ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

Content Extraction Model Using DOM and NLP

Sarada Devi. CH, Dr. Kumanan .T, Prameela Devi. Chillakuru

Research Scholar, Meenashi Academy of Higher Education and Research, Chennai, India

Professor, Dept. of CSE, Meenashi Academy of Higher Education and Research, Chennai, India

Research Scholar, Meenashi Academy of Higher Education and Research, Chennai, India

ABSTRACT: Dom tree model-based content extraction method is proposed. DOM processor will do some pre-treatment such as calculating content density and hyperlink density values for extracting the related content. And also vector based tokenization is proposed for split up the contents. Finally POS tagging and map reducer model is used for extracting. The experimental results shows that the proposed model gives better performance in terms of accuracy, precision and recall.

I. DOM MODEL

The Document Object Model (DOM) is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document. The DOM represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects. DOM methods allow programmatic access to the tree; with them one can change the structure, style or content of a document. Nodes can have event handlers attached to them. Once an event is triggered, the event handlers get executed.

The principal standardization of the DOM was handled by the World Wide Web Consortium, which last developed a recommendation in 2004. WHATWG took over development of the standard, publishing it as a living document. The W3C now publishes stable snapshots of the WHATWG standard. In this work DOM tree model is used in the pre-processing phase for identifying the contents efficiently [1].

1.1 TOKENIZATION METHODS

Tokenization is the process of breaking a stream of textual content up into words, terms, symbols, or some other meaningful elements called tokens. There are several open source tokenization tools are there. The list of tokens turns into input for in additional processing including parsing or text mining. Tokenization is beneficial both in linguistics and in computer science, where it forms the part of lexical analysis. Generally, the process of tokenization occurs at the word level.

- Tokens are separated by the way of whitespace characters, such as a space or line break, or by punctuation characters.
- A Scriptio continuum, also known as scriptura continua or scripta continua, is a style of writing without any spaces or other marks in between the words or sentences. The main use of tokenization is to identify the meaningful keywords. The disadvantage of tokenization is difficult to tokenize the document without any whitespace, special characters or other marks [3].

1.2 TOOLS FOR TOKENIZATION

For tokenize a document, many tools are available. The tools are listed as follows,

- Nlpdotnet Tokenizer
- Mila Tokenizer
- NLTK Word Tokenize

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

- Text Blob Word Tokenize
- MBSP Word Tokenize
- Pattern Word Tokenize
- Word Tokenization with Python NLTK

In this work NLP based tokenizer is used to split up the sentences for an efficient extraction process.

II. PROPOSED METHODOLOGY

This section describes the proposed content extraction model. This model consist of four phase. First one is Document Object Model(DOM) tree based content extraction, Second one is Natural language processing (NLP) based tokenization, Third one is POS tagging using the NLP and the final phase is content retrieval using map reducer. Figure 1.1 Shows the overall Architecture diagram of the proposed model based web content extraction.

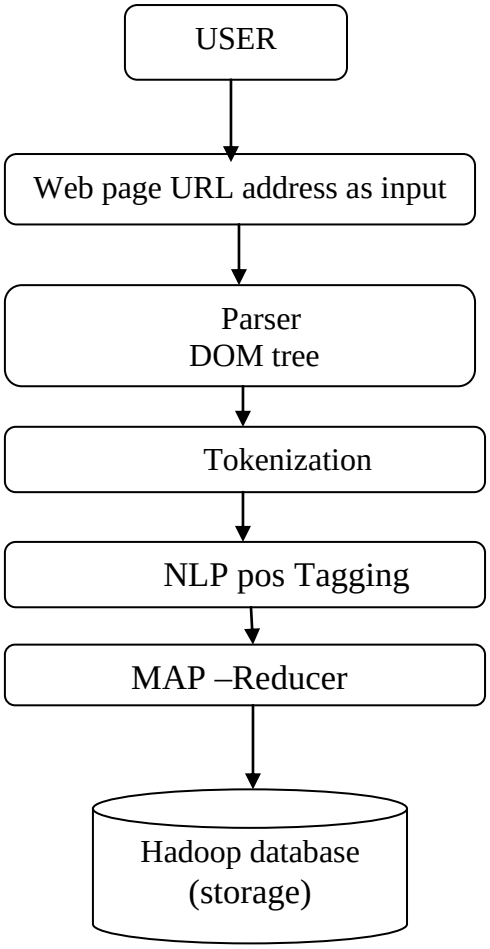

Figure 1.1 Overall Architecture Diagram

Input:

The framework offered input name as an URL then extracts all types of contents from web pages, and is shown in the textbox.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

2.1 PRE-PROCESSING

DOM is a common tool to represent web pages. In DOM, web pages are represented as a set of tags and the Hierarchical relationship between these tags. According to the function of each tag, classify the html tags into four categories:

- Tags for interaction. This kind of tag allows users to create a message and interact with the server or the presupposed functions in the page.
- Tags for style. This kind of tag is used to personalize the style of the content on the page.
- Tags for metadata description. This kind of tag is used to describe the metadata and basic attributes for the whole web page. Typical tags are <head>, <title>, <Meta>, <link>, <style>, etc.
- Tags as a container. This kind of tag is always to arrange the content of a web page. In general, the core content is always put in these tags. In this study, our target is extracting the main content from a web page. So, will construct the DOM according to the container-kind tags. Other kinds of tags such as tags for interaction, tags for style and tags for metadata will be blocked by filters at the first step. After the construction of DOM for a web page, need to choose characteristics to describe the web page.

2.2 CONTENT AND HYPERLINK IDENTIFICATION METHOD

Choose text density δ (b) and hyperlink density θ (b) to depict a web page. Their definitions are as follows

Definition 1.Text density δ (b) measures the number of text in a link b in DOM. link b is a pair of container kind tags. The calculation method is shown in formula (1):

$$\delta(b) = \begin{cases} \frac{T'(b)}{(l(b)-1*\max Len} & L(b) > 1 \\ \frac{T(b)}{maxLen} & L(b) \leq 1 \end{cases} \quad (1)$$

In this formula, L (b) represents the number of lines in link b. T (b) represents the number of characters when the number of lines in this link is 1. T'(b) represents the number of characters without the last line when the number of lines in this link is more than 1

Hyperlink density measures the number of hyperlinks in a link b in DOM. The calculation method is shown in formula (2):

$$\theta(b) = \frac{Ta(b)}{T(b)} \quad (2)$$

In this formula, Ta (b) represents the number of characters which are in the tags <a> and . T (b) represents the total number of characters in this link.

2.3 VECTOR BASED TOKENIZATION

In the proposed algorithm, tokenization is done based on the set of training vectors which are initially provided into the algorithm to train the system. The training documents are of different knowledge domain, are used to create vectors. The created vector helps algorithm to process the input documents. Tokenization is used to split the content into small pieces like sentence or words .The tokenization on documents is performed with respect to the vectors, use of vectors in pre tokenization helps to make whole tokenization process more precise and successful. Following algorithm shows the proposed tokenization of documents [9].

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

2.4 TOKENIZATION ALGORITHM

Input (Di)
Output (Tokens)
Begin
Step 1:
Collect Input documents (Di) where i=1, 2, 3….n;
Step2:
For each input Di;
Extract Word (EWi) = Di;
// apply extract word process for all documents i=1, 2, 3…n in and extract words//
Step 3:
For each EWi;
Stop Word (SWi) =EWi;
// apply Stop word elimination process to remove all stop words like is, am, to, as, etc. //
Stemming (Si) = SWi;
// It create stems of each word, like "use" is the stem of user, using, usage etc. //
Step 4:
For each Si;
Freq_Count (WCi)= Si;
// for the total no. of occurrences of each Stem Si. //
Return (Si);
Step 5:
Tokens (Si) will be passed to an **IR System**.
End

---

3 PART OF SPEECH (POS) TAGGING

Part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context—i.e., its relationship with adjacent and related words in a phrase, sentence, or paragraph.

POS tagging is now done in the context of computational linguistics, using algorithms which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags. POS-tagging algorithms fall into two distinctive groups: rule-based and stochastic. E. Brill's tagger, one of the first and most widely used English POS-taggers, employs rule-based algorithms.

The Parts of Speech of a given sentence can be detected using OpenNLP and then can be printed. Instead of full name of the parts of speech, OpenNLP uses short forms of each part of speech.

To tag the parts of speech of a sentence, OpenNLP uses a model, a file named en-posmaxent.bin. This is a predefined model that is trained to tag the parts of speech for the given raw text [10].

Steps

- Loading the model
- Tokenizing the sentence
- Instantiating the POSTagger ME class
- Generating the tags

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

- Printing the tokens and the tags

3.1 Map-Reducer

In a large dataset, a programming framework called map reduce is allowed for distributed and parallel processing in a distributed environment. It performs well without considering the issues namely fault tolerance, reliability, etc. So, map reduced.

At the end, the Map Reduce gives a flexibility to create code logic deprived of considerate regarding system design concerns [11, 12].

Let the basic unit of data in Map-Reduce computations be, in which keys including the values are continually only binary strings. The Map-Reduce program contains the sequence of reducers and mappers. Let the input be a multiset of pairs which is indicated through input to execute a program. For r=1, 2… R, do:

- For Execute Map, let be a multiset of pairs output through , i.e.,
- For Shuffle, let be a multiset of values for each k, then.
- For Execute reduce, the sequence of tuples such as, are generated by the reducer. Let be the multi-set of pairs output by , i.e.,

3.2 Mapper tasks

- Mapper in Hadoop takes each record created by the Record Reader as input. Then processes each document and create key-value pairs.
- This key-value pair is entirely unlike the input pair.
- The mapper output is called intermediate output which is saved on the local disk. Mapper doesn't save its production on HDFS, as it is temporary data and stored on HDFS to create multiple copies.

3.3 Reducer tasks

Reducer is a phase in Hadoop which comes after Mapper stage. The yield of the mapper is provided as the contribution for Reducer which proceeds and creates another arrangement of creation, which will be put away in the HDFS. Reducer first handles the in-between values for individual key produced as a result of the map function in addition to further providing an output.

Then, a map in addition to reduce phases run in slots implies that each node could run beyond one Map or else Reduce task in matching; generally, the slots quantity is correlated through cores quantity present in a specific node. Here, the Hadoop implementation time in all the phases are:

$$t_{total} = t_{map} + t_{shuffle} + t_{reduce} \tag{3}$$

The product of a particular map task and quantity of map tasks for each and every node is called as the time of the Map phase.

$$t_{map} = \frac{W_{map} * C_{U_{map}}}{T} * \frac{n_{map}}{p} \tag{4}$$

$$t_{reduce} = \frac{W_{U_{reduce}}^{in} * C_{U_{reduce}}}{T} * \frac{n_{reduce}}{p} \tag{5}$$

The output data size product of a single map task is the quantity of enduring shuffle tasks divided by means of a network bandwidth,

$$t_{shuffle} = \frac{W_{U_{map}}^{out} * (n_{map} \bmod p)}{B} \tag{6}$$

The performance model of map reduce in Hadoop is given by,

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

$$t_{total} = \frac{W_{map} * C_{U_{map}}}{T} * \frac{n_{map}}{p} + \frac{W_{map}^{out} * (n_{map} \bmod p)}{n_{map} * B} + \frac{W_{U_{reduce}}^{in} * C_{U_{reduce}}}{T * p} \quad (7)$$

Where - output information of a map phase, - number of map tasks, -single reduce task.

IV. RESULT AND DISCUSSION

This section discusses the experimental results of proposed model. The model is implemented using JAVA. And the proposed DOM-MR model is compared with the existing NLP-MR method in terms of precision, recall and accuracy for the dataset is extracted from several popular English, Simpli_ed Chinese, and Bengali news article websites on the Internet. This dataset contains URLs from diverse websites across the Internet, such as, personal blogs, articles, news etc. See Table 1.1 for some information about our dataset.

Website Language # of Webpages	NPR English 25	Prothom Alo Bengali 24
QQ Simpli_ed Chinese 25	Sina Simpli_ed Chinese 25	TechCrunch English 16
The Verge English 16	USA Today English 20	Chaos Mixed, includes RTL 189
Website Language # of Webpages	NPR English 25	Prothom Alo Bengali 24
QQ Simpli_ed Chinese 25	Sina Simpli_ed Chinese 25	TechCrunch English 16
The Verge English 16	USA Today English 20	Chaos Mixed, includes RTL 189
Website Language # of Webpages	NPR English 25	Prothom Alo Bengali 24

Table 1.1 Collected Dataset

PERFORMANCE METRICS

1) PRECISION

Precision refers to the percentage of the results which are relevant. It is defined as

$$\text{Precision} = \frac{True\ positive}{true\ positive + false\ positive} \quad (8)$$

2) **RECALL**

Recall refers to the percentage of total relevant results correctly classified by the proposed algorithm which is defined as

$$\text{Recall} = \frac{True\ positive}{true\ positive + False\ \ Negative} \quad (9)$$

3)ACCURACY

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions this model got right. Formally, accuracy has the following definition:

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

$$\text{Accuracy}=\frac{True\ positive+True\ Negative}{Total} \quad (10)$$

Methods	Precision	Recall	Accuracy
NLP-MR	87.91	88.89	86
DOM-MR	89.13	91.11	88

Table 1.2 Performance Comparison Result Values

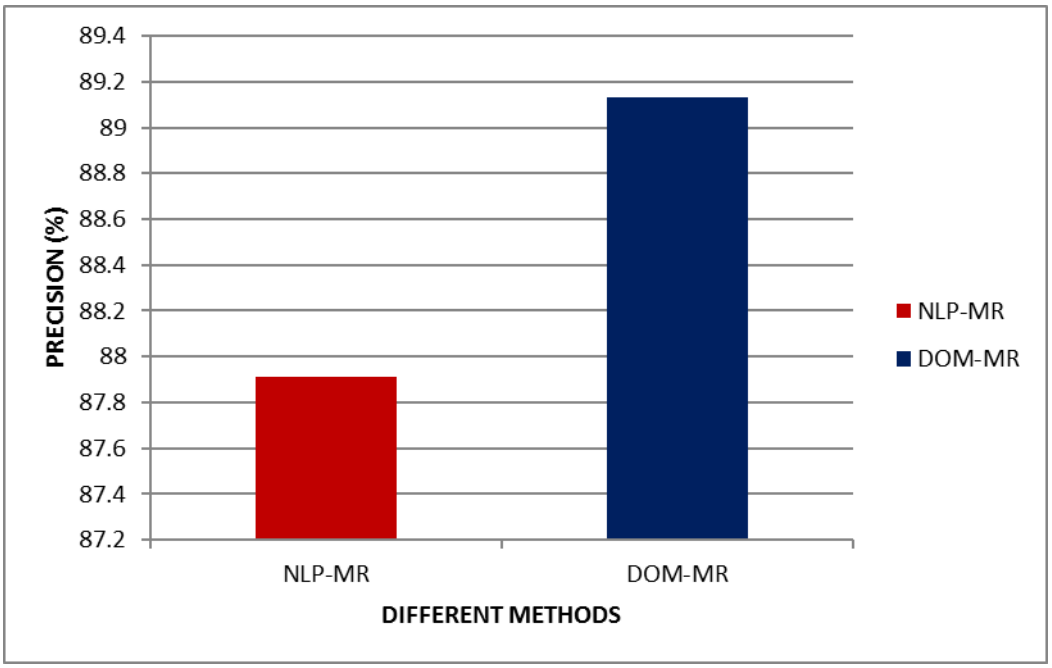

Figure 1.2 Precision Results of different Methods

Performance comparison results for precision of the proposed DOM-MR method is higher which is 89.13 % than the existing NLP-MR method produces only 87.91%.In the above graph different methods are taken as x-axis and the precision value is taken as Y-axis.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

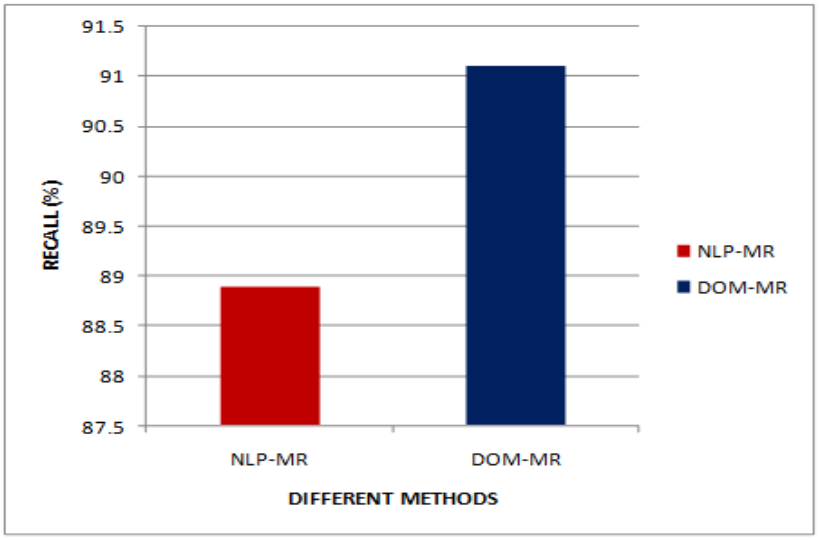

Figure 1.3 Recall Results of different Methods

Figure 1.3. Shows the performance comparison results of the proposed DOM-MR method and the existing NLP-MR method in terms of Recall**.** From the results it concludes that the proposed DOM-MR model produces higher **Recall results** of 91.11% whereas existing NLP-MR method produces only 88.89% respectively.

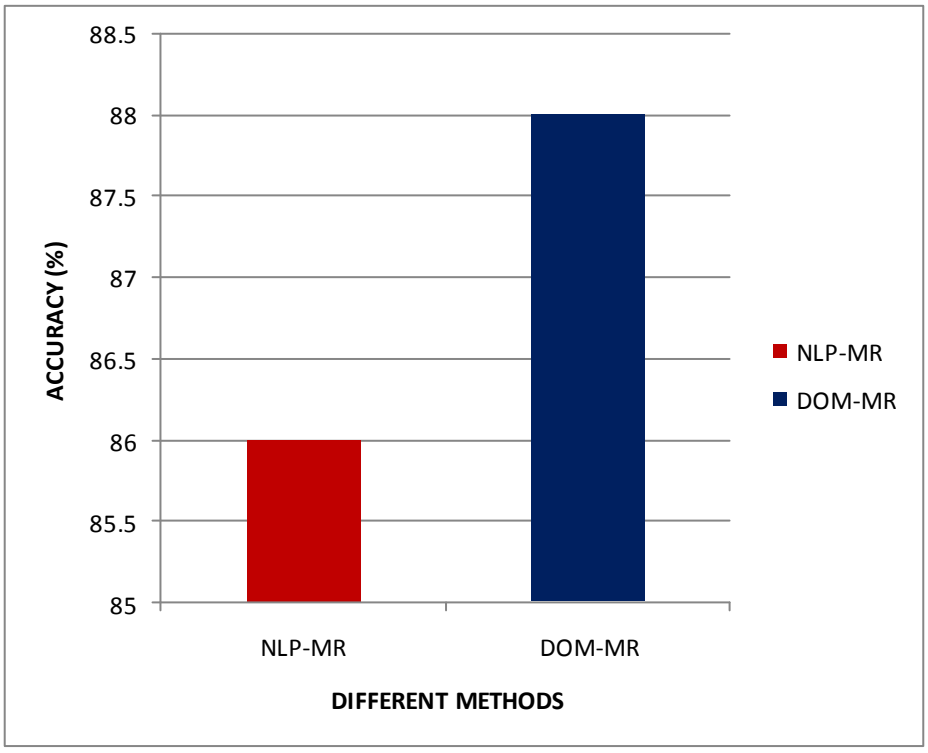

Figure 1.4 Accuracy Results of different Methods

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

The above figure shows performance comparison results of the proposed DOM-MR method is higher than the existing NLP-MR method in terms of accuracy**.** From the results it concludes that the proposed DOM-MR model produces higher accuracy results of 88% whereas existing NLP-MR method produces only 86% respectively.

REFERENCES

1. Sharma, V. and Yadav, P., 2019. Web Service Discovery Approach Among Available WSDL/WADL Web Component. In *Recent Findings in Intelligent Computing Techniques* (pp. 339-344). Springer, Singapore.
2. Siqueira, W.G. and Baldochi, L.A., 2018, June. Leveraging analysis of user behavior from Web usage extraction over DOM-tree structure. In *International Conference on Web Engineering* (pp. 185-192). Springer, Cham.
3. Nayak, A.S., Kanive, A.P., Chandavekar, N. and Balasubramani, R., 2016. Survey on pre-processing techniques for text mining. *International Journal of Engineering and Computer Science*, *ISSN*, pp.2319-7242.
4. Vijayarani, S., Ilamathi, M.J. and Nithya, M., 2015. Pre-processing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), pp.7-16.
5. Palanichamy, K., Hart, K.A., Mondal, S.P. and Namasivayam, B., Qualcomm Inc, 2015. *Systems and methods for parallel traversal of document object model tree*. U.S. Patent Application 14/075,920.
6. Kim, Y. and Lee, S., 2017, February. SVM-based web content mining with leaf classification unit from DOM-tree. In *2017 9th International Conference on Knowledge and Smart Technology (KST)* (pp. 359-364). IEEE.
7. Zhou, J., Chen, L., Chen, C.P., Zhang, Y. and Li, H.X., 2016. Fuzzy clustering with the entropy of attribute weights. *Neurocomputing*, *198*, pp.125-134.
8. Gosain, A. and Dahiya, S., 2016. Performance analysis of various fuzzy clustering algorithms: a review. *Procedia Computer Science*, *79*, pp.100-111.
9. Dolan, B., GCP IP HOLDINGS I, LLC, 2017. Generation of computer-based discovery avatars based on tokenization of prioritized source data. U.S. Patent 9,740,987.
10. Mattsson, U. and Rozenberg, Y., Protegrity Corp, 2016. Use rule-based tokenization data protection. U.S. Patent 9,430,652.
11. Nguyen, D.Q., Vu, T., Nguyen, D.Q., Dras, M. and Johnson, M., 2017. From Word Segmentation to POS Tagging for Vietnamese. ArXiv preprint arXiv: 1711.04951.
12. Fatma, N., Singh, H.K., Ahmad, S. and Srivastava, P., 2016, November. Locality premised reducer scheduling in Hadoop. In 2016 International Conference System Modeling & Advancement in Research Trends (SMART) (pp. 222-224). IEEE.