International Journal of Knowledge and Language Processing
Volume 12, Number 2, Dec **2021**

AKLP International ©2021 ISSN 2191-2734
pp. 1—16

Review of Self-Adaptive Web Crawler Research Based on Comprehension of Webpage HTML

Ru Liu and Yu Zhai

Peking University
No.5, Yiheyuan Road, Haidian District, Beijing, China
1801210589@pku.edu.cn

Received October 2021; revised October 2021

ABSTRACT. *The technology of obtaining network resources is of great significance for the intelligence research fields such as policy research, public opinion analysis and news briefing push. But the government sites, think tank research sites, and business news sites are often different from personal websites, reflecting a more stringent and complicated data organization layout. Their source codes are more detailed and updated in the website maintenance, along with those frequently occurring website redesign phenomena, the web information extraction task for crawler technology is now facing great challenges. Self-adaptive crawlers based on comprehension of webpage HTML understanding can adaptively adjust crawler logic and rules according to environmental changes and improve resource acquisition efficiency. This paper illuminated the research contents and development of software adaptability, web page source code representation, and crawler source code representation, and pointed out the technical difficulties of self-adaptive crawler research and the existing problems in related research. The purpose of this paper is to provide ideas and suggestions for the subsequent research of self-adaptive crawler generation and the improvement of adaptive crawler workflow from the perspective of structured source code extraction and similarity calculation, combined with the analysis of problems existing in adaptive crawler code generation research.*

Keywords: Webpage Shift, HTML Source Comprehension, Generation of Self-Adaptive Web Crawler

1. Introduction

Network resource has a great deal of valuable information such as scholarly information and social information, which contain abundant explicit or tacit knowledge. The surge of big data has accelerated the explosive growth of network resources, leading to more diverse ways of classification, aggregation, organization, and display forms of network resources, which has brought great challenges to the automatic network

information collection and data mining technology.

In Internet application services, the technology of automatic collection and analysis of network information generally refers to crawler technology. According to the general process of crawler, crawler business can be divided into four stages: "communication-download-analysis-warehousing". The existing open-source crawler framework mainly focuses on network communication and web page download based on the HTTP request, as well as automatic extraction and analysis of target data in web pages. There are many problems in the existing research on crawler technology. Firstly, the current research on extracting target data from web pages is mostly oriented to a single context, which is difficult to adapt to complex and diverse crawler business scenarios. Secondly, webpage shifting frequently occurs, which leads to the code failure of webpage information extraction module, but there is little research on crawler technology aiming at the change of webpage source code. Finally, the interruption of crawler service caused by webpage shifting often requires a lot of human resources to repair. Self-adaptive crawler technology can improve the stability and adaptive repairability of the crawler system. However, the existing research lacks a summary of the characteristics of webpage source code changes and ignores the relationship between webpage source code and crawler adaptability. Therefore, starting from the self-adaptive crawler theories based on source code understanding, this paper illustrated the present situation and existing problems of related research from three aspects: self-adaptive system, webpage information extraction, and crawler source code representation. This paper also introduced the technical difficulties and existing problems of the self-adaptive crawler based on source code understanding, intending to provide ideas and suggestions for highly automated webpage resource acquisition.

2. **Research status of self-adaptive crawlers**

Crawlers can in a broad sense refer to automatic tools based on the understanding of web page source code and network communication. They collect webpage information based on the topological network composed of web page link nodes through certain strategies. In a narrow sense, it can also refer to the web page information extraction technology for a single web page.

The core of the self-adaptive crawler system is a service model based on depth perception and automatic discrimination. By constructing intelligent tools to regularly monitor the internal and external environment of crawler applications, the interaction and collaborative calculation of web source code and crawler code are realized, and the behavior logic and organization mode of crawlers are dynamically adjusted according to web response, crawler log and self-change types, so that crawler applications can continuously obtain web information when facing a new web environment [9]. In the crawler running environment, there are often a large number of interactive information such as web page response, crawler log, storage queue, etc. They construct the crawler's supervision and feedback system on web page changes. But with the uncertain factors brought by the dynamic environment, the traditional preset rules are difficult to handle all

the problems faced by the self-adaptive crawler. Therefore, the logical rules of the crawler should be adaptively adjusted according to the environmental changes to achieve better robustness and provide stable service output. For the application of self-adaptive crawler, it is defined as having the following three abilities: (1) Combining the crawler's perception information of the webpage source code, extracting and abstracting features from the webpage source code, and dynamically perceiving the changes of the webpage source code; (2) According to the interactive data generated by crawler application, through knowledge map and data mining technology, the source code change types are analyzed and deduced; (3) To complete the adaptive decision adjustment in the changing webpage environment through logical calculation, deductive reasoning, and planning model, along with perceptual analysis and objective reasoning.

2.1 **Research on software adaptability**

Software self-adaptation or self-adaptive system have the same definition except for a slight difference in emphasis: they all mean that the system can autonomously adjust its behavior according to its perception of the external environment and internal system when running, in order to effectively cope with the dynamically changing environment and uncertain demands, thus enabling the system to have better flexibility, reliability, and robustness [9]. The rule-based "condition-action" response mechanism is the simplest reflection of that. With the classical cybernetics [11-13], modern cybernetics [14], intelligent cybernetics [15] and other theories put forward, the adaptive software architecture has also expanded. A series of adaptive system models based on feedback control loops have come out. IBM put forward the famous adaptive MAPE-K ring theory, which divides the adaptive process into four stages: monitoring, analysis, planning and execution, and serves as an auxiliary knowledge base. The core of an adaptive system is to construct adaptive strategies [16]. Based on the concept of the adaptive system, Wang Yang et al. put forward a data-driven application adaptive method in the environment of man-machine-object integration and advocated constructing a knowledge map with presentation knowledge base and process decision base as the core through reasoning and cognitive calculation such as process mining and pattern matching [18-19], so as to realize scene-oriented adaptation and self-evolution. Zhang Mingyue [20] introduced the method of association and combination between software adaptive task and machine learning model algorithm from the perspective of machine learning. However, mature research and application examples for adaptive software systems are rare, especially the research combining natural language processing with deep learning methods is even rarer.

As for the research on the adaptability of crawler software, some papers [20-26] with the title of self-adaptation have not provided the perspective of the self-adaptive system. They still lingering at the research stage of putting forward the automatic implementation method for realizing one certain part of crawler, such as automatic extraction of webpage information or automatic downloading of webpages, etc. In addition, most of these studies focus on the topic crawler, which mainly discusses the crawler's ability to judge the topic of webpage content independently when collecting webpages. Therefore, the existing research needs to learn from the core idea of the self-adaptive system and discuss the adaptive

scenarios faced by crawlers.

2.2 **Crawler research for web page changes**

In the related research of crawler technology, the crawler research on webpage shift is very few, but some methods of web page entity extraction research have some reference value for this thesis.

Joseph[21] proposed an Xpath generation method from the perspective of subtree matching for web page changes. Choudhary[22] studied the changes of web page source code from the perspective of automatic testing of web applications and conducted correlation research on the source code before and after the changes. Concerning the invalidation of information extraction code caused by HTML page changes, Choudhary et al. divided the unavailability of Xpath information extraction code into three types: 1) The addition, deletion, and modification of DOM tree structure of web source code led to the no selection or wrong selection; 2) Abnormal data content leads to invalid Xpath selection; 3) Implicit changes that the client cannot find. Except that, other self-adaptive crawler research studies how to automatically extract data when facing new web pages by suggesting how to automatically generate wrappers or extract data, which is quite different from the subject of this paper.

Entity extraction, also known as web Information Extraction (IE), was put forward by VALTER et al. as early as 2004 [20-28], which is a process of extracting the target information needed by users from semi-structured web data, reorganizing its form and forming structured data. With the development of search engines, the demand for web information collection and extraction technology is increasing. The technology now has become one of the most important parts of crawler technology, alongside web communication download technology. The development of machine learning and deep learning also brings inspiration to automatic information extraction technology, and the concepts of Wrapper Induction (WI) [29-31] and Vision-based Page Segmentation (VIPS)[32,4] are derived from webpage information extraction. On the one hand, according to the classification of target entities, information extraction methods for data features of target entities are proposed, such as webpage date extraction method [36-38], text extraction method [39-41], etc. On the other hand, web page information extraction takes methodology as the starting point and focuses on heuristic rules-based methods, HTML structural templates-based methods, statistics-based and machine learning-based methods, and web page segmentation methods with machine vision empowerment [32-33].

In the research of webpage text extraction, most scholars abstract the text features of the text in HTML source code, and combine the rule-based or machine-learning methods to extract the text. Among them, Miao Lin [42] proposed a top-down text information extraction method for Web pages, which is adaptive to a certain extent according to the characteristics of each node, their text length, text link rate, and other data information, rather than a specific Web page template. Lv Rongzheng [41] proposed a text extraction method based on leaf node classification and calculation of the highest signal-to-noise ratio. Pan Xinyu [40] proposed a text extraction method based on node path similarity of DOM tree of web pages, and Liu Zhiqiang [43] proposed that the distribution of information to be

extracted in web documents has the characteristics of order relationship, for example, the key information of web news, such as title, date, source, and text, traversed in a different order in DOM tree. Therefore, it is considered that the timing and strong modeling ability of Hidden Markov Model(HMM) is suitable for the task of extracting key information from web news.

Most of the research on date extraction transforms this task into a named entity recognition task, which is mainly divided into two methods: rule-based method and machine learning method, which is transformed into classification or sequence labeling problem. As the time information in the web page can be divided into two categories as a whole: the publication time of the report and the event time contained in the report [37]. The former is generally located below the news headlines and can be divided into absolute time and relative time. Absolute time refers to the specific time when the report is published. Relative time refers to the time expressed in natural language relative to the current system. As for the research of date text and structural features of web pages, the current research includes the textual features such as part of speech, position, grammar, and the semantic role of date, and the boundary calculation of date and time based on machine learning.

Although the automatic extraction method of webpage information based on machine learning has made some progress, the mainstream and engineering methods still mainly focus on Xpath path expression templates, like manually writing Xpath or CSS selectors corresponding to target information in HTML according to website version to extract information. Therefore, more methodologies have emerged in the research of an automatic generation of Xpath. For example, Joseph[20] proposed the method of matching DOM tree subtrees to automatically generate Xpath code; Due to the problem of Xpath writing in automatic testing of Web applications, Maurizio[23] proposed a rule-based method for generating XPath locators, Robula+; Oliver et al. [24] proposed a ranking mechanism based on the degree of association between Xpath generated by rules and search terms. Wu Gongqing [44], a domestic scholar, also studied Xpath, analyzed the source code of web pages and its DOM tree structure, summarized the distribution characteristics of text entities in the DOM tree, and proposed a method of extracting web page text to distinguish noises.

In the research of crawler technology using machine learning or deep learning, some scholars take advantage of the strong structural features of the HTML source code of webpages and use graph theory in deep learning to express learning of webpage source code. For example, Gogar[45] proposed a DCNN image classifier based on the convolutional neural network(CNN), which combined with machine vision to extract webpage information. Tan[6] proposed that the links in web pages should be constructed into a network, and the hyperlink text features of phishing web pages should be extracted to establish a random forest classifier for phishing detection. By comparing the classifiers such as support vector machine and Naive Bayes, it was verified that the random forest algorithm had the best effect in the graph model classification task.

2.3 **Research on Source code representation**

In the aspect of semantic representation of code, the research on the representation of program language carries out structured text processing through identifier processing, API processing, and abstract syntax tree analysis on the one hand, and carries out semantic representation of code based on machine learning algorithm on the other hand. For example, combining vector space model to represent program language fragments and natural language texts in the same semantic space, and then carrying out subsequent feature extraction and similarity calculation for code recommendation task. Faced with the problem that the semantic space of VSM is too sparse, some scholars express code fragments by combining implicit semantic analysis and hidden Dirichlet distribution to enhance the matching ability between natural language and programming language [46].

Code2vector method combines abstract syntax tree to represent code fragments. After sampling the path of the abstract syntax tree of the source code, it is input as a sentence into the depth model for representation. The method of code2seq[48] is similar to it, and its main idea is to represent the code text through LSTM long-short memory network and represent the code fragment by serialization vector. The word2API proposed by He Jiang et al. [49] uses word embedding technology to jointly model phrases in natural language and APIs in the programming language to solve the problem of word mismatch in the process of cross-language matching. The model is verified by combining retrieval tasks, which has some reference value for the research of modeling combining natural language and programming language.

Up to now, word2vec, doc2vec, or LSTM-based sequential network models are mostly used for code representation learning. However, the semantic representation methods of these codes usually only use the shallow semantic features of the code but fail to combine the strong structural features of the code itself, such as the association characteristics including the interaction of methods, functions and variables in the code, and the dependency call. Therefore, by constructing the graph model of the code, we can use the network representation learning model in the research of the graph neural network(GNN) to study the HTML source code of the webpage, the code of the crawler, Xpath and other structural expressions.

In the aspect of network representation learning algorithm, Chen Bin [46] uses multiple neighborhood masks to learn multiple node representations from different order neighbors of nodes based on attention mechanism and adopts dynamic routing algorithm to adaptively calculate the contribution of these initial representations to the final node representation, thus obtaining the node representation aggregated by the two. Reference [51-52] puts forward different improvement mechanisms in the network representation learning method, among which PGE model uses node clustering to distribute deviation to distinguish the neighbors of nodes, and uses neighbor-based biased sampling mechanism to fuse more attribute information, which is 10 percentage points higher than the baseline model of DeepWalk and 3 percentage points higher than GCN network. Reference [50] puts forward a Graph-to-Graph algorithm for modeling the mapping relationship between graph models.

Based on source code or code representation, many researchers have further explored

the downstream machine learning tasks. In the aspect of similarity measurement, Liu Wenbin [53] proposed a sentence alignment method based on multi-similarity fusion, which fused three features, namely BLEU score, cosine similarity, and Manhattan distance, for semantic representation. Liang Hongxiang [54] proposed a similar case recommendation method based on network representation learning, which was applied to the Text CNN model to complete the task of text classification.

3. **Technical difficulties and existing problems of self-adaptive crawler generation**

To deal with the crawler failure caused by website style revision in the process of obtaining network resources, it is necessary to analyze the web source code and crawler code in a structured way and process the deep semantics of the resources. Based on a deep understanding of the two resources, the source code and crawler code, we can combine the features of the source code to determine the error types of webpage shifting, thus effectively guiding the adaptive generation of crawler code. There are two technical difficulties involved:

(1) It is necessary to establish an adaptive perception and representation mechanism for changes in the source code of web pages. To realize the self-adaptive perception ability in the self-adaptive system, it is necessary to construct a representation model with generalization ability to meet the measurement requirements of source code changes of unknown web pages.

(2) It is necessary to build an effective code generation model. Transforming Xpath code generation into the classification of nodes in HTML source code, and transforming codes such as data analysis into code recommendation tasks depends on effective and robust algorithm modeling.

However, the adaptability of web page information extraction technology researched at home and abroad is not strong enough, and the research on self-adaptive crawlers often ignores the characteristics of webpage source code changes, which brings some obstacles to the subsequent code generation. Therefore, according to the existing problems in the research of self-adaptive crawler technology, this section will introduce the advantages and disadvantages of related technologies and their existing problems from three aspects: source structured information processing, HTML tree structure similarity calculation, and self-adaptive crawler code generation, trying to provide ideas and suggestions for the follow-up research work.

3.1 **Structured source code information extraction technology**

The source code information is divided into web page structure information and crawler code structure information. In the current research at home and abroad, there are several types of web page information extraction methods: information extraction methods based on wrappers and heuristic rules, extraction methods based on text features in web pages [1-2], information extraction methods based on visual blocks [3-4], and extraction methods based on statistics and machine learning. These methods have high accuracy for specific types of webpage information extraction tasks, but when faced with complex and diverse crawler business objectives, these methods facing a single context show poor

adaptability and generalization ability, and cannot meet the requirements of highly automated webpage information collection and analysis. Using more advanced network representation learning methods and graph model node classification algorithms, deep learning based on web source code and code network construction can better represent and count the structure and semantic information of source code and lay a good foundation for adaptive generation of crawler code. Therefore, this section will start from the network representation learning and node classification algorithms, and then elaborate problems of existing algorithms and suggestions for further research.

Network representation learning, also known as graph representation learning or graph embedding algorithm, refers to a technology of representing the data in mesh or tree structure and transforming it into vectors to be applied to downstream tasks. Usually, the downstream tasks include the classification and clustering of subgraphs or nodes and the mining of node paths in graphs. The core of the graph embedding algorithm is to capture the relationship between nodes by traversing the nodes in the graph model and to transform this topological structure into a mathematical expression, and then use skip-gram or softmax model to build a probability model. Therefore, the emphasis of graph embedding algorithm may include three aspects: semantic information of nodes, semantic information of node relations, and traversal mode of topology. However, most of the existing researches focus on the latter two, which largely ignores the rich semantic information contained in the nodes and edges represented by attribute graphs, which is a major defect for natural language processing applications.

Hou et al. proposed a graph representation learning framework "PGE", which incorporated the attributes of nodes and edges into the graph embedding process. PGE uses node clustering to distribute deviation to distinguish neighbors of nodes and uses multiple data-driven matrices to aggregate attributes of neighbors sampled based on deviation strategy.

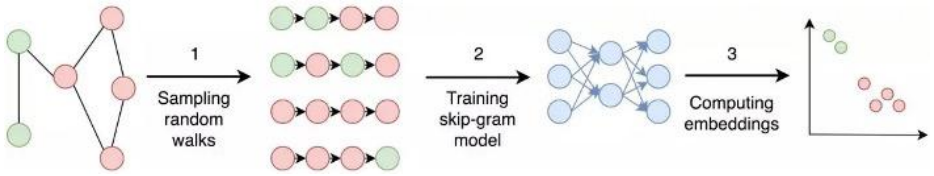

FIGURE 1: PRINCIPLE OF NETWORK REPRESENTATION LEARNING MODEL.

Node classification and link prediction are the most common tasks in graph model applications. The node classifier is designed to classify tags in HTML source code and take the target fields obtained by crawlers as classification objects. For example, the target objects of the classifier can be set as title tags, text tags, date tags, metadata tags, frame structure tags, multimedia tags, advertisement tags, etc. Most researchers use traditional machine learning classification methods, such as support vector machines (SVM), logistic regression, and other probabilistic classification methods, to test the effectiveness of network representation learning methods. However, the purpose of the research on self-adaptive crawlers to classify element nodes is to lay the foundation for the subsequent

task of identifying difference tags and matching and parsing code. The subject to be classified is an element tag in the source code along with its attributes and text, which is usually classified as a short text classification problem. On the other hand, considering that the graph representation learning algorithm mentioned above can train the DOM tree vector, the node classification algorithm in the graph neural network is more suitable for the research task of this paper.

Semi-supervised learning for node-level classification usually refers to a single network in which some of the specified nodes are labeled. GNN can learn a robust model and efficiently identify the category labels of unlabeled nodes [14]. Before Xgboost was put forward, the most famous tool of GBDT was Xgboost, which was a decision tree algorithm based on the pre-ranking method. The basic idea of this decision tree algorithm is as follows: firstly, all features are pre-sorted according to their numerical values; Secondly, when traversing the segmentation points, we find the best segmentation point in features. Finally, according to this segmentation point, we split the data into left and right sub-nodes, and complete the task of node classification.

Existing network representation learning models mainly focus on the feature representation of nodes. Most models represent the information of nodes, but ignore the label information of nodes. Therefore we can combine DeepWalk with Max-Margin in terms of the loss function as MMDW, while in training, it optimizes disambiguation and representation at the same time to achieve better results. The main purpose of the PTE model is to show the predicted information in the last embedding layer, but instead of directly nesting a complex prediction model like CNN or RNN model, it defines three kinds of networks, word-word, word-document, and word-label. Then it summarizes their loss functions together, defines the empirical probability and the target probability to find KL distance to evaluate the advantages and disadvantages of network representation. TriDNR models the correspondence between labels and words by maximizing the probability of a sequence of words labeled with a specified category. Word2vec or Doc2Vec model learns paragraph and document embedding through distributed memory and distributed word bag model and extracts contextual features to characterize text paragraphs. As the HTML documents in the research pages of this paper are a manually defined markup language, except for the body of the HTML document, they have non-textual characteristics such as conciseness, multiple abbreviations, and multiple symbols. Therefore, efficient algorithmic models need to be explored in subsequent research so as to complete the learning of HTML document structure features and text features of web pages as well as the classification and prediction of tag nodes in HTML to support subsequent research work on adaptive crawler code generation.

3.2 **Similarity calculation of HTML tree structure**

HTML tree structure, as the final display form of web page layout and style structure, can approximate the similarity and difference of web page structure by similarity calculation, thus improving the adaptability of web page parsing code and effectively dealing with the code recommendation problem of crawler failure caused by webpage shifting. Due to the limitation of the front-end template and framework of web pages, a

website usually contains a limited number of web page styles, but the structure of web pages varies widely among the massive Internet pages. Therefore, it is necessary to effectively represent the structural features of web pages and adopt appropriate structural similarity measurement methods.

The premise of effectively characterizing the structural features of web pages is to simplify the source code structure. Reference [15] puts forward the concept of the same structure and regards nodes with the same Xpath as the same structure as one of the features of webpage information extraction. In the reference [28], the similarity of brother nodes is also calculated to determine whether the group of brother nodes belong to the same target data type. However, these studies start with horizontal sibling nodes with similar structures, which contain more redundant information, so they cannot simplify the DOM tree structure from a higher dimension, and improve the accuracy of semantic calculation of the source code structure. Therefore, it is necessary to explore a more in-depth node concept to ensure that the DOM tree can be characterized as a vector by using an algorithm model with higher adaptability efficiently after the structure tree is constructed by the webpage source code, and the similarity between two HTML trees can be measured by the existing similarity measurement method.

Similarity measurement is a common method in document clustering and classification. Some classical similarity measurement methods are listed below.

Euclidean distance is the most commonly used distance calculation formula, which measures the absolute distance between points in multidimensional space. When the data is dense and continuous, it is a good calculation method. Because the calculation is based on the absolute values of the features of each dimension, Euclidean measurement needs to ensure that the indicators of each dimension are at the same scale level. For example, in KNN, features need to be normalized.

Levenshtein distance, also known as L's editing distance, refers to the minimum number of editing operations required to change from one string to another between two strings. Editing operations include addition, deletion, and modification, so generally speaking, the smaller the Levenshtein distance, the greater the similarity between two strings. Levenshtein distance is widely used in DNA analysis, pinyin error correction, named entity extraction, entity co-referencing, etc.

KL divergence (Kullback-Leibler Divergence), also known as relative entropy, is used to measure the difference between two probability distributions. In information theory, KL divergence can effectively measure the difference between two probability mass distributions, which has been widely used in informatics, statistics and physics [68]. KL divergence can be extended and applied to the difference measurement of other objects, such as vectors and matrices.

Cosine similarity is a measure of the distance between two vectors in mathematical space. The cosine of the angle between two vectors is used to measure the difference between individuals, and its calculation formula is as follows.

$$cos\theta = \frac{\sum_{i=1}^{n} x_i * y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}} \quad (1)$$

Compared with distance measurement, cosine similarity pays more attention to the difference between two vectors in the direction, rather than in the distance or length. Since the source codes of web pages belong to the text collections with different styles and different time tags in different websites, the web pages of different websites are quite different in structure, and the texts with close time intervals are more similar. Therefore, it is more appropriate to divide the data sets according to different websites and different time nodes, and the cosine similarity value is used to measure the vectorized data of word2vec and other models.

3.3 **Self-adaptive crawler code generation**

In chapter 2.3, the related technologies of crawler code representation have been summarized. However, the technology of code representation is only the key premise of self-adaptive crawler code generation, and more research work is needed to improve the code generation and recommendation mechanism and complete the "perception-decision-execution-evaluation" process to form a crawler system with adaptive processing capability.

Through long-term observation in the project practice, it is found that frequent webpage shifting events lead to the invalidation of webpage information extraction module code in crawler code. Because the code of the entity extraction module is mainly Xpath or CSS selector code, this part of the code reflects the location structure, text identification, and other information of the target entity in the webpage source code, which is the direct reflection of the target entity to be acquired by the crawler in the webpage source code. Therefore, when the website source code changes due to webpage update and maintenance, style revision, etc., the entity extraction module in the crawler code will be affected, resulting in a stop of the crawler business process, which is manifested by the interruption of the crawler thread and the failure to obtain target data. In a large-scale system platform, the interruption of the crawler process often needs a lot of human resources to repair and maintain, while the existing research seldom mentions the adaptive adjustment and repairability of the crawler system. Through searching a large number of papers, it is found that a small number of researches on "self-adaptive crawler" mainly focus on web page relevance and crawler access path in topic crawler [6-8]. On the one hand, these studies try to study from the angle of automatic webpage information extraction or automatic webpage collection, ignoring the relationship between the changing characteristics of webpage source code and the adaptive modification of crawler code; On the other hand, few studies focus on the frequent revision of webpages and lack the summary of the characteristics of the real event, namely, the change of webpage source code.

Therefore, in addition to constructing an efficient web page information extraction model and summarizing the method for calculating the similarity of source code structures, it is also necessary to explore and summarize the characteristics of web page source code changes and their types on the premise of accumulating a large amount of actual project

data, and to carry out subsequent research on the generation or recommendation mechanism of web crawler code and its quality assessment from the perspective of improving the generalization perception of web page source code changes in crawler systems and the ability to generate crawler code adaptively, thus complete the process of adaptive crawler generation.

4. **Conclusion**

Given the failure of crawler code caused by webpage shifting, this paper introduces the research and development status of related technologies and thinks that the parsing and processing process based on the understanding of source code structure can well support the subsequent research work of self-adaptive crawler generation. A complete self-adaptive crawler code generation system based on the understanding of source code structure can realize the complete workflow of adaptive error perception, code generation, and activation, thus effectively solving the real scene problem of network resource acquisition, improving the automatic processing ability of crawler technology in dealing with webpage shifting events, and promoting the highly automated process of webpage collection.

REFERENCES

[1] Zhou Y, Li J, Song Q. A Webpage Information Extraction Method Based on SVM and Te xt Density Features [J]. Computer Applications and Software,2020,36(10):251-255+261.

[2] Corchuelo Rafael, Jimenez Patricia. On learning web information extraction rules with TANGO[J].Information systems,2016,62(Dec.).

[3] Wang X, Guo Y, Liu Y, Yu X, Cheng X. Research on Web Information Extraction Method Based on Visual Features[J]. Journal of Chinese Information Processing, 2020, 33(05): 103-112.

[4] Wang W, Liang C, Min Y. Multi-record complex webpage information extraction algorithm based on visual blocks[J]. Computer Science, 2019, 46(10): 63-70.

[5] Martinez-Rodriguez JL, Lopez-Arevalo I, Rios-Alvarado AB. Mining information from sentences through Semantic Web data and Information Extraction tasks [J]. JOURNAL OF INFORMATION SCIENCE. 2020. DOI: 10.1177/ 0165551520934387.

[6] Tan CL, Chiew KL, Yong KSC. A graph-theoretic approach for the detection of phishing webpages. 2020[J]. COMPUTERS & SECURITY. DOI: 10.1016/j.cose.2020.101793.

[7] Halle, Sylvain, Le Breton, Gabriel, Maronnaud, Fabien, Masse, Alexandre Blondin, Gaboury, Sebastien. Exhaustive Exploration of Ajax Web Applications With Selective Jumping. 2014 [C]// 7th IEEE International Conference on Software Testing, Verification and Validation (ICST)

[8] Raj, S., Krishna, R., & Nayak, A. Distributed Component-Based Crawler for AJAX Applica -tions[C]. In Proceedings of 2018 2nd International Conference on Advances in Electronics, Computers and Communications, ICAECC 2018. 2018. [8479454] https://doi.org/10.1109/ICA

ECC.2018.8479454

[9] Wang Y, Dai H, Ren H. A preliminary study on data-driven application adaptation in a human-machine-object fusion environment[J]. Communications of the Chinese Computer Society. 2020.16(4):25-30.

[10] [10] Yang Q, Ma X, Xing J, Hu H, Wang P, Han D. Software adaptation: a method based on control theory[J]. Chinese Journal of Computers, 2016, 39(11): 2189-2215.

[11] Shen J, Wang Q, Mei H. Self-adaptive software: Cybernetic perspective and an application server supported framework.[C] In: Proc. of the Computer Software and Applications Conf. IEEE Computer Society, 2004. 92–95. DOI: 10.1109/CMPSAC.2004.1342684.

[12] Silva Souza VE, Lapouchnian A, Robinson WN, et al. Awareness requirements for adaptive systems.[C] In: Proc. of the 6th Int'l Symp. on Software Engineering for Adaptive and Self-managing Systems. Waikiki: ACM, 2011. 60–69.

[13] Karlsson M, Karamanolis C, Zhu X. Triage: Performance differentiation for storage systems using adaptive control.[C] ACM Trans. on Storage, 2005,1(4):457–480.

[14] Hu H, Jiang CH, Cai KY. Adaptive software testing in the context of an improved controlled Markov chain model.[C] Proc. of the 32nd Annual IEEE Int'l Computer Software and Applications Conf. IEEE, 2008. 853–858.

[15] Liu C, Jiang C, Hu H, et al. A control-based approach to balance services performance and security for adaptive service based systems (ASBS). In: Proc. of the 33rd Annual IEEE Int'l Computer Software and Applications Conf., Vol.2. IEEE, 2009. 473–478.

[16] Baah GK, Gray A, Harrold MJ. On-line anomaly detection of deployed software: A statistical machine learning approach.[C] Proc. of the 3rd Int'l Workshop on Software Quality Assurance. Portland: ACM, 2006. 70–77.

[17] Salehie M, Tahvildari L. Self-adaptive software: Landscape and research challenges[J]. ACM Transactions on Autonomous and Adaptive Systems, 2009, 4(2).

[18] De Leoni M, VandAWMP, Dees M. A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. [J] Information Systems, 2016, 56(MAR.):235-257.

[19] Fradkin, Dmitriy, Mörchen, Fabian. Mining sequential patterns for classification[J]. Knowledge & Information Systems, 2015, 45(3):731-749.

[20] Zhang M, Jin Z, Zhao H, Luo Y. Overview of Software Adaptability Empowered by Machine Learning[J]. Journal of Software, 2020, 31(08): 2404-2431

[21] Joseph Paul Cohen, Wei Ding, Abraham Bagherjeiran. XTreePath: A generalization of Xpath to handle real world structural variation. arXiv:1505.01303 [cs.IR]

[22] Choudhary SR, Zhao D, Versee H, Orso A. WATER:Web application TEst repair[C]. Proceedings of the 1st International. Workshop on End-to-End Test Script Engineering, ETSE 2011, ACM,

2011; 24–29, doi:10.1145/2002931.2002935.

[23] Maurizio Leotta, Andrea Stocco, Filippo Ricca, et la. ROBULA+: An Algorithm for Generating Robust Xpath Locators for Web Testing. Journal of Software: Evolution and Process, Volume 28, Issue

[24] Oliver Jundt, Maurice Van Keulen. Sample-based Xpath Ranking for Web Information Extr action.[J] Advances in Intelligent Systems Research. 2013, https://doi.org/10.2991/eusflat.201 3.27.

[25] Edwards, J., McCurley, K. S., and Tomlin, J. A. (2001). An adaptive model for optimizing performance of an incremental web crawler. In Proceedings of the Tenth Conference on World Wide Web (Hong Kong: Elsevier Science): 106–113. doi:10.1145/371920.371960.

[26] Kumar Sharma D, Khan M A. SAFSB：A self-adaptive focused crawler, 2015[C]//IEEE, 2015.

[27] VALTER CRESCENZI, GIANSALVATORE MECCA. Automatic Information Extraction from Large Websites. Journal of the ACM, Vol. 51, No. 5, 2004, pp. 731–779.

[28] Oren Etzioni, Michael Cafarella, Doug Downey, et la. 2004. Web-scale information extraction in knowitall: (preliminary results). Proceedings of the 13th international conference on World Wide Web. Association for Computing Machinery, 100–110.DOI: https://doi.org/10.1145/988672.988687

[29] Mei X, Cheng X, Guo Y, Zhang G, Ding G. A method of automatically generating webpage information extraction wrapper[J]. Journal of Chinese Information Processing, 2008(01): 22-29.

[30] Liu L, Pu C, Han W. XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources[C]//Proceedings of the 16th International Conference on Data Engineering. IEEE, 2000: 611-621.

[31] Joseph Paul Cohen, Wei Ding, Abraham Bagherjeiran. Semi-Supervised Web Wrapper Repair via Recursive Tree Matching. arXiv:1505.01303 [cs.IR]

[32] Song X, Zhao T. Analysis and comparison of network information extraction technology[J]. Intelligent Computers and Applications, 2013, 3(05): 24-27+30.

[33] Zhang Li. Research on Automatic Indexing of Web Information [D]. Zhejiang University, 2014.

[34] Fu Yan, Xu Zhaobang, Xia Hu, et al. Semi-automatic construction method of e-commerce website entity template based on reverse matching[J]. Journal of Chinese Information Processing, 2015, 29(2): 157-162,178. DOI:10.3969/j. issn.1003-0077.2015.02.019.

[35] Wang X, Chen X, Wang H, Wang W. Automatic extraction of key information from news web pages based on tags and block features[J]. Journal of Shandong University (Science Edition). 2019(03).

[36] Wang L. Research on the Online Extraction Method of Web News Publication Time [D]. Hefei University of Technology, 2018.

[37] Zhong Z, Li C, Qiao L, Zhang W, Guan Yan. An efficient method for extracting the publication time of Web news[J]. Journal of Chinese Computer Systems, 2013, 34(09): 2085-2089.

[38] Zhao X, Jin P, Yue L. TTP: A topic time parser for Chinese news web pages [J]. Journal of Chinese Computer Systems, 2013, 34(05): 1042-1049.

[39] Liao J. Automatic extraction of web page text based on tag style and density model[J]. Information Science, 2018, 36(07): 123-129.

[40] Pan X, Chen C, Liu R, Wang M. Text extraction based on path similarity of web page DOM tree nodes[J]. Journal of Chinese Computer Systems, 2016, 35(19): 74-77.

[41] Lv R, Liu J. Adaptive web page text extraction method based on decision tree[J]. Modern Computer (Professional Edition), 2019(07): 16-22.

[42] Miao Lin. Research on Application of Web Information Extraction Technology in Enterprise Competitive Intelligence Platform [D]. University of Electronic Science and Technology of China, 2010.

[43] Liu Zhiqiang, Du Yuncheng, Shi Shuicai. Web page news key information extraction based on improved hidden Markov model[J]. Data Analysis and Knowledge Discovery, 2019, 3(03): 120-128.

[44] Wu G, Hu Jun, Li L, Xu Z, Liu P, Hu X, Wu X. Online Web news content extraction based on tag path feature fusion [J]. Journal of Software, 2016, 27(03): 714-735.

[45] Gogar T, Hubacek O, Sedivy J. Deep Neural Networks for Web Page Information Extraction. [J] IFIP International Conference on Artificial Intelligence Applications and Innovations. 2016

[46] Chen B, Li J. Network representation learning model based on attention mechanism fusion multi-level neighborhood information [J]. Journal of Chinese Computer Systems, 2021, 42(04): 761-765.

[47] Wen D. Code retrieval based on ranking learning [D]. Liaoning: Dalian University of Technology, 2019.

[48] Alon, Uri, Omer Levy, and Eran Yahav. code2seq: Generating sequences from structured representations of code. [C] ICLR 2019

[49] Li X, Jiang H, Kamei Y, et al. Bridging Semantic Gaps between Natural Languages and APIs with Word Embedding[J]. IEEE Transactions on Software Engineering, 2018

[50] Wengong J, Kevin Y, Regina B. Learning Multimodal Graph-to-Graph Translation for Molecular Optimization[J],arXiv:1812.01070 [cs.LG] 2019

[51] Hu, Baotian, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences[C]. In Advances in neural information processing systems, pp. 2042-2050. 2014.

[52] William L. Hamilton, Rex Ying, Jure Leskovec. Inductive Representation Learning on Large

Graphs[C]. NIPS 2017. arXiv:1706.02216 [cs.SI].

[53] Liu Wenbin, He Yanqing, Wu Zhenfeng, Dong Cheng. Research on sentence alignment me thod based on BERT and multi-similarity fusion [J/OL]. Data analysis and knowledge disco very: 1-19[2021-04-16].http:/ /kns.cnki.net/kcms/detail/10.1478.G2.20210402.1512.002.html.

[54] Liang Hongxiang, Zhang Buye, Li Weizhuo, Cheng Qianya. Discovery of similar cases co mbining network representation learning and text convolutional networks[J/OL]. Computer E ngineering and Applications: 1-9[2021-04-19].http: //kns.cnki.net/kcms/detail/11.2127.TP.20201 229.1248.002.html.