expanding continuously, traditional text extraction and analysis techniques are increasingly overwhelmed. Visualization technology, through its capacity to efficiently manage and present large datasets via summary views and interactive exploration features, empowers users to grasp a higher-level understanding of the data. This, in turn, allows for the rapid identification of areas of interest, followed by detailed analysis [10, 22].

In summary, the realm of webpage information extraction technologies currently grapples with significant challenges, chiefly arising from the diversity and complexity of web content, the proliferation of noise and extraneous information, and the dynamic evolution of webpage formats and structures. These obstacles underscore the critical necessity for extraction systems that are both highly adaptable and technologically advanced,
equipped to identify and process a wide array of information types. The progression of competitive intelligence systems is contingent upon the development of more refined algorithms dedicated to noise filtration and the incorporation of cutting-edge techniques in text and image processing. Such enhancements are essential to augment the precision and depth of information extraction. Tackling these prevalent issues is crucial for the enhancement of competitive intelligence systems, enabling them to effectively and accurately harness the extensive reservoir of data available online.

Hence, the efficacy of competitive intelligence systems hinges critically on their ability to precisely distill targeted text information from the disarray of webpage source codes and to subsequently generate standardized structured documents. Such a foundational step is indispensable for the successful execution of text classification and text mining processes. In this research, we introduced a comprehensive visual webpage information extraction framework tailored to meet the specific text mining needs of competitive intelligence systems. The primary endeavors and contributions of this study are summarized as follows:

1. This research meticulously articulated the design objectives and requirements for webpage information extraction, aiming to develop a framework that stands out for its efficiency, accuracy, and user-friendliness. Unlike existing methodologies that often struggle with the dynamic and complex nature of web data, this framework is specifically engineered to adeptly handle such challenges. It focuses on extracting crucial information by ensuring data integrity and accuracy, enhancing processing speed, and expanding system scalability, all while improving the ease of user interaction. This approach addresses a significant gap in current practices, where the balance between comprehensive data extraction and user-centric design often remains unachieved.

2. We introduced a holistic visual webpage information extraction framework, distinct from current solutions by its integration of cutting-edge technologies across data crawling, natural language processing, and visualization. This framework is designed to facilitate automated data extraction from a diverse array of webpages with minimal human intervention. The novel incorporation of visual elements specifically aims to demystify the user interaction process and elevate the intuitiveness of data presentation. This strategy marks a departure from traditional methods that may not fully leverage visual cues for user engagement and data interpretation, highlighting the innovative edge of our approach in enhancing both system usability and information clarity.

3. The development and validation of a comprehensive visual webpage information extraction process underscored its applicability and effectiveness across various webpage types, both static and dynamic. The empirical evidence gathered from these experiments showcases our method’s superiority in not only improving the precision of information extraction but also in significantly lightening the user’s workload. This contrasts sharply with many existing techniques that may exhibit limitations in versatility across different webpage formats or impose a heavier analytical burden on users. Furthermore, the identification of potential areas for future enhancements opens new avenues for advancing the state-of-the-art in webpage information extraction. These findings offer a critical reflection on the gaps within current methodologies and provide a clear direction for subsequent research efforts, aiming to refine and augment the capabilities of competitive intelligence systems in navigating the vast and varied terrain of web information.

The remainder of this study was organized as follows: In Section 2, the design objectives and requirements of visual webpage information extraction were introduced; in Section 3, the overall framework for the visual webpage information extraction system was generalized, and three subsystems—webpage information crawling, visual webpage information extraction rule template generation, and webpage text information extraction—were elaborated on; in Section 4, the relevant experimental environment and experimental data were summarized, and the experimental results were analyzed; in the final section, conclusions were drawn and the future research directions were put forward.

2. Design objectives and requirements of visual webpage information extraction. In the context of the contemporary era, characterized by an explosion of network information, the utilization of web crawlers for webpage crawling on the Internet has become a pivotal means of information collection. However, the challenge lies in the complexity of the original webpage content obtained by these crawlers, which is often encumbered with a plethora of HTML tags. Hidden within these tags is a vast amount of text information, making the extraction process intricate and demanding.

The primary design objective of visual webpage information extraction is to navigate through this complex
maze of webpage source codes and distill high-quality text information that aligns with user needs. This task involves meticulously parsing the cluttered webpage data, identifying and extracting relevant information, and transforming it into standardized data formats such as titles, texts, and other pertinent entries. Achieving this requires a sophisticated system capable of discerning and isolating useful content from the plethora of unstructured data typically found in webpages.

From a practical application standpoint, this objective is not only feasible but increasingly necessary. The vast and growing volume of web-based information necessitates efficient and accurate extraction methods to harness this data for meaningful use. The proposed system addresses this need by employing advanced extraction techniques, focusing on the critical aspects of accuracy, reliability, and speed. In terms of system implementation, however, the feasibility is grounded in current technological advancements in web crawling, HTML parsing, and data extraction algorithms. The system’s design will leverage state-of-the-art techniques in these areas, ensuring that it can effectively handle the diverse and dynamic nature of web content. This includes the capability to adapt to various webpage structures, handle different types of content (including multimedia elements), and process information rapidly and accurately.

In summary, the design objective of extracting high-quality, user-centric text information from webpages is both practical and attainable. It resonates with the current demands of information extraction in the digital age and is supported by feasible technological solutions. The implementation of such a system promises significant benefits in terms of enhancing the efficiency and effectiveness of web-based information collection and analysis. Based on the analysis aforementioned above, the design and requirements of the webpage information extraction system must focus on the following key aspects:

1. Accuracy and completeness: The accuracy of extraction is the primary system design objective, meaning that the system should focus on extracting specific target items (such as titles, texts, and abstracts) while excluding all unspecified noise information. In addition to accuracy, completeness is also of crucial importance, ensuring that the extracted target items retain complete contextual semantics so as to provide support for subsequent operations on standardized documents.

2. Timeliness and efficiency: Considering the huge amount of webpage resources on the Internet, the webpage information extraction system needs to have efficient processing ability. Web crawlers, which usually run in a distributed and multithreaded way, can grab a large number of original webpages in a short time. Therefore, the system should be able to extract specific target items accurately and completely from the original webpages in time and efficiently and quickly form standardized documents.

3. Adaptability: The rapid development of Web technology means that the structure of webpages is frequently updated. In this context, the web information extraction system needs to be self-adaptable to some extent. When the existing information extraction templates and rules fail to correctly extract webpage information, for instance, the system should be able to give an alarm and use the wrong feedback information to adjust the extraction rules and templates in time to adapt to the latest webpage structure.

4. Practicality: While meeting the professional requirements, the webpage information extraction system should also consider the use needs of non-professional users. This means that the user interface of the system should be intuitive and easy to use, while ensuring the powerful and stable functions of the system to adapt to the operating habits and skill levels of different users.

To sum up, in order to effectively support the collection and analysis of competitive intelligence, the web information extraction system must meet high standards in accuracy, timeliness, adaptability, and practicality.

3. Overall framework for the visual webpage information extraction system. The visual webpage information extraction system is comprised of three integral subsystems: (1) a webpage information crawling subsystem, which retrieves data from the internet; (2) a visual webpage information extraction rule template generation subsystem, tasked with creating rules for data identification and extraction; and (3) a webpage text information extraction subsystem, dedicated to isolating textual content from webpages. The comprehensive structure of this system is illustrated in Figure 3.1, providing a cohesive overview of its components and operational flow.

From the overall framework for the visual webpage information extraction system as shown in Figure 3.1, the system is generally divided into three subsystems whose composition, functions, and techniques used are briefed as follows:
Zhiwei Zhang, Wenbo Qin, Haifeng Xu

Web sites

Page dimensions: 595.3x841.9

Fig. 3.1: Overall framework structure for the visual webpage information extraction system.

(1) Webpage Information Crawling Subsystem: This subsystem employs a webpage crawler to gather webpage information (source code) from designated sites based on predetermined seed site URLs and crawling strategies. The captured data is stored in the “sample webpage library” depicted in Figure 3.1 for future use or as input for the “webpage text information extraction subsystem” for text data extraction. Developed on the open-source Nutch search engine, this subsystem has been customized to fulfill the unique requirements of our research.

(2) Visual Webpage Information Extraction Rule Template Generation Subsystem: As illustrated in Figure 3.1, this subsystem integrates the “sample webpage library,” “sample webpage classifier,” “visual webpage extraction template generator,” and “webpage extraction template library.” Representing the innovative core of this study, it generates extraction rule templates for specific webpages using samples from different sections of the target site, storing these templates in the “webpage extraction template library.” It primarily utilizes the open-source htmlparser, which has been adapted and refined for this research.

(3) Webpage Text Information Extraction Subsystem: Comprising the “webpage extraction template selector,” “webpage information extractor,” “extraction effect evaluator,” and “webpage template adjuster” as shown in Figure 3.1, this subsystem extracts information from the original webpages collected by the “webpage information crawling subsystem.” It utilizes site-specific extraction rule templates developed by the “visual webpage information extraction rule template generation subsystem” to output standardized webpage text information. Utilizing the open-source htmlparser for formatting and extracting text from specific webpage tags, this component has been enhanced for research purposes.

The aforesaid subsystems will be introduced one by one from the angles of design and implementation, and the algorithm used by each subsystem will be introduced and analyzed in detail.
3.1. **Webpage information crawling subsystem.** This subsystem focuses on retrieving webpages from the Internet, archiving the initially crawled content in a sample webpage library for subsequent use or directly forwarding it to the webpage text information extraction subsystem for data extraction. It has been developed and enhanced using the open-source Nutch search engine, tailored specifically to meet the research requirements of this study. The comprehensive structure of the webpage crawling subsystem is depicted in Figure 3.2.

The workflow of the improved webpage crawling subsystem is as follows:

1. An initial set of URLs for crawling sites is established, comprising a document that lists URLs from various seed sites.
2. These seed site URLs are then populated into CrawlDB, a database dedicated to storing the URLs and their status information for webpages.
3. Utilizing the data within CrawlDB, a comprehensive list of URLs for site crawling is compiled, based on both the URL and status information.
4. Employing the Nutch multi-threaded crawler, the system proceeds to crawl webpages as delineated by the “URL list of crawling sites,” generating relevant webpage snapshots from the content retrieved during the crawl and logging the crawl process. Concurrently, hyperlinks contained within the webpages are analyzed, facilitating the output of the original webpage stream.

The webpage hyperlink information parsed in Step (4) is used to update CrawlDB; Steps (3) to (4) are repeated until reaching the preset crawling depth or the crawling task is manually stopped, so as to form a cyclic process of “generation-crawling-updating”.

3.2. **Visual webpage information extraction rule template generation subsystem.** In the visual operation environment, the objective is to achieve precise and clear generation of information extraction rule templates, thereby obviating the need for complex algorithm-derived parsing rule templates. This section delves into and implements the visual webpage information extraction rule template generation process and its specifics. Within the browser/server (B/S) visual environment, the “target extraction item region” is selected via mouse, facilitating the automatic generation and extraction of rules for the “target extraction items.” Concurrently, specific extraction rules are compared and calibrated within this environment. The detailed implementation steps are outlined as follows:
(1) All the sample webpages belonging to the site S are read from the sample webpage library, and displayed after sorting according to their "webpage URL", so as to facilitate the subsequent "URL template" configuration for each module of the site S through the observation method.

(2) With site S’s samples displayed post-sorting, webpages from different modules naturally cluster together. A URL template for module T of site S is set up through observation—differentiating sample webpages by URL strings with identical prefixes and manually crafting matching URL regular expressions.

(3) A unique “extraction template” is created for each module. If module T on site S already has a corresponding “extraction” template, this step is bypassed; otherwise, a unique extraction template P is generated.

(4) Specific extraction rules for “target extraction items” are established for all webpages within module T of site S. It’s important to recognize that all webpages in module T are issued through the same “information release template” backstage, making their structures identical (or similar) aside from text differences. In this research, three sample webpages from module T were chosen at random to visually create extraction rules for “target extraction items.”

3.3. Webpage text information extraction subsystem. The webpage information extraction subsystem constitutes a crucial component of the comprehensive visual webpage information extraction system. It employs predefined extraction rule templates to retrieve pertinent data from “target extraction items” on webpages, aggregating these items into a standardized document output. This subsystem is comprised of the extraction template selector, webpage information extractor, extraction effect evaluator, and webpage extraction rule template adjuster, as depicted in Figure 3.3.

3.3.1. Relevant data structure. The data structures utilized for extracting information from webpages via extraction rule templates are outlined as follows:

(1) Webpage parsing result object tagNodeList: The NodeList-type object obtained by parsing the webpage source code with htmplparser is a tree-structured type of data and an operation object generated by “extracting target items” with the webpage extraction rules.

(2) Extraction rule ParseRule: The position of the “target extraction item region” in the webpage is marked, that is, the absolute path (webpage tag sequence) from the DOM tree root node of the webpage to the “target extraction item region”, aiming to guide the “webpage information extraction subsystem” to extract the specific “target extraction items”.

(3) Extraction template ParseTemplate: It is a set of the extraction rules for each “target extraction item”, and the concrete templates for the specific sites are encapsulated.

(4) Extraction rule list ruleList: It is a data structure of List<ParseRule> type, which stores the extraction rules for all “target extraction items” under the same extraction template.

(5) Extraction rule list itemRuleList of “target extraction items: It is also a data structure of List<ParseRule> type, which encapsulates multiple extraction rules for a specific “target extraction item”.

(6) Extraction rule tag list ruleTagList: It is the object of List<String> type. The extraction rule is a tag string sequence composed of all tags on the path from the tag of the root node of the webpage to the tag of the parent node of the target extraction item, which is a whole character string, and the ruleTagList stores a list of character strings with a single tag as a character string, that is, a list composed of single tags.

(7) Matching tag sequence stack pathStack and node child queue childQueue: The pathStack records already matched path tags, being the objects of Stack<TagNode> type, and childQueue encapsulates all child nodes of one node.

(8) Standard document StructuredDoc: It is a data structure of Map<TargetItem, Text> type, which encapsulates the text information of target extraction items, including the titles, abstracts, keywords, and texts of webpages.

3.3.2. System workflow. Figure 3.3 illustrates the comprehensive workflow of the "webpage text information extraction subsystem,” detailed as follows:

(1) The web crawler of the “webpage crawling subsystem” designed in Section 3.1 is enabled to configure the information of the crawling site and the related attributes of the web crawler to the site $S$ (siteID) in a multi-threading way for webpage crawling. Then, the URL pageURL of webpages is extracted and the webpages are returned in the form of source codes to generate the character string pageSource output of webpage source
Fig. 3.3: The workflow of the webpage text information extraction subsystem. Initially, the subsystem commences with the advanced data crawling phase, employing state-of-the-art technology to navigate and retrieve content from a diverse array of web sources. This phase is critical for ensuring that the most current and relevant webpage data is captured for analysis, addressing the challenge of the internet’s ever-evolving content landscape. Then the subsystem applies natural language processing (NLP) tools to the raw webpage content. This involves sophisticated semantic analysis to identify and isolate valuable text information from the surrounding web elements and noise. The NLP phase is crucial for refining the data into a format that is both meaningful and actionable, setting our approach apart from conventional methods that may struggle with the complexity of web content structures. The final stage of the workflow integrates visualization technology to present the extracted information in an intuitive and user-friendly manner. This not only simplifies the interaction process for users but also enhances their ability to comprehend and analyze the data.

codes.

(2) SiteID in Step (1) is used to acquire all URL template urlTemplateList belonging to this site from the database. If urlTemplateList is empty, skip to Step 3.3.2, or otherwise, pageURL in Step (1) is matched with the URL regular expression in urlTemplateList one by one. If matching fails all the time, skip to Step (11). If matching succeeds for a single template, the urlTemplateId of this URL template is recorded.

(3) The extraction template ID parseTemplateId belonging to this URL template is acquired from the database. If it is empty, skip to Step (11), or otherwise, the next step will be implemented.
(4) All extraction rules belonging to this extraction template are searched in the database via parseTemplateId in Step (3), and a value is assigned to ruleList; if ruleList is empty, skip to Step (11), or otherwise, the next step will be implemented.

(5) The extraction rules in Step (4) are classified according to the target extraction items to form the extraction rule itemRuleList of each target extraction item PT.

(6) The pageSource in Step (1) is parsed using htmlparser to generate tagNodeList. Since the operation on tagNodeList is destructive, pageSource is parsed again using htmlparser to generate new tagNodeList before extracting the text information of different “target extraction items”.

(7) Each extraction rule object in itemRuleList in Step (5) is cyclically traversed. Each extraction rule object is converted into ruleTagList, and a pointer i pointing to ruleTagList is set and initialized as 0, i.e., pointing to the first element of ruleTagList.

(8) The node information on the first tag node curNode of tagNodeList is matched with ruleTagList[i] using tagNodeList in 3.3.2 and ruleTagList in Step (7). If matching fails, turn to Step (10); if matching succeeds, curNode joins in pathStack, and all child nodes of curNode enqueues in childQueue; next, an element is dequeued from childQueue, a value is assigned to curNode, and ++i, meaning that the pointer pointing to ruleTagList shifts forward by one position.

(9) The node information on curNode is matched with ruleTagList[i]: 1) if matching succeeds, curNode joins in pathStack, childQueue is emptied, and all child nodes of curNode re-enqueue in childQueue; 2) if matching fails, a node element is unqueued from childQueue to assign a value to curNode, which matches with ruleTagList[i] once again, and if this succeeds, Step 1) is repeated; if the matching fails, Step 2) is repeated; 3) if all child nodes fail in matching, a node element is shifted out of pathStack, and a value is assigned to curNode, namely, –i, childQueue is emptied, all matched child nodes of curNode are enqueued in childQueue, and Step (9) is repeated; 4) if all ruleTagList[i] are matched, the matching succeeds, all text information targetString under curNode is taken out to form <PT,targetString> pairs, which are stored in structuredDoc, and then turn to Step (5), the information of the next “target extraction item” is extracted. If the information of all target extraction items is extracted, turn to Step (12); 5) if a ruleTagList[i] fails in matching, or the child nodes of one curNode are empty and ruleTagList is not completely traversed, turn to Step (10); If pathStack is empty or the tag path composed of the nodes therein is different from ruleTagList, turn to Step (10).

(10) If extraction (matching) is incorrect, “one is added to the extraction error counter” to calculate the extraction error rate W. If W is greater than or equal to the preset threshold TH, turn to Step (13), or otherwise, turn to Step (1).

(11) If the working state of the web crawler is “off”, exit from the whole webpage information extraction system; if the working state of the web crawler is “on”, turn to Step (1), followed by the next round of webpage crawling.

(12) Turn to Step (2) for information extraction for the webpages crawled in this round but not subjected to information extraction (the web crawler crawls multiple webpages in each ground under the multi-threading mode); if all webpages are extracted, turn to Step (1).

(13) The specific extraction rules are regenerated according to the generation and calibration method for the extraction rule template of “target extraction items”.

From the perspectives of system implementation and application, the detailed process of visual webpage text information extraction is depicted through the following pseudo-code, as presented in Algorithm 1.

4. Experimental results and analysis.

4.1. Experimental environment. To assess the performance and efficiency of the “visual webpage information crawling” system within a specific hardware and software environment, a detailed experimental scheme was devised to evaluate the system’s real-world operational efficacy. The hardware setup for the experiment comprised a computer equipped with an AMD Ryzen 5 PRO 2500U processor, featuring a Radeon Vega Mobile GFX integrated graphics card and a base clock rate of 2.00 GHz, alongside 8GB of RAM. On the software front, the experiment utilized Apache Nutch, a highly extendable, open-source webpage crawler software designed for harvesting original webpage data from the Internet. This combination of tools enabled the effective processing of Chinese texts and facilitated the extraction of information from a broad array of online resources.
Algorithm 1 parseToFormStrDoc(String pageURL, int siteId, String pageSource)

Input: Web Page URL pageURL, website ID siteId, webpage source pageSource
Output: structed document structedDoc

1: Map<Integer, StringBuffer>structedDoc; // the extracted structured documents.
2: // obtaining a url template matching the web page URL based on the webpage
3: // and its associated site id
4: URLTemplate urlTemplate = getMatchURLTemplate(pageURL, siteId);
5: // acquiring the webpage parsing template based on the URL template
6: ParseTemplate parseTemplate = getParseTemplateByURLTemplate(urlTemplate);
7: // obtaining the webpage extraction template and all extraction rules
8: List<ParseRule>ruleList = getParseRulesByParseTemplate(parseTemplate);
9: // categorizing by ‘extraction target items’ to form sets of extraction rules for each
10: // respective extraction target item
11: for parseRule in ruleList do
12:  List<ParseRule>itemRuleList; // Extraction rule list.
13:  utilizing the extraction rule set ‘itemruleList’ to extract text information corresponding to the ‘extraction target
14:  item’;
15:  for itemRule in itemRuleList do
16:   StringBuffer targetString;
17:   // separating ‘itemRule’ (Label Path) and storing it as an array of strings
18:   itemParseRule = splitToStrList(itemRule); // HTML text parse rules.
19:   int ruleListLength = itemParseRule.size();
20:   int ruleListIndex = 0;
21:   // utilizing parsing rules for the actual analysis of web pages
22:   NodeList childList; // The child node list of a specified node in a webpage.
23:   if (matching the root element of tagNodeList with itemParseRule[ruleListIndex]) then
24:     childList = rootTagNode.getChildren();
25:     ++ruleListIndex;
26:   else
27:     Terminating the Information Extraction for the Specified ‘Extraction Target Item’;
28:   end if
29: end for
30: while childList not empty do
31:  int childIndex = 0;
32:  for childIndex <childList.size() do
33:   Node tmpNode = childList.elementAt(childIndex);
34:   if tmpNode instanceof TagNode then
35:     TagNode curTagNode = (TagNode) tmpNode;
36:     tag1 = curTagNode.getText(); // Extract the text information from a webpage node.
37:     tag2 = ruleList.get(ruleListIndex);
38:     if tag1 == tag2 then
39:       if ruleListIndex != (ruleListLength - 1) then
40:         childList = curTagNode.getChildren(); // Get child nodes of the current node.
41:         ++ruleListIndex;
42:         break;
43:       else if complete matching of parsing rules then
44:         NodeList curChilds = curTagNode.getChildren();
45:         for (int j = 0; j <curChilds.size(); ++j) do
46:           targetString+=Text Information of Each ‘Child Tag’;
47:         end for
48:         end if
49:       structedDoc.put(extraction of target item number, extraction of target item text information);
50:     end if
51:   end if
52: end for
53: end while
54: if (several nodes in the extraction rule do not match) then
55:  terminating the information extraction for the specified ‘Extraction Target Item’;
56: end if
57: end for
58: return structedDoc;
4.2. Evaluation indexes. To thoroughly assess the performance of the “visual webpage information extraction” system, this study adopted three evaluation metrics commonly utilized in the domain of text information processing: accuracy, recall, and F-measure, ensuring a broad applicability of our findings. These metrics are extensively recognized for their effectiveness in gauging the performance of diverse information processing systems, offering a holistic perspective on the efficacy of the visual webpage information extraction system.

Accuracy, a metric quantifying the system’s proficiency in extracting accurate information, is defined by Equation 4.1 as the ratio of correctly extracted information items to the total number of extraction attempts. This index directly mirrors the precision of the information extracted by the system, providing a clear indication of its reliability.

\[
\text{Accuracy} = \frac{TP}{TP + FP} \quad (4.1)
\]

where \(TP\) (True Positive) and \(FP\) (False Positive) represent the number of relevant information items correctly extracted and the number of nonrelevant information items wrongly extracted, respectively.

Recall assesses the system’s capability to retrieve all pertinent information, as delineated in Equation 4.2. Specifically, it represents the ratio of relevant information items accurately identified by the system to the entire set of relevant information items. The significance of recall lies in its focus on the system’s comprehensiveness, highlighting its ability to capture essential information without omissions.

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (4.2)
\]

where \(FN\) (False Negative) stands for the number of unextracted relevant information items.

In summary, accuracy serves as a comprehensive indicator of the model’s performance in all detections, whereas recall specifically zeroes in on the model’s proficiency in accurately identifying true positives among all actual positives. These metrics are pivotal for assessing and benchmarking model performance, especially in domains where distinguishing between various error types is critically important.

Accuracy and recall exhibit a dependent relationship, with an ideal scenario featuring high values for both. Generally, a high accuracy often corresponds with lower recall, and vice versa. Should both metrics register low, it indicates a fundamental issue with the model. Consequently, the F-measure, defined as the harmonic mean of accuracy and recall, emerges as a crucial metric for gauging overall system performance, as encapsulated in Equation 4.3. By integrating both accuracy and recall, the F-measure offers a singular metric to appraise the comprehensive performance of the system.

\[
F\text{-measure} = 2 \cdot \frac{\text{Accuracy} \cdot \text{Recall}}{\text{Accuracy} + \text{Recall}} \quad (4.3)
\]

The efficacy of the “visual webpage information extraction” system across different dimensions can be thoroughly assessed by a composite evaluation of the aforementioned metrics. Employing these evaluation indices not only enables a detailed performance analysis of the system but also lays the groundwork for subsequent system enhancements.

4.3. Analysis of data and experimental results. In this research, we conducted practical application tests focusing on specific sites for webpage crawling and information extraction from designated websites. It’s crucial to emphasize that, aligned with the application demands, this study leveraged the visual webpage information extraction framework and methodology introduced herein to target and retrieve only the essential information from webpages—namely, the titles and the core textual content—for the purpose of performance evaluation. Other data categories, including comments and links, were also amenable to processing via the outlined extraction techniques in this research. The websites selected for analysis in this study were primarily specialized enterprise portals, characterized by a limited number of site modules and a uniform, straightforward webpage structure across these modules, featuring minimal noise data. Given that the webpage information on these sites predominantly consisted of webpage titles and texts, the extraction rule templates were specifically
Table 4.1: Test results for accurate extraction of webpage text information.

<table>
<thead>
<tr>
<th>Site Name</th>
<th>#webpages</th>
<th>Title</th>
<th>Main Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Accuracy</td>
<td>F-measure</td>
</tr>
<tr>
<td>Yunnan Tobacco</td>
<td>1650</td>
<td>99.27%</td>
<td>99.83%</td>
</tr>
<tr>
<td>Yunnan Tin Group</td>
<td>2001</td>
<td>99.37%</td>
<td>99.52%</td>
</tr>
<tr>
<td>DIHON</td>
<td>1076</td>
<td>98.63%</td>
<td>90.49%</td>
</tr>
<tr>
<td>Yunnan Copper Group</td>
<td>2439</td>
<td>95.32%</td>
<td>92.66%</td>
</tr>
<tr>
<td>Yunnan Ruisheng Pharmaceuticals</td>
<td>3083</td>
<td>93.26%</td>
<td>93.27%</td>
</tr>
</tbody>
</table>

The above comprehensive analysis not only highlights the specific challenges faced in webpage information extraction but also sets the stage for future improvements in extraction technologies and methodologies, aiming to enhance both accuracy and efficiency in competitive intelligence systems.

Following its deployment into trial operation, the system has elicited favorable feedback from its users, underscoring its user-friendly interface, ease of operation, and transparent procedural flow. Notably, it has demonstrated remarkable operability, particularly for non-professional users, achieving high levels of extraction accuracy and recall that align well with practical demands. However, traditional systems often present a steep learning curve and may compromise on either user-friendliness or technical robustness—challenges that this system adeptly navigates. Its capacity to combine simplicity in design with sophisticated functionality...
Zhiwei Zhang, Wenbo Qin, Haifeng Xu addresses a crucial gap in the field, where the balance between accessibility for novice users and meeting the advanced needs of professional scenarios has frequently been elusive.

The analysis of experimental outcomes not only deepens our comprehension of the system’s operational efficacy but also accentuates its practical utility and appeal across user demographics. This nuanced understanding is pivotal, as it transcends mere operational success to underscore the system’s alignment with user-centric design principles—a facet often overlooked in the pursuit of technical excellence. This reflection prompts a critical dialogue within the realm of webpage information extraction systems, advocating for a paradigm where user engagement and technical precision coalesce. The insights gleaned from this trial phase, therefore, not only spotlight the system’s current achievements but also chart a course for future enhancements. Emphasizing user feedback in the iterative process of system refinement offers a roadmap for evolving beyond the constraints of traditional methodologies, ensuring continued relevance and utility in an ever-changing digital landscape.

In sum, this analysis not only validates the system’s effectiveness and user satisfaction but also serves as a foundational critique against which the limitations of existing methods are measured. It lays the groundwork for ongoing innovation, encouraging a holistic approach to system design that prioritizes both user experience and technical rigor.

5. Conclusion. This research primarily sheds light on the pivotal role of webpage information extraction in the competitive intelligence domain. Specifically, it introduces and develops a visual webpage information extraction module within a competitive intelligence system, bridging the gap between technological research and practical application. Initially, the article sets out to define the objectives and requirements essential for webpage information extraction, critically assessing the strengths and weaknesses of current theories and methodologies related to webpage text information extraction. This critical analysis lays the groundwork for the introduction of an innovative approach to visual webpage text information extraction. Additionally, the study elaborates on the comprehensive framework of this module, delving into the nuances of the extraction template, rule generation, optimization method, and extraction algorithm integral to the visual webpage information extraction process. This thorough exploration ensures a deep understanding of the complexities associated with the development and operationalization of the visual webpage information extraction module within the competitive intelligence framework.

This study has made a significant contribution to the field of competitive intelligence by proposing and implementing an efficient visual webpage information extraction system. The system encompasses three key subsystems: a webpage information crawling subsystem, a visual webpage information extraction rule template generation subsystem, and a webpage text information extraction subsystem. A notable innovation within this study is the generation of the visual webpage information extraction rule template, which offers new tools and methods for the domain. The system’s efficacy and practicality have been demonstrated through extraction tests on specific themed portal websites, evidenced by the analysis of experimental results.

Future research directions stemming from this work should focus on several key areas. Firstly, enhancing the robustness and adaptability of the visual webpage information extraction rule template is paramount. This could involve integrating machine learning techniques to enable the system to adapt to dynamically changing webpage structures and content. Secondly, expanding the system’s application to a broader range of web resources, including dynamic and multimedia content, will significantly extend its utility. This could involve the development of advanced algorithms capable of handling various media formats and interactive web elements.

Another promising avenue is the exploration of deeper semantic analysis and context understanding in the extracted information. Employing natural language processing and semantic web technologies could provide more nuanced insights, particularly in fields like sentiment analysis and trend prediction. Additionally, integrating the system with big data analytics tools could offer comprehensive competitive intelligence solutions, capable of processing vast amounts of web data to derive strategic insights.

Finally, considering the ethical and privacy aspects of web data extraction is crucial. Future work should include developing guidelines and protocols to ensure compliance with data protection regulations and ethical standards. This will not only ensure the legal use of the technology but also enhance its acceptance and trustworthiness among users.

In conclusion, while this study lays a strong foundation, these future research directions offer avenues for further enhancement and application of the visual webpage information extraction system, ensuring its continued
relevance and effectiveness in the evolving landscape of competitive intelligence and web data analytics.

Acknowledgments. This research received support from several grants, including the Natural Science Foundation of Anhui Province under Grant No. 1908085QF283, the Doctoral Startup Research Fund with Grant Nos. 2019J008 and 2023bk024, the University Synergy Innovation Program of Anhui Province with Grant No. GXXT-2022-047, the Open Research Fund of the National Engineering Research Center for Agro-Ecological Big Data Analysis & Application at Anhui University under Grant No. AE202201, and the Natural Science Research Projects in Universities under Grant No. 2023AH040314. This work was also supported by the Scientific Research Projects Funded by Suzhou University under Grant No. 2021XJPT50, and the Excellent Young Teacher Training Program under Grant No. YQYB2023053.

REFERENCES


Edited by: Jingsha He

Special issue on: Efficient Scalable Computing based on IoT and Cloud Computing

Received: Dec 26, 2023
Accepted: Mar 12, 2024