Web Page Extraction and Classification Using JSOUP and Naïve Bayes

S Cokrowibowo¹, N Nur¹, and Irmayanti¹*

¹Informatics Engineering, Faculty of Engineering, Universitas Sulawesi Barat, Majene, Indonesia

*Email: yirma294@gmail.com

Abstract. Classification of web pages manually requires quite a long time because most of the available web pages are not structured, so the classification method is needed quickly and accurately. Naïve Bayes algorithm with a good probabilistic approach in classifying web pages, seen from the advantages that are included in the classical category with a simple probability concept. However, this algorithm provides pretty good performance for many modern cases with large data. For the process of extracting information from web pages, it is proposed to use JSOUP which is a java library that provides a good API for extracting, manipulating data, and completing the initial data cleaning using the best methods from DOM, and CSS. The use of the JSOUP library makes it possible to be able to do web page analysis without having to save web documents to a computer store, so computer storage resources will be constant even though the amount of training data is increased. In this study, implementing JSOUP as a tool for extracting information from web pages and Naïve Bayes algorithm for classification of web pages.

1. Introduction

Web page classification, also known as web page categorization, is the process of assigning web pages to one or more predefined category labels. Classification has traditionally been proposed as a supervised learning problem [1] where a set of labeled data is used to train a classifier that can be applied to label as an example of the future. Classification of web pages manually requires quite a long time, therefore, a method that can be used in the classification process is fast and accurate.

One of the classification methods commonly used is Naïve Bayes. The Naïve Bayes algorithm is a classic category with a simple probability concept. However, this algorithm provides pretty good performance for many modern cases with large data. The Naïve Bayes algorithm uses an ancient theorem of the 18th century heritage, discovered by Thomas Bayes [1]. The Naïve Bayes method can do the job efficiently. If input is considered a set of word attributes we can calculate the probability of whether a web page belongs to a certain class or not. This process consists of two phases; training phase and classification phase. In the training phase, the filter will be trained using a collection of words that are known to be classified into web pages. Based on the appearance of each word given a probability for its capacity to determine its class classification. Then, in the classification phase use a collection of words to classify each web page [2].

Extracting web pages to be classified requires quite a long time if done manually. Therefore we need a tool to be able to extract web pages automatically. Web page contains huge amount of information so we must be able to extract valuable information [3]. By using the JSOUP java library to work with...
HTML it provides a good API for extracting, manipulating data, and completing initial data cleaning using the best methods from DOM, CSS, and other methods similar to jQuery. JSOUP implements the WHATWG HTML5 specification, and parses HTML to the same DOM as modern browsers do [4].

2. Related Research

Aris J. Ordoñez, Rommel Evan J. Paje, and Rodel N. Naz [2] in 2018 conducted a study that discussed how to classify SMS received by rescue agents including spam, invalid, warning 1, warning 2, and warning 3. The method used in solving the problem is applying the Naïve Bayes algorithm, an independent learning algorithm, and together with Natural Language Processing. From this research the writer will use the Naïve Bayes algorithm in the case of different classifications.

Patel and Pandya [5] in 2017 conducted a study explaining the model for categorizing articles related to child development and parenting context, which began with cataloging through identification & analysis carried out with consideration of category keywords along with relevant information has been taken from various web sources to achieve classification with expected accuracy. From this study the authors took a reference on how to categorize web pages into three different classes.

Lassri Safae et al. [6] in 2018 conducted a study that discussed the characteristics of web pages, classification by reviewing machine learning with different algorithms to categorize web pages. Thus, tracing some of the assumptions of the method studied. From this study the authors used the Naïve Bayes algorithm which is one of machine learning in solving classification problems.

Wang Jie et al. [7] in 2015 conducted a study that resolved problems in maintaining the balance of agricultural production and markets by proposing a platform that included data crawling, information extraction and analysis of big data of agricultural products from the Internet. In its system data crawling and extraction uses JSOUP, an HTML collector as a fundamental technology. From this study the authors use JSOUP in solving different problems.

3. Literature Review

3.1. Classification

Web page classification, also known as web page categorization, is the process of assigning web pages to one or more predefined category labels. Classification has traditionally been proposed as a supervised learning problem how to learn the behavior of a function that maps a vector of dimension $N$ (for example $[X_1, X_2, ..., X_N]$) into one from several classes (which the user wants), just by looking at a few examples of the input-output pairs of the function [1].

3.2. JSOUP

JSOUP is a java library for working on HTML documents that provides a good Application Programming Interface (API) for extracting, manipulating data, and completing preliminary data cleaning using the best methods from Document Object Model (DOM), CSS, and other methods similar to jQuery. JSOUP implements the WHATWG HTML5 specification, and parses HTML to the same DOM as modern browsers do [3, 7]. Following are the main services available in the JSOUP library.

- Scrape and parse HTML from URL, file or string.
- Find and extract data using DOM traversal and CSS selector.
- Manipulate HTML elements, HTML attributes, and text.
- Clean the content sent by users using safe white-lists to prevent XSS attacks.
- Generates HTML tidy.

3.3. Naïve Bayes Algorithm

The Naïve Bayes algorithm is a classic category with a simple probability concept. However, this algorithm provides pretty good performance for many modern cases with large data. Several studies that use Naïve Bayes algorithm such as web page classification [8] and classification for diabetes.
mellitus [9]. The Naïve Bayes algorithm uses an ancient theorem of the 18th century heritage, discovered by Thomas Bayes [1]. In Bayes' Theorem, a conditional probability is stated as follows.

$$P(P|X) = \frac{P(P|H)P(H)}{P(X)}$$  (1)

In the process of the Naïve Bayes classifier, the D training set data set that contains a number of tuples X along with a number of m class labels (C1, C2,.....Cm) is categorical type data, each tuple with dimension n expressed as A1, A2,....An. Calculates the prior probabilities of each class so that a class probability table is generated, where $P(C_i) = \frac{\text{the number of tuples in the class } C_i}{\text{all tuples in the } D \text{ training data set}}$. Calculate the probability of each value for all n attributes (A1, A2,....An) for the whole class m (C1, C2,.....Cm) resulting in a number of n probability tables, if there is a probability value of 0, Laplacian correction did. Learning outcomes are as many as (n + 1) tables, namely: one class probability table and n tables that contain the probability of each value on all existing attributes. Here is the Laplacian correction equation.

$$P(X_{kj}|C_i) = \frac{f(X_{kj}|C_i)+1}{f(C_i)+|W|}$$  (2)

The following equation can be used to classify an input tuple by finding the maximum probability of all existing classes.

$$P(X|C_i)P(C_i) = \left(\Pi_{k=1}^{n}P(X_k|C_i)\times P(C_i) = P(X_1|C_i)\times\ldots\times P(X_n|C_i)\right)\times P(C_i)$$  (3)

$$P(X|C_j)P(C_j) > P(X|C_j)P(C_j) \text{ for } 1 \leq j \leq m, j \neq i$$

Equation (3) can be simplified into the following equation.

$$P(C_i) = P(C_i)\Pi_{k=1}^{n}P(X_k|C_i)$$  (4)

4. Research Design
The classification process consists of four main phases; Extraction, preprocessing, attribute selection, training data, and testing data.

![Flowchart](image)

**Figure 1.** Research design flow

4.1. Extraction
In the extraction process the writer uses the JSOUP library. The system will accept input in the form of a web page URL that will be extracted. Then JSOUP will load the web page based on the input URL. Next will be extracted data from web pages by separating between the text from HTML tags [10]. Figure 2 shows an example of web page extraction result using JSOUP.
4.2. Preprocessing
Preprocessing aims to prepare data so that it is easily analyzed. Preprocessing is divided into several processes consisting of case folding, tokenizing, and filtering. In Figure 3, there is a preprocessing process. Case folding is the initial stage of preprocessing text that changes all the letters of the text into lowercase characters. Characters accepted are only ‘a’ to ‘z’. Characters other than letters will be removed and considered as delimiter. Tokenizing is the stage of cutting the input string based on each word that makes it up. Filtering is a process carried out by taking important words from the token results and deleting stop words. Stop words are words that are not descriptive so that they can be discarded or removed and have no effect in the process. In Indonesian, the examples of stop words is "that", "and", "from", "in", "like" and others [11].

4.3. Attribute Selection
Attribute selection is the process of determining what words (terms) will be used to represent web pages. In addition to describing the contents of web pages, these terms are also useful for differentiating one web page from another web page in a collection of web pages.

4.4. Training Data
In the data training phase, web pages are trained using the Attribute selection which is known to be classified into web pages using the Naïve Bayes method with Equation (2). Based on the appearance of each word, a probability for its capacity to determine its class classification is given.

4.5. Testing Data
After conducting the training data, the next step is to test the model of the training data results by using a new web page that is not yet known which class category. This is where the process of classifying new data into classes that has been determined by studying the trace attributes that have been previously trained with Equation (4). Figure 4 shows the stages or flowchart of classification process:
Figure 4. Training data and testing data

5. Results and Discussion

Naïve Bayes classification model testing is done using the Java programming language. In this paper the test is carried out 6 (six) times with the amount of training data for the first test until the last test in a row of 9, 18, 36, 72, 150, and 300 URLs of web pages that have been saved into the text document (*.txt), with the same testing data as many as 100 URLs of web pages that have been saved into the text document (*.txt). The overall results of the training data can be seen in the following graphic image.

![Graph showing accuracy results]

Figure 5. Test result accuracy

From the graph above it can be seen that the level of accuracy is directly proportional to the amount of training data, meaning that the more the amount of training data, the level of accuracy will increase. This is because the more training data, the more word attributes (tokens) that characterize a particular class attribute. Classification errors occur when there are web pages that should be categorized into
certain class categories but the Naive Bayes classification model identifies into other class categories. This is due to the lack of word attributes (tokens) from web pages in the training data for a class category. So the probability of probability for that class category is small compared to other classes.

6. Conclusion
Classifying web pages using the Naive Bayes algorithm with a large amount of training data can maximize the level of accuracy, with the use of JSOUP allows being able to do web page analysis without first having to save web pages to storage computers. So that the efficient use of computer storage resources will be constant even though the amount of training data is increased.

References
[1] Suyanto, Basic and Advanced of Machine Learning, Bandung: Informatika Bandung, 2018. [In Bahasa]
[2] R. E. J. P. R. N. Aris J. Ordoñez, “SMS Classification Method for Disaster Response,” pp. 233-236, 2018.
[3] C. Saini and V. Arora, “Information Retrieval in Web Crawling: A Survey,” pp. 2635-2643, September 2016.
[4] J. Hedley, “jsoup: Java HTML Parser,” 2019. [Online]. Available: https://jsoup.org/. [Accessed October 8, 2019].
[5] A. D. Patel and D. V. N. Pandya, “Web Page Classification based on Context to the Content Extraction of Articles,” pp. 539-541, 2017.
[6] L. Safae, B. E. Habib and T. Abderrahim, "A Review of Machine Learning Algorithms for Web Page Classification," 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, 2018, pp. 220-226.
[7] J. Wang, S. Yang, Y. Wang and C. Han, “The Crawling and Analysis of Agricultural Products Big Data based on JSOUP,” pp. 1197-1202, 2015.
[8] B. P. Asatkar, P. K. Wagh and D. P. N. Chatur, “Classification of Web Pages Using Naive,” IJJRSET, pp. 112-117, 2017.
[9] S. Sankaranarayanan and T. P. Perumal, “Analysis of Naïve Bayes Classification for Diabetes Mellitus,” JCSE, pp. 520-524, 2018.
[10] S. Cokrowibowo and Ismail, “Extraction and Visualization of Web Text Mining Using Jsoup” pp. 1-10, 2018. [In Bahasa]
[11] I. G. A. Socrates, A. L. Akbar and M. S. Akbar, “Naïve Bayes Optimization with Feature Selection and Weight Gain Ratio, [In Bahasa]” pp. 22-30, April 2016.