JRSSEM 2022, Vol. 01, No. 7, 785 – 794
E-ISSN: 2807 - 6311, P-ISSN: 2807 - 6494

AUTOMATIC WEB NEWS CONTENT EXTRACTION

Gusti Lanang Putra Eka Prismana*
Universitas Negeri Surabaya
e-mail: lanangprismana@unesa.ac.id
*Correspondence: lanangprismana@unesa.ac.id

Submitted: 27 January 2022, Revised: 06 February 2022, Accepted: 18 February 2022

Abstract. The extraction of the main content of web pages is widely used in search engines, but a lot of irrelevant information, such as advertisements, navigation, and junk information, is included in web pages. Such irrelevant information reduces the efficiency of web content processing in content-based applications. This study aimed to extract web pages using DOM Tree in the rationality of segmentation results and efficiency based on the information entropy of nodes from the DOM Tree. The first step of this research was to classify web page tags and only processed tags that affected the structure of the page. The second step was to consider the content features and structural features of the DOM Tree node comprehensively. The next was to perform node fusion to obtain segmentation results. Segmentation testing was carried out with several web pages with different structures so that it showed that the proposed method accurately and quickly segmented and removed noise from web page content. After the DOM Tree was formed, the DOM Tree would be matched with the database to eliminate information noise using the Firefly Optimization algorithm. Then, testing and evaluating the Firefly Optimization method in effectiveness aspect were done to detect and eliminate web page noise and produce clear documents.

Keywords: DOM tree; web; news; extraction; firefly.

DOI : 10.36418/jrssem.v1i7.107

INTRODUCTION

Online news is one of the big data sources. Information in the form of news articles is published every minute (Allen, Howland, Mobius, Rothschild, & Watts, 2020). There is a lot more information than the person doing the analysis, and this is a potential problem where a lot of data can be ignored (Newman & Cain, 2014). Search engines are often used to obtain information. Search engines use web spiders to surf the web and retrieve links that may contain the information sought, and present the information in the form of a collection of hyperlinks. Search engines are capable of retrieving information from the web but not from the unseen or hidden web, so this makes data extraction a very impractical task. The challenges faced by extractors include heterogeneous formats, changes in the structure of web pages, the introduction of more and more advanced technologies to improve UX, and others.

Extracting information from multiple sources has many problems such as finding useful information, extracting knowledge from large data sets, and studying individual users. Various methods and techniques have been developed (Abburu & Golla, 2015). Because the amount of information obtained on the web increases radically, the amount of redundant web content also grows at the same time. Therefore, updating the incoming data and retrieving useful information without duplicating data from the web, the web mining research community pays attention to an existing activity related to the information from the web quickly and efficiently (Dey & Jain, 2020).

The articles published on a website are mostly in the form of unstructured information because they usually contain main information or main content, advertisements, navigation, and other additional information. The amount of that information resulted in the difficulties of getting the main core information and finding relevant values and knowledge in the form of structured information, such as the form of a database. The mechanism for extracting a collection of texts to obtain facts in the form of events, entities, and relationships in the form of structured information as input to a database or ontology is called information extraction (Kara et al., 2012).

This study aimed to extract DOM Tree-based web pages in the rationality of segmentation and efficiency results. This study used a method based on the information entropy of nodes from the DOM Tree. The first step carried out in this research was classifying web page tags and only processing tags that affected the structure of the page. The second step was considering the content features and structural features of the DOM Tree node comprehensively, calculating the information entropy of the nodes and the maximum text density of subnodes, and determining whether a node was a block page or independent. The third step was performing node fusion to obtain segmentation results. After getting the segmentation results, the web page noise will be removed to match the DOM Tree built with the database (Velloso & Dorneles, 2013). After the DOM Tree was formed, then the DOM Tree was matched with the database to eliminate information noise by

DOI : 10.36418/jrssem.v1i7.107

using the Firefly Optimization algorithm. Furthermore, testing and evaluating the Firefly Optimization method in effectiveness is done to detect and eliminate web page noise and produce clear documents (Yu & Jin, 2017). This research is expected to get a better approach for extracting data from semi-structured documents, both based on structure and data using several optimization methods, techniques, and algorithms as well as finding a method for removing web page noise that often arises from web content extraction.

METHODS

A. System Architecture

The system architecture of the information extraction is an adaptation of the general architecture of the information extraction system and the general architecture of the system. The input of the information extraction system is in the form of unstructured natural language text from the source text on the web page (HTML text). The information extraction process according to the OBIE concept will involve an ontology as an extraction guide and produce output in the form of extracted information which is represented in the form of XML and annotated text. The ontology-guided extraction process will extract things such as classes, properties, and instances (Wimalasuriya & Dou, 2010).

The information extraction system architecture in this study was divided into three phases, namely the training phase, the development phase, and the evaluation phase. In the training phase, the system identifies patterns and lists of dictionaries (called semantic lexicon), which were learned using a bootstrapping approach. Previously, the corpus had to go through the preprocessing stage before the training process. The purpose of the training phase was to generate patterns and semantic lexicon. The development phase was a phase to identify and classify relevant information in a new collection of texts. The text used was not included in the corpus in the training process. The pattern was generated to get the extraction rules. In the development phase, the input text was passed to the OBIE system to produce an output. The last phase was the evaluation or testing phase. The architecture of the OBIE system in this study can be seen in Figure 1 below:

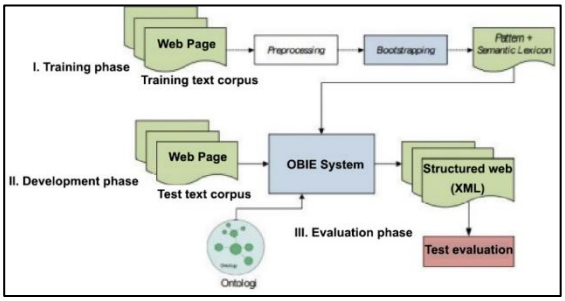

Figure 1. System Architecture

B. Preprocessing

The initial stage of information extraction to perform preprocessing on text input which aims to prepare the text into data that can be processed as input to the information extraction system. The parsing process was carried out on all text documents to identify all nouns or noun phrases (NP) and their

contexts. The parsing process in preprocessing consists of sentence detection, cutting sentences into tokens or words (tokenization), providing syntactic information (POS tagging), and cutting phrases (NP chunker). After parsing the text was complete, then the indexing and filtering process was carried out. The sentence indexing process was done by detecting words/ NP phrases in sentences, tokens on the left of NP, and tokens on the right of NP. After the indexing process was complete, the output (called a document-set) was stored in the database for processing.

C. DOM Tree

The Document Object Model (DOM) specification is an object-based interface developed by the World Wide Web Consortium (W3C) that constructs XML and HTML documents as tree structures in memory. Applications access XML data through an in-memory tree, which is a replication of how the data is structured. The DOM also allows users to dynamically traverse and update XML documents. It provides a model for the entire document, not just for a single HTML tag. The Document Object Model represents a web document as a tree. It is highly adaptable and can be used to renovate entire web pages. This is an explicit HTML document model. Some HTML tags do not include closing brackets. For some of these tags, the closing parenthesis is inferred by the following tag, for example, the <LI> tag is closed by the following </LI> tag. To analyze a web page, first, check the HTML document syntax because most HTML Web pages are not well-formed. After that, it passes the web page through an HTML parser which increases the markup and generates a DOM tree. Then the system breaks it down into several sub-trees according to the threshold value. Different websites have different layouts and serving styles, therefore the depth of the Web page tree varies according to the presentation style (Kim & Lee, 2017).

The system must know the maximum level of the DOM tree to select the best option. DOM tree-based method, as a segmentation method of high interest, is proposed based on the characteristics in which after parsing the DOM, HTML documents can form a tree structure that can accurately describe the hierarchical relationship between elements in a web page and is convenient for computer processing. After the web pages were parsed into a DOM tree, the algorithm grouped the web pages mainly by content features and DOM tree structure features. DOM is a common tool for representing web pages. In the DOM, the web page will be represented as a set of tags and a hierarchical relationship between the tags with the function of each tag, which allows the user to classify a message and an HTML tag (Sun, Song, & Liao, 2011).

D. Firefly Algorithm

This algorithm is inspired by nature which is based on a firefly's flash of light and mimics how fireflies interact with each other. In the firefly algorithm, some web documents or web pages are

taken as input. The following are the steps in implementing the firefly algorithm as shown in the figure below.

Firefly Algorithm for Web Page Noise Removal
Input: Multiple web pages
Output: Extract relevant content from web pages
Step 1: Access multiple web page
Step 2: Read one by one page
Step 3: Check web HTML tag
Step 4: Consider the document with various tags
Step 5: Objective function f(wi) w=(w1,w2,w3..)
Step 6: Generate an preliminary population of fireflies
Step 7: Formulate light intensity
Step 8: Define absorbtion coefficient γ
Step 9: While (t<Max_ Generation)
Step 10: For i=1:n
Step 11: for j=1:n (n fireflies)
Step 12: If(I_j> I_i)
Step 13: Move firefly i to j
Step 14: Calculate new solutions and update light intensity
Step 15: End if
Step 16: End for j
Step 17: End for i
Step 18: Identify the noisy information
Step 19: Eliminate the noises

After reading the web page, the checked HTML tags were given in step 3 then consider the web document with various tags. In step 5, the objective function was calculated and generated the initial population of fireflies in step 6. The light intensity was formulated in step 7 and determines the absorption coefficient in step 8. In step 9, the maximum generation was evaluated based on the new solution of updated light intensity. In steps 18 and 19, noisy information is identified and eliminated (Bumbaca et al., 2011).

Finally, the main content was extracted. All information related to web pages was stored for efficient pattern retrieval using the Firefly technique. A database was created using an artificial neural network to store related data from web pages. Matching the constructed DOM tree with the database was to eliminate noisy information (Mangat, 2014). In the end, we can get the main content. The initialization of the objective function *f (wi)* is calculated using the light intensity *I (o)* which varies according to the inverse square law with the following formula:

$$I(o) = \frac{I_s}{I_o} \text{..... (1)}$$

I(o) is the intensity at the source and *r* is the distance of the observer. The light intensity *I* varies with the square of the distance *d*. The absorption coefficient *s* calculated using the following formula:

$$II_o e^{-yd^2} \text{..... (2)}$$

The attraction of fireflies is proportional to the intensity of light perceived by other fireflies. The brightness observed by adjacent fireflies calculated using the formula below:

$$\beta\beta_0 e^{-yd^2} \text{..... (3)}$$

The next step is initializing the firefly population. Firefly *i* is attracted to firefly *j* which is more attracted, its movement is evaluated using the following fouls:

$$x_i = x_i + \beta(x_j - x_i) + \alpha\varepsilon \text{ ..... (4)}$$

The webpage noise removal funtto is calculated using the following formula:

$$fitnes = \alpha\frac{toT_{neg}}{T_{tot}} + \frac{\beta}{F} \text{.....} \quad (5)$$

Ttot is represented as the total number of tags on a web page, meanwhile, *Tneg* is the negative tag on

a web page. Then, *F* is denoted as *F* and β is firefly attraction.

RESULT AND DISSCUSION

To verify the effect of the proposed method, the algorithm was implemented using the DOM method to analyze HTML tags which were then processed using an optimization algorithm to remove page noise using the firefly method. There were trials of several pages from different websites, such as *Baidu Encyclopedia, Sina Blog, Tencent News, Blog Park,* and other websites. This page has a clear distinction in content and structure and can illustrate the implementation of the algorithm well. This study proposed a DOM tree-based web page segmentation method that comprehensively considered the structural features and content features of web pages, used node information entropy to group nodes from the parsed DOM tree, and obtained the final segmentation results with node fusion in the form of news text content.

1. Description of Datasets

To carry out the experiment, datasets were collected from different web pages. These web pages contain meaningful content and also have noise such as advertising banners, copyright links, page icons, irrelevant navigation, and junk information included in the web pages (Kaur, 2014).

2. Performance Measure

In this study, valid blocks, invalid blocks, execution time, precision, recall, and F-Measure were considered as performance factors that were evaluated based on the number of negative tags and the total number of tags. Experiments were carried out to test and evaluate the proposed method, the effectiveness of detecting and removing noise to block web pages and produce clear documents. The validity and accuracy of the proposed algorithm were checked using recall, precision, and F-measures from the information retrieval field. The datasets used in the experiment consisted of several pages from different websites.

3. Scrapping Web Using DOM

The first step done in this research was to classify the web page tags and only process the tags that affected the page structure. The second step was to comprehensively consider the content features and structural features of the DOM tree node. This calculated the node information entropy and the maximum sub-node text density to determine whether or not a node was an independent page block. Next, the node fusion was carried out to obtain segmentation results. This research was applied to several pages from different websites for scraping news content on web pages. The following was a DOM standard used for initial website identification.

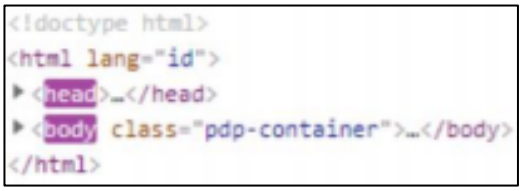

Figure 2. DOM Standard

The input in this process was the URL address or the HTML code of the web page to be extracted. The initial step was to convert HTML documents

from web pages into text. The next data process used variable processing in the form of a string.

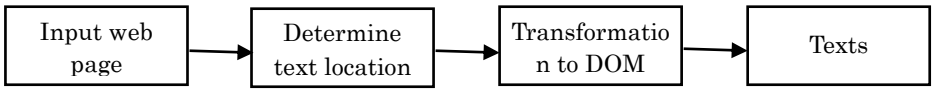

Figure 3. DOM Process

There are several sections on a web page such as texts, images, lists, or tables. Therefore, the first step was to determine which part was the text of the HTML document. The texts were detected by making use of the string match function. Texts in HTML documents can be identified by the tag <p> at the beginning and <\p> at the end of the table. The tag <p> was used to detect reading in the form of a paragraph. If there are two or more paragraphs on a web page, the application stores the table at the second index of the variable in the form of an array. The next step was to create a DOM tree from the detected paragraphs. The DOM tree is composed of text-forming tags, namely tags <p>. This tag is detected to determine the news content portion of the paragraph. The third step was to extract or retrieve the data portion of the DOM tree. After the data section in the table was obtained, the next step was to save the data in the form of a CSV file. The selection of the form of the storage file is intended so that the extraction results can be used for subsequent needs, such as integration with data from other tables or to be stored in a database. Different websites have different layouts and serving styles, therefore, the depth of the Web page tree varies according to the presentation style. The system must know the maximum level of the DOM tree to select a good choice of threshold levels. That is why the system traverses the entire DOM tree to obtain the maximum DOM depth.

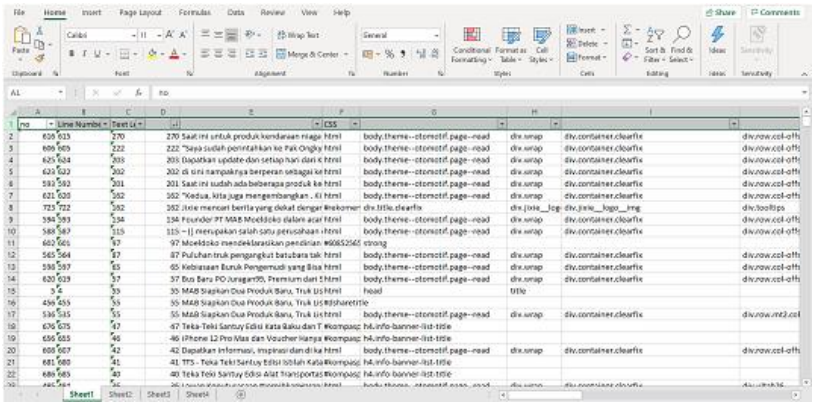

Figure 4. Scrapping 1

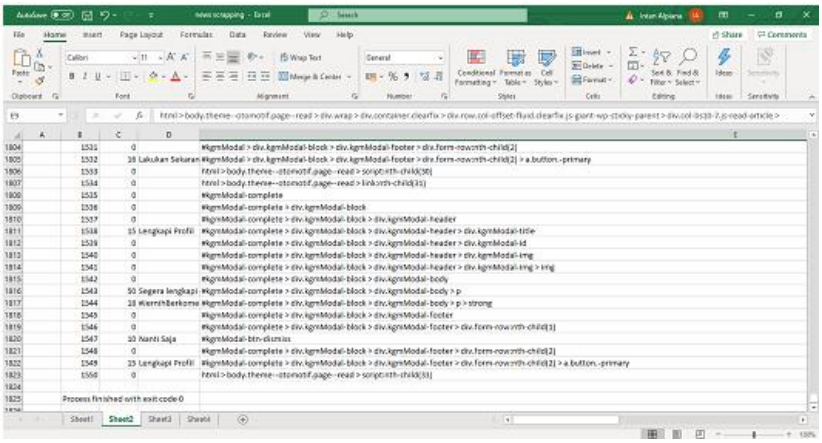

Figure 5. Scrapping 2

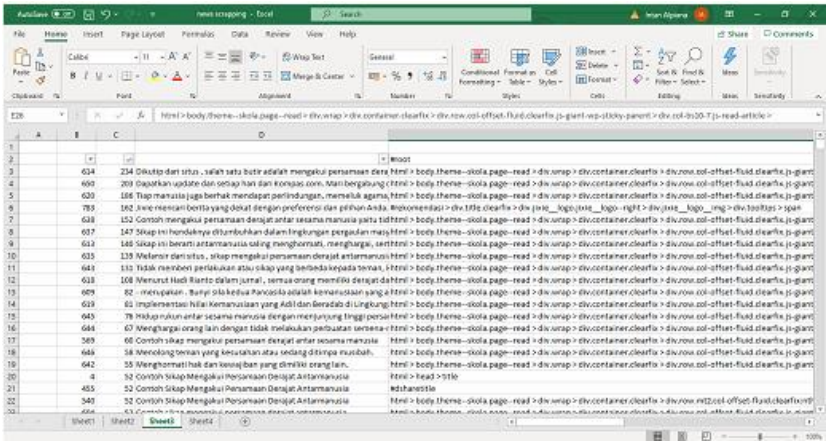

Figure 6. Scrapping 3

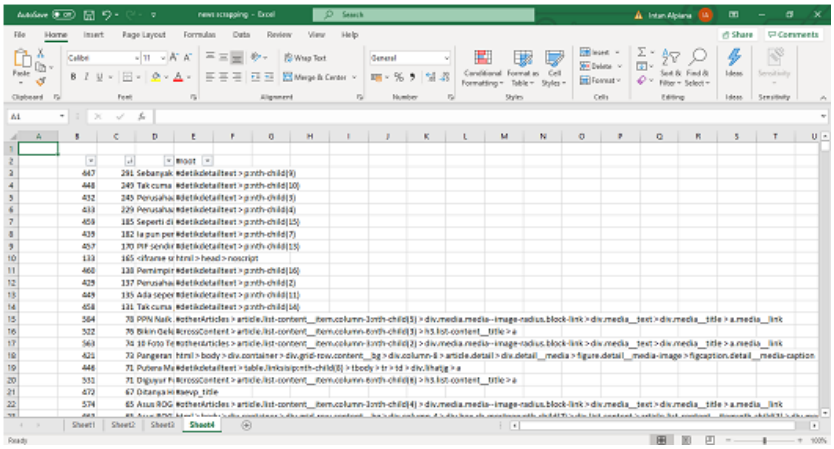

Figure 7. Scrapping 4

4. Optimization of Firefly Algorithm Method

All information related to the web pages was stored for efficient pattern retrieval using the Firefly technique. A database is created using an artificial neural network to store related data from web pages. Matching the created DOM tree with the database is to remove the noise information to get

the main content (Liu, 2017). The initialization of the objective function f *(wi)* is calculated using the light intensity I(o) which varies according to the inverse square law. The formula used I as follows

$$I(o) = \frac{I_s}{I_o}$$

I(o) is the intensity at the source and r is the observer's distance. The light intensity I varies with the square of the distance d. The absorption coefficient γ is calculated using the following formula:

$$II_o e^{-yd^2}$$

The steps of the firefly algorithm can be described as follows:

a) Step 1: Access several web pages
b) Step 2: Read every web page, one by one
c) Step 3: Check web HTML tags
d) Step 4: Consider documents with multiple tags
e) Step 5: Objective function of *f (wi) w = (w1, w2, w3 ..)*
f) Step 6: Produce an initial population of fireflies
g) Step 7: Formulate light intensity
h) Step 8: Determine the absorption coefficient γ
i) Step 9: Meanwhile, *(t <Max_* Generation)
j) Step 10: For *i = 1: n*
k) Step 11: For *j = 1: n (n of fireflies)*
l) Step 12: If *(Ij> Ii)*
m) Step 13: Move the fireflies to *j*
n) Step 14: Calculate the new solutions and renewing light intensity
o) Step 15: End *if*
p) Step 16: End for *j*
q) Step 17: End for *i*
r) Step 18: Identify noisy information
s) Step 19: Eliminate noise
t) Step 20: End it

CONCLUSIONS

Based on the study having been conducted entitled *Automatic Web News Content Extraction*, there are several things that can be concluded as follows:

a. Automatic Web News Content Extraction Algorithm can be used in optimal extraction of main web page content (news) and results in a better approach.
b. The use of different datasets has a varying effect on the performance of the indicated precision, recall, and f-measure parameters. This depends on the page reference level of similarity with the extracted page. The more similar, the more stable the performance shown.
c. The use of raw and valid datasets also has varying effects on precision, recall, and f-measure performance. It depends on the validation process of the tag structure of the website page.
d. The researcher suggests that extracting web page content can be tried using other methods that will produce a better approach for data extraction.

REFERENCES

Abburu, Sunitha, & Golla, Suresh Babu. (2015). Satellite image classification methods and techniques: A review. *International Journal of Computer Applications*, *9*(8).

Allen, Jennifer, Howland, Baird, Mobius, Markus, Rothschild, David, & Watts, Duncan J. (2020). Evaluating the fake news problem at the scale of the information ecosystem. *Science Advances*, *6*(14), 3539. 10.1126/sciadv.aay3539

Bumbaca, Daniela, Wong, Anne, Drake, Elizabeth, Reyes II, Arthur E., Lin, Benjamin C., Stephan, Jean Philippe, Desnoyers, Luc, Shen, Ben Quan, & Dennis, Mark S. (2011). Highly specific off-target binding identified and eliminated during the humanization of an antibody against FGF receptor 4. *MAbs*, *3*(4), 376–386. Taylor & Francis. https://doi.org/10.4161/mabs.3.4.15786

Dey, Arnab, & Jain, Sudhanshu. (2020). Automatic skimming of web pages on a single click efficiently. *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 596–602. IEEE. 10.1109/ICOEI48184.2020.9143003

Kara, Soner, Alan, Özgür, Sabuncu, Orkunt, Akpınar, Samet, Cicekli, Nihan K., & Alpaslan, Ferda N. (2012). An ontology-based retrieval system using semantic indexing. *Information Systems*, *3*(4), 294–305. https://doi.org/10.1016/j.is.2011.09.004

Kim, Yeongsu, & Lee, Seungwoo. (2017). SVM-based web content mining with leaf classification unit from DOM-tree. *2017 9th International Conference on Knowledge and Smart Technology (KST)*, 359–364. IEEE. 10.1109/KST.2017.7886134

Newman, George E., & Cain, Daylian M. (2014). Tainted altruism: When doing some good is evaluated as worse than doing no good at all. *Psychological Science*, *25*(3), 648–655. https://doi.org/10.1177/0956797613504785

Sun, Fei, Song, Dandan, & Liao, Lejian. (2011). Dom-based content extraction via text density. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 245–254. https://doi.org/10.1145/2009916.2009952

Velloso, Roberto Panerai, & Dorneles, Carina F. (2013). Automatic web page segmentation and noise removal for structured extraction using tag path sequences. *Journal of Information and Data Management*, *4*(3), 173.

Wimalasuriya, Daya C., & Dou, Dejing. (2010). Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science*, Vol. 3, pp. 306–323. Sage Publications Sage UK: London, England. https://doi.org/10.1177/0165551509360123

Yu, Xin, & Jin, Zhengping. (2017). Web content information extraction based on DOM tree and statistical information. *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, 1308–1311. IEEE. 10.1109/ICCT.2017.8359846