UNVEILING NOISE IN THE WEB: AN IN-DEPTH ANALYSIS OF MS-WORD TO WEB PAGE CONVERSION ACROSS POPULAR BROWSERS

Bisma Sultan

Lecturer

Department of computer Science

University of Kashmir, Srinagar, india

Abstract: World Wide Web made the internet well known and pulled in people/associations to effectively trade data. Many web-authoring tools have been delivered to make web pages snappy and in WYSIWYG way. Then again every use of the MS-Word incorporates a choice to save a document as web page. The consistence of standard turns into an issue here and subsequently a noise is said to enter in the page. This noise not just bothers the introduction of the page yet in addition takes additional capacity. This paper exhibits a study on MS-Word documents which were changed over into web pages and run in four famous browsers viz. Google chrome, IE7, Mozilla Firefox and musical drama. The experimental results demonstrate that web pages made with the distinctive alternatives of conversion prompts diverse types of noise. Results additionally show that the noise emerges in view of various browsers reacting distinctively to DOM. Results also show that DOM guidelines helps us to remove noise the noisy elements from web pages.

Index Terms: **Web Page Noise, Web Page Conversion, Browser Compatibility, Document Object Model (DOM), Standardization in Web Authoring**

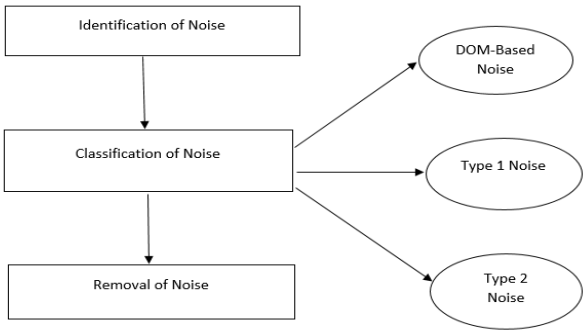
1. Introduction

Since the introduction of World Wide Web, internet has picked up a great deal of prominence. World has achieved a period were associations/ people impart and trade data through web. Various web composing tools have been produced for making web pages quick and in WYSIWYG way. On the other hand, MS-Word utilizes three alternatives to convert a document into web page. At whatever point we have to put a Microsoft document on web, the word document ought to be first changed over to HTML in order to show it on web. MS-Word offers three alternatives to change a word document into web page viz. “Web Page”, “Web Page, filtered” and “Single File Web Page”. While converting a word document into web page utilizing "Web Page" and “Single File Web Page” alternative of MS-Word, MS-Word adds Microsoft specific tags to design the web page with the goal that we keep on editing our pages utilizing full usefulness of word where as only regular tags are being used by MS-Word when a document is saved as web page using “Web Page, filtered” option resulting in smallest size web page compared to then other two types. We can't utilize all the designing elements of MS-Word to alter our page once we converted document into web page using “web page, filtered” type. Furthermore, "Web Page" and "Web Page, filtered" types store HTML document and extra asset records independently where as "Single File Web Page" type makes a solitary MHT document called web archive .MS-Word likewise adds some useless HTML labels to these web pages [5]. All these superfluous, unnecessary, tags are considered as Noise. This Noise takes away the look and feel of a web page, it can seriously harm for web miners by extracting whole document rather than the informative content and also retrieve non relevant results [4].Eliminating noise from web page improve the performance of web page clustering, classification, content mining summarization etc [6].

In this study, various kinds of noises are identified using four well known web browsers such as Google chrome, Internet Explorer 7, Mozilla Firefox and Opera and then classified into different types based on the source of word document. J Query has been used for removing the identified noise with the help of DOM.

2. The Proposed Work

In this study, 50 web pages were created by converting word documents generated from different sources into web pages using three different web page creation options provided by MS-Word. These web pages were tested run on four popular web browsers. The non-compliance of DOM guidelines by these web browsers results in added garbage or undesired output on a web page which is considered as noise. The identified noise is then classified into various categories and finally removed using DOM.



The word documents were generated from the following sources.

- Pdf files were converted into word document using an online pdf-to-word converter.
- Word documents were generated by copy and paste method.
- Word documents were generated by simply writing the data in the MS-Word.

These word documents were converted into web pages with the help of three web page creation types of MS-Word which are "Web Page", "Web Page, filtered" and "Single File Web Page"and run on four popular web browsers.

2.1 Identification of Noise

The analysis of MS-Word created web pages in the four browsers shows that the code associated with these web pages contains Unnecessary <*div*> tags, unnecessary <*a name* > attributes, unnecessary hyperlinks which disturb the web page contents hence are treated as noise. In addition, styles like *mso-spacerun* are not supported by few browsers which result in noisy elements when these web pages are run in such browsers. The position property of all the images present in the web pages has been set as absolute by MS word which results in overlapping of text and images hence taking away the look and feel of the web page and making it difficult to read the contents of web page. Web pages created using "web page" and "single file web page "option include font definitions, style definitions and list definitions in the head tag where as Web pages created using "web page, filtered" include only font and style definitions in head tag.

2.2 **Classification of Noise**

2.2.1 DOM-Based Noise

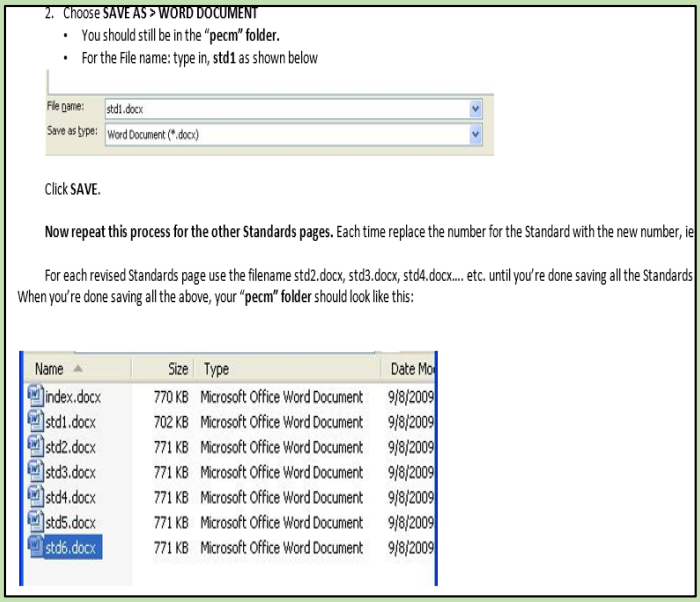
Document Object Model is a structured overview of HTML document created by web browser whenever it receives a raw HTML code from web server. In DOM tree, tags are represented by internal nodes and leaf nodes represents detailed texts, images or hyperlinks [1]. MS-Word creates a web page similar to the way it will be displayed in Microsoft Internet Explorer [2]. Different browsers responding differently to DOM result in web page noise.

Original web page	Same web page run in opera browser
Let us illustrate this with an example: <! DOCTYPE html> <html> <head> <title> title </title> </head> <body> <h1 id="title">OUR WEBSITE! </h1> Link </body> </html> This is a simple document with header and a link. When the parser loads the html file and it locates that first h1 element, it will create an object that represents that h1 element in browser than attaches that new h1 element as a child to its parent object which in this case is the body element itself and it keeps doing this until the browser has complete repres memory in a tree like structure and that tree like structure of related objects in the browser's memory is the Document Object Model- The DOM.	Let us illustrate this with an example: <! DOCTYPE html> <html> Â Â <head> Â Â Â <title> title Â </title> Â Â </head> Â <bod y> Â Â Â Â <h1 id=Â"titleÂ">OUR WEBSITE! </h1> Â Â Â Â &l href=Â" Â" > Link Â </b ody> </html> This is a simple document with header and a link. When the parser loads the html file and it locates t than attaches that new h1 element as a child to its parent object which in this case is the body elemer tree like structure and that tree like structure of related objects in the browserÂ' memory is the Docu

Figure 1 Example of DOM-Based Noise

Figure 1 shows the original web page and the same web page run in opera browser. The noisy elements appear because of *mso- spacerun* style which is used by MS-Word every time we use two spaces between two words or to sentences and is not supported by opera browser hence results in noisy elements when page is run in opera.

Original web page



Same web page run in opera browser

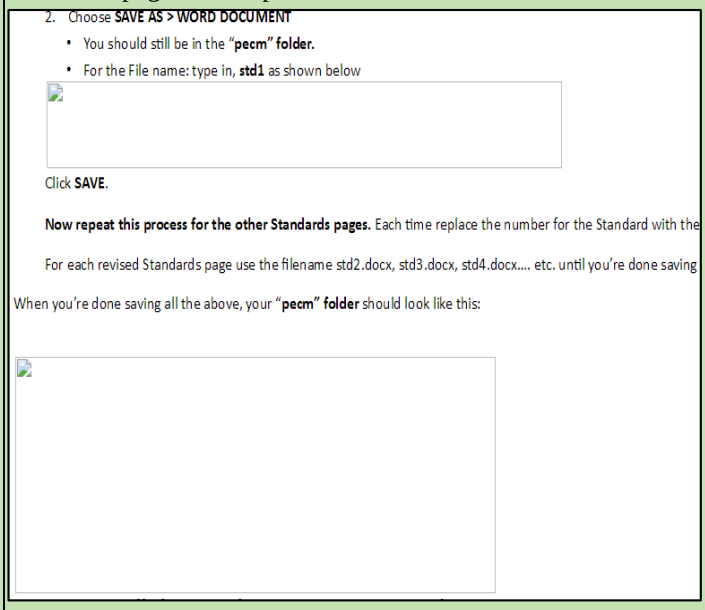


Figure 2 Example of DOM-Based Noise

Figure2 represents a web page and the same run in opera browser. Noise arises because images created by MS-Word using vector markup language(VML) do not open in opera browser as opera do not support VML.

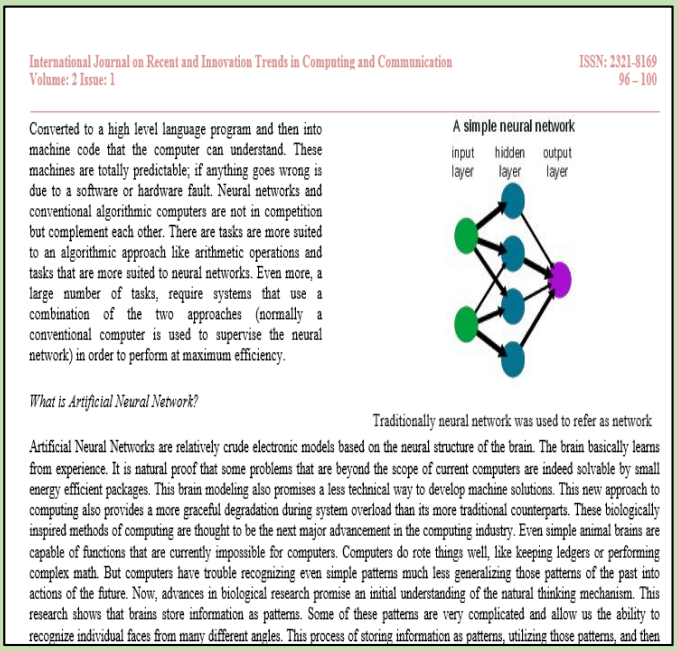
2.2.1.1. Type 1 Noise

Type 1 name was given to the noise that was seen in web pages that were created from documents which have been generated by converting pdf files into word documents. It has been analyzed that the code associated with such web pages contain:

- Unnecessary <*div*> tags and unnecessary <*a name*> attributes that result in noise and redundant data.

- The value of the position property of images has been set as *absolute* which disturbs the presentation of the web page taking away the look and feel of web page. The positioning method used for an element is specified by the position property (static, relative, absolute or fixed) [3]. When an element is specified as *absolute*, it becomes a floating element and any element defined after this element takes its position disturbing the whole environment.

Word Document


International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 1 96 – 100

Converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault. Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

A simple neural network

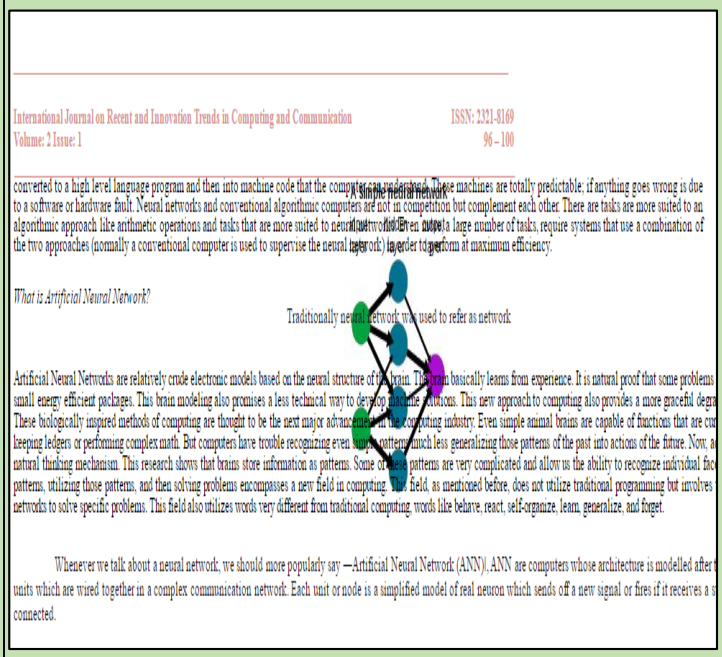
input layer hidden layer output layer

What is Artificial Neural Network?

Traditionally neural network was used to refer as network

Artificial Neural Networks are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems that are beyond the scope of current computers are indeed solvable by small energy efficient packages. This brain modeling also promises a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degradation during system overload than its more traditional counterparts. These biologically inspired methods of computing are thought to be the next major advancement in the computing industry. Even simple animal brains are capable of functions that are currently impossible for computers. Computers do rote things well, like keeping ledgers or performing complex math. But computers have trouble recognizing even simple patterns much less generalizing those patterns of the past into actions of the future. Now, advances in biological research promise an initial understanding of the natural thinking mechanism. This research shows that brains store information as patterns. Some of these patterns are very complicated and allow us the ability to recognize individual faces from many different angles. This process of storing information as patterns, utilizing those patterns, and then

Web Page of the same Document run in Firefox


International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 1 96 – 100

converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault. Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

What is Artificial Neural Network?

Traditionally neural network was used to refer as network

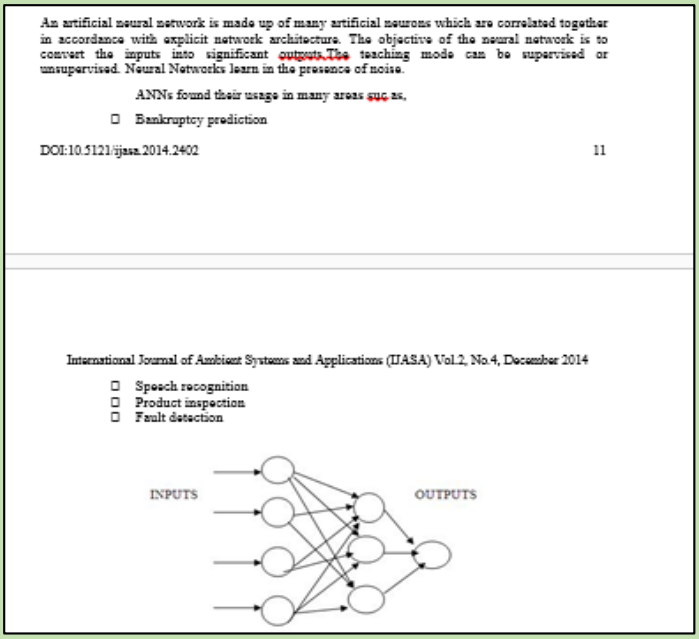
Artificial Neural Networks are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems small energy efficient packages. This brain modeling also promises a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degra These biologically inspired methods of computing are thought to be the next major advancement in the computing industry. Even simple animal brains are capable of functions that are cu keeping ledgers or performing complex math. But computers have trouble recognizing even simple patterns much less generalizing those patterns of the past into actions of the future. Now, a natural thinking mechanism. This research shows that brains store information as patterns. Some of these patterns are very complicated and allow us the ability to recognize individual fac patterns, utilizing those patterns, and then solving problems encompasses a new field in computing. This field, as mentioned before, does not utilize traditional programming but involves networks to solve specific problems. This field also utilizes words very different from traditional computing, words like behave, react, self-organize, learn, generalize, and forget.

Whenever we talk about a neural network, we should more popularly say –Artificial Neural Network (ANN). ANN are computers whose architecture is modelled after t units which are wired together in a complex communication network. Each unit or node is a simplified model of real neuron which sends off a new signal or fires if it receives a s connected.

Figure 3 Effect of setting *absolute* value of position property

Figure 3 shows a word document and the web page of the same document run in Mozilla Firefox. It can be seen how the *absolute* value of position property of images disturbs the environment making it difficult to read the contents of a web page.

Word Document


An artificial neural network is made up of many artificial neurons which are correlated together in accordance with explicit network architecture. The objective of the neural network is to convert the inputs into significant outputs.The teaching mode can be supervised or unsupervised. Neural Networks learn in the presence of noise.

ANNs found their usage in many areas suc as,

- Bankruptcy prediction

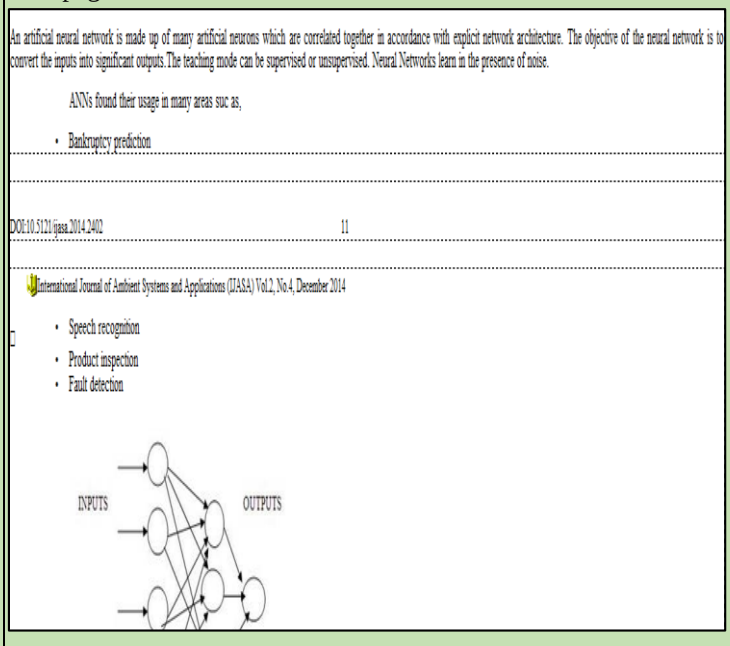
DOI:10.5121/ijasa.2014.2402 11

International Journal of Ambient Systems and Applications (IJASA) Vol.2, No.4, December 2014

- Speech recognition
- Product inspection
- Fault detection

INPUTS OUTPUTS

Web page of the same document


An artificial neural network is made up of many artificial neurons which are correlated together in accordance with explicit network architecture. The objective of the neural network is to convert the inputs into significant outputs.The teaching mode can be supervised or unsupervised. Neural Networks learn in the presence of noise.

ANNs found their usage in many areas suc as,

- Bankruptcy prediction

DOI:10.5121/ijasa.2014.2402 11

International Journal of Ambient Systems and Applications (IJASA) Vol.2, No.4, December 2014

- Speech recognition
- Product inspection
- Fault detection

INPUTS OUTPUTS

Figure 4 Unnecessary <*div*> tags

Figure 4 shows a word document and its web page in which unnecessary tags are presents. It can be seen that a separate division has been created by MS-Word, while converting a document into web page, for a single line.

2.2.1.2 Type 2 Noise

Type 2 name has been given to the noise that was present in web page which were created from document that have been generated by copy and paste method. It has been observed that such web page contain:

Unnecessary Hyperlinks and unnecessary <*a name*> attributes which result in redundancy and noise when such web pages were run in web browsers like Firefox, opera, chrome.

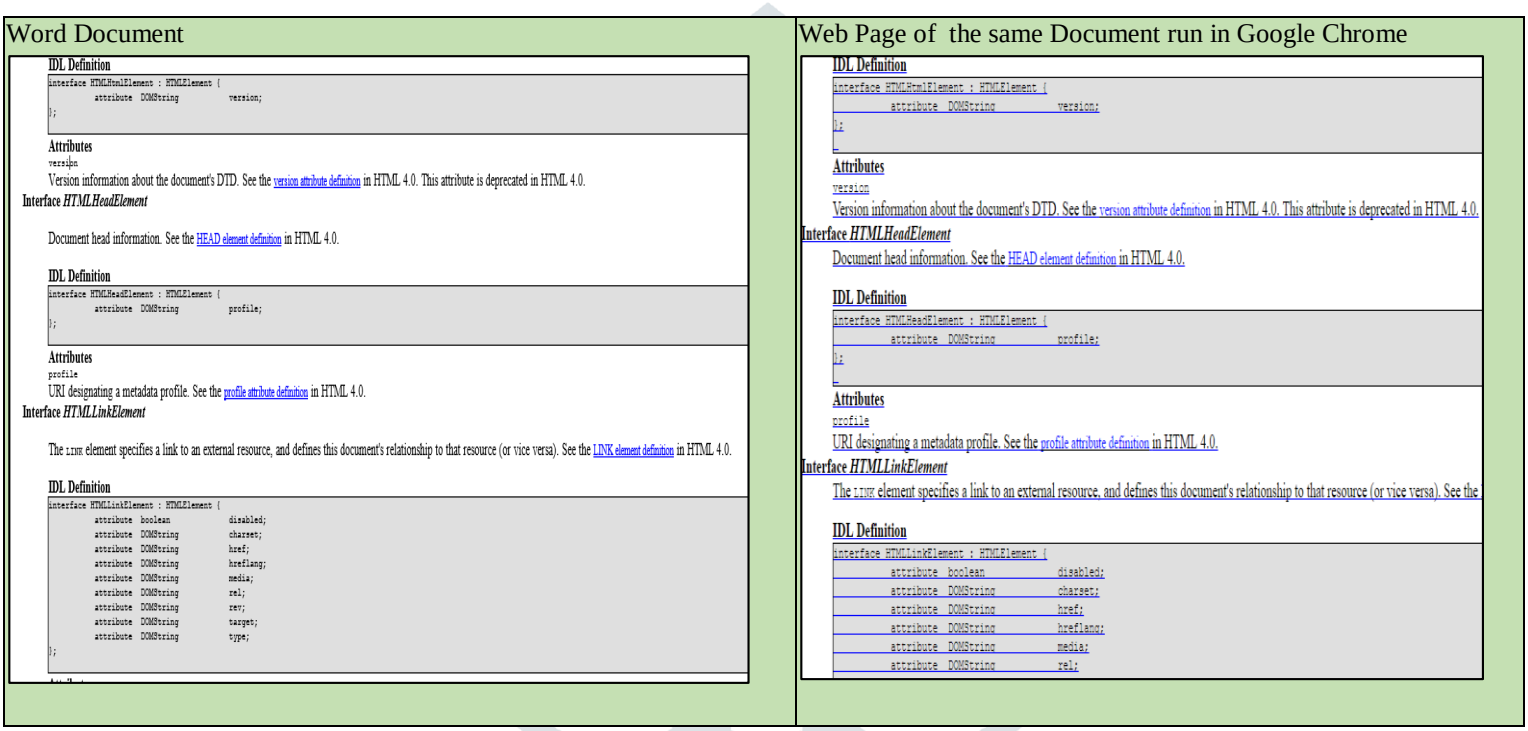


Figure 5 Unnecessary hyperlinks tags

A word document and its web page run in Google chrome is represented in figure 5, it can be clearly seen that how the simple text has been converted into hyperlinks by MS-Word when converting document into web page.

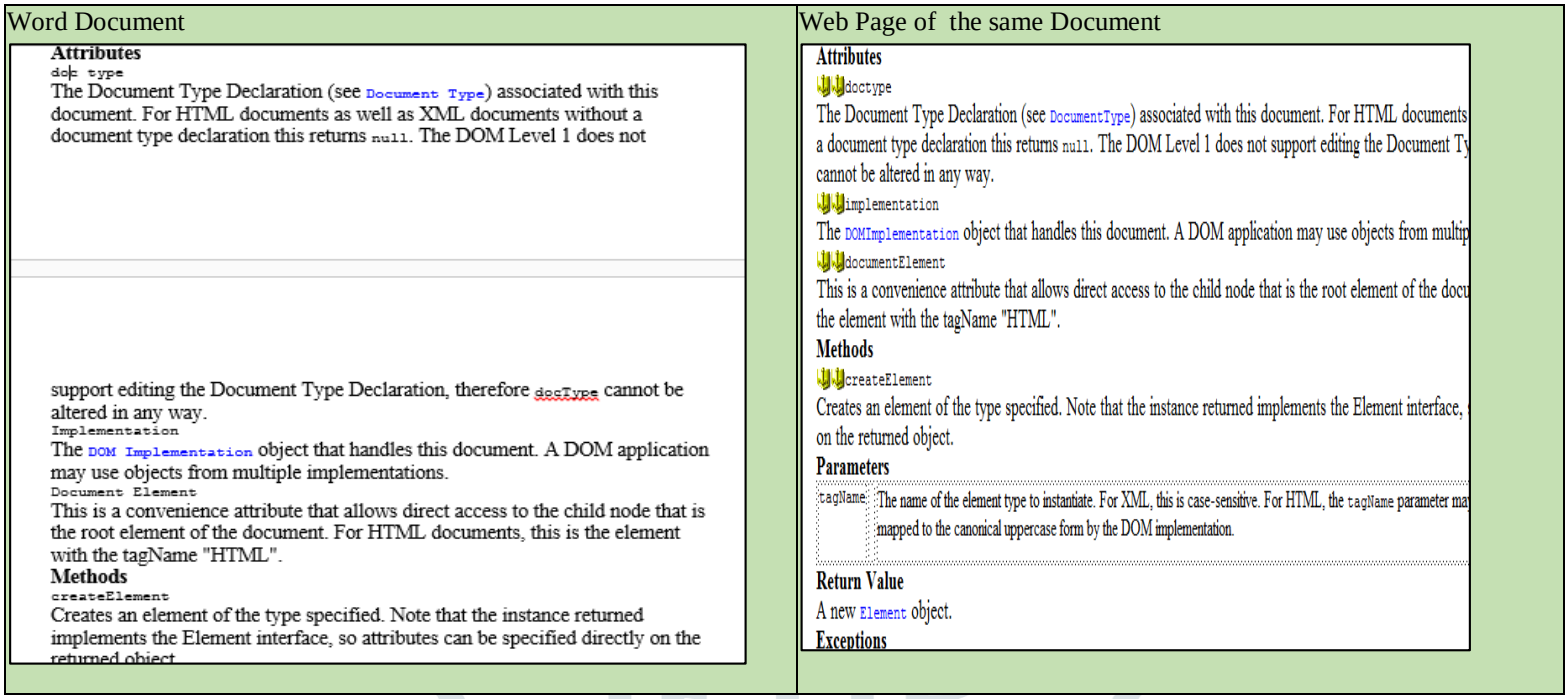


Figure 6 Unnecessary <*a name*> attributes

A simple word document and its web page is shown in figure 6 in which the noise elements have been added because of <*a name*> attributes. This type of noise was seen in both Type 1 and Type 2 Noise.

2.2.2 Removal of Noise

The identified noise was removed with the help of Document Object Model. The code has been written using jQuery. The code traverses through the DOM of a web page, selects the noisy elements and removes them.

Removal of tags.

The proposed step by step procedure for the removal of unnecessary <*div*> tags is given below. The div tags containing characters less than 200 have been removed and the text associated with such elements is appended with the previous element in the DOM.

Algorithm Removediv ()

1. *Traverse through the DOM of the web page and calculate the total number of tags present in the web page.*
2. *For each div calculate the length of text.*
   - **2.1** *If the length of the text is less than 200*
     - **2.1.1** *Append the text with previous element in the DOM.*
     - **2.1.2** *Remove current element.*
3. Calculate the total number of elements.

End

End

End

Firstly, the total number <*div*>elements present in the web page has been calculated by traversing through the Document Object Model of the web page. In the next step, length of the text for each element is calculated. All the spaces are removed and only the length of the

characters is calculated. In the following step, for some <*div*>, if the length of the text is found to be less than 200 then the text associated with that <*div*> is appended with the previous element in the Document Object Model and the current <*div*> is removed. Lastly, new number of <*div*> elements are calculated.

Changing CSS Position property

The step by step procedure for the changing the value of position property from absolute to static is shown below:

Algorithm changeposition ()

1. *Select the elements present inside a element*
2. *For each element select the position property of CSS.*
   - ***1.1*** *Change the position property from absolute to static*

End

End

The first step in the algorithm is to traverse through the DOM and access the span tags present inside element. In the next step, the position property of span element is selected and in the last step, this property is changed from absolute to static.

Removal of <a name > attributes

The Step by Step procedure for the removal of <*a name*> attributes is given below:

Algorithm Removeaname ()

1. *Select <a> elements present inside a element.*
2. *For each <a> element select the name attribute.*
   - ***2.1*** *If name attribute ≠ Null*
   - ***2.2*** *Remove name attribute.*

End

The above mentioned algorithm removes all the <a name> attributes that has been unnecessarily added by MS word while converting a document into web page. In the first step, all the<a> child elements of the parent element have been accessed. In the next step, name attribute of each <a> element is selected and if it is not null, it is removed in the last step.

Removal of empty <o: p> tags

<*o: p*> tags are the Microsoft specific tags. The code for the removal of empty <o: p> tags from the web pages created using MS Word has been developed in this paper. The step by step procedure for the removal of empty <*o:p*> tags is given below:

Algorithm RemoveOP ()

1. *Traverse through the DOM of the web page and calculate the total number of <o: p> tags present in the web page.*
2. *Select the empty <o: p> tags and remove them*
3. *Calculate the number of <o :p> tags*

In the first step, the total number of <o: p> tags present in a web page are calculated by traversing through the DOM. In the next step, empty <o: p> tags are selected and removed. Lastly, the new number of <o: p> tags are calculated.

3. Results and Discussions

3.1 RESULTS OF REMOVAL OF * * AND *<o: p>* TAGS.

Dependent or paired t-test was applied to determine whether there is a significant difference in the number of * *and *<o: p>* tags in the web pages before and after the noise removal. This test was performed for 10 web pages. The formula for dependent t-test is given below:

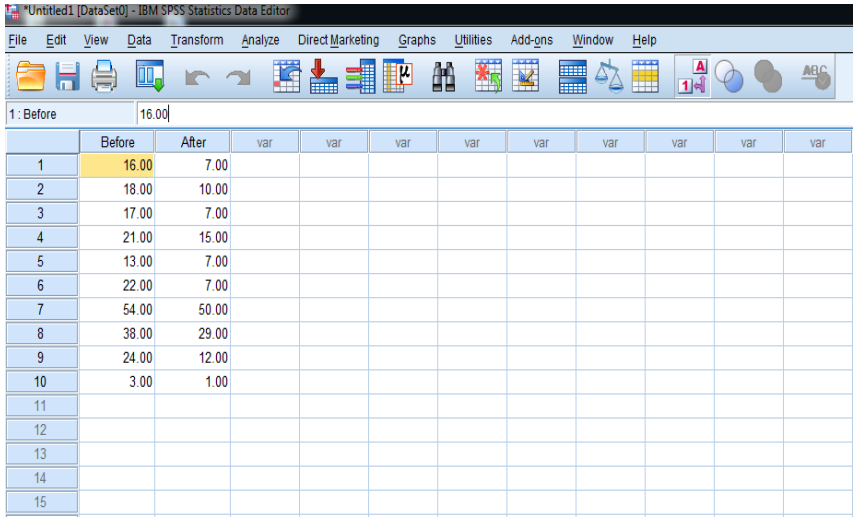
$$t = \frac{\sum d}{\sqrt{\frac{n(\sum d^2) - (\sum d^2)}{n-1}}}$$

Where, $\sum$ d= sum of differences.

Paired T-test for tags.

The unnecessary * * tags added by MS Word in the web pages while converting a document in web page have been removed and the number of * * elements before and after removal were calculated with the help of step by step procedure (*Removediv()*) that has been mentioned above. Statistical Package for the Social Sciences (SPSS) has been used for performing T-test. Before performing the test hypothesis has been set as

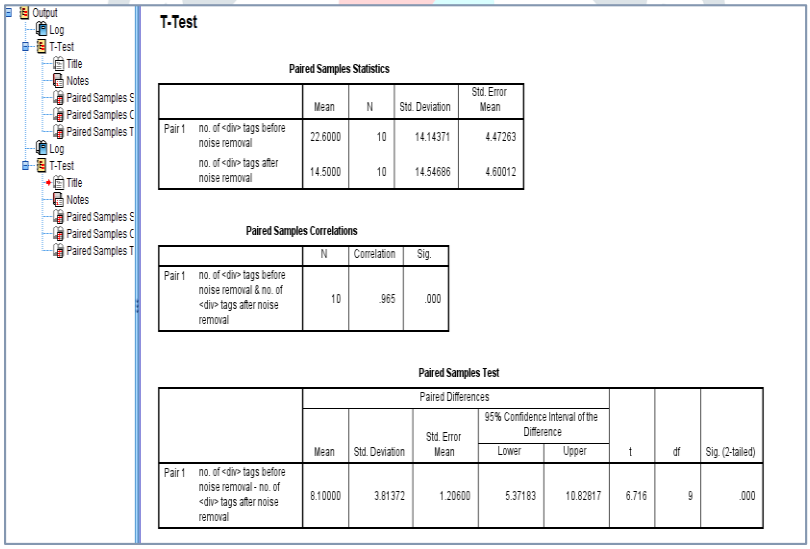
Hypothesis: There is no significant difference between the number of * * tags before and after Noise removal in web pages.



	Before	After	var	var	var	var	var	var	var	var	var
1	16.00	7.00									
2	18.00	10.00									
3	17.00	7.00									
4	21.00	15.00									
5	13.00	7.00									
6	22.00	7.00									
7	54.00	50.00									
8	38.00	29.00									
9	24.00	12.00									
10	3.00	1.00									
11											
12											
13											
14											
15											

Figure 1 Screenshot of <*div*> tag data input into SPSS

Figure 1 gives the data view of SPSS. Column 1 represents the number of <*div*> tags present before the noise removal in web page for 10 web pages and column 2 represents the number of <*div*> tags present after the removal of noise for the same 10 web pages. After entering the data in SPSS, paired t-test was performed on the above data whose output is given below in figure 2



T-Test

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	no. of tags before noise removal	22.6000	10	14.14371	4.47263
	no. of tags after noise removal	14.5000	10	14.54686	4.60012

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	no. of tags before noise removal & no. of tags after noise removal	10	.965	.000

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	no. of tags before noise removal - no. of tags after noise removal	8.10000	3.81372	1.20600	5.37183	10.82817	6.716	9	.000

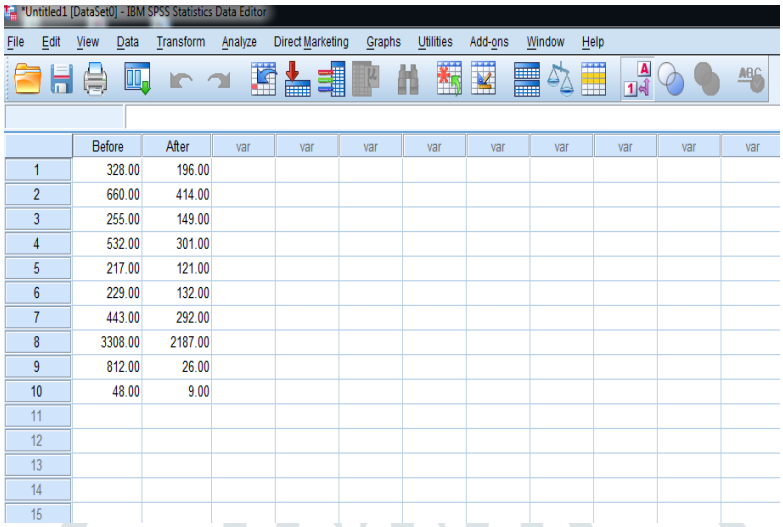
Figure 2 Screenshot of the output of paired T-test for tags.

The Paired Samples Test Box in figure 2 shows that the Sig. (2-Tailed) value is 0.000. This value is less than 0.05. Therefore the Hypothesis is rejected and it is concluded that there is a statistically significant difference in the number of <*div*>tags before and after noise removal.

Paired T-test for <o: p> tags

Empty <*o: p*> tags present in the web pages created by MS-Word were removed and the number of <*o: p*> tags before and after removal were calculated using the step by step procedure (*RemoveOP*) that has been discussed earlier. Dependent T-test was also performed for determining whether there is a significant difference in the number of <*o: p*>tags before and after the noise removal. The Hypothesis has been set as

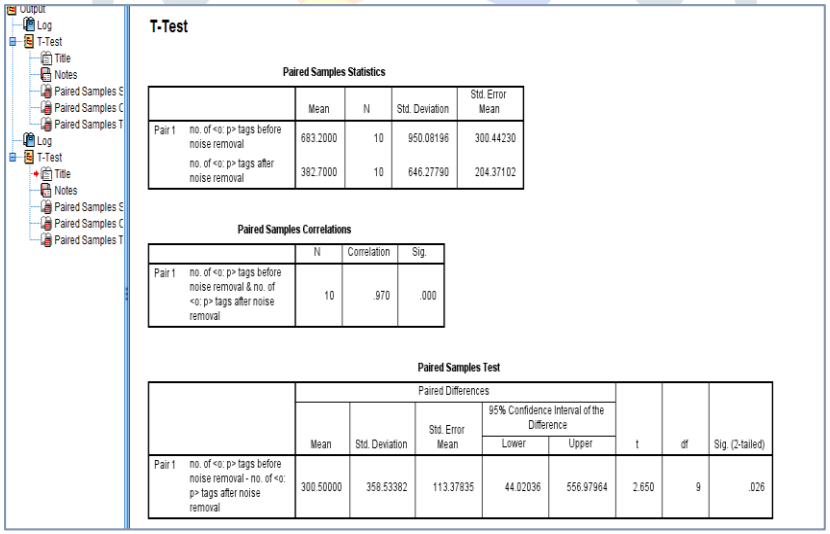
Hypothesis: There is no significant difference in the number of <*o: p*> tags before and after the noise removal in web pages.



	Before	After	var	var	var	var	var	var	var	var	var
1	328.00	196.00									
2	660.00	414.00									
3	255.00	149.00									
4	532.00	301.00									
5	217.00	121.00									
6	229.00	132.00									
7	443.00	292.00									
8	3308.00	2187.00									
9	812.00	26.00									
10	48.00	9.00									
11											
12											
13											
14											
15											

Figure 3 Screenshot of <*o: p*> tag data input into SPSS

In figure 3, Column 1 represents the number of <*o :p*> tags present before the noise removal in web page for 10 web pages and column 2 represents the number of <*o: p*> tags present after the removal of noise for the same 10 web pages. The results of paired t-test performed on the above data is given below in figure 4



T-Test

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	no. of <o: p> tags before noise removal	683.2000	10	950.08196	300.44230
	no. of <o: p> tags after noise removal	382.7000	10	646.27790	204.37102

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	no. of <o: p> tags before noise removal & no. of <o: p> tags after noise removal	10	.970	.000

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	no. of <o: p> tags before noise removal - no. of <o: p> tags after noise removal	300.50000	358.53382	113.37835	44.02036	556.97964	2.650	9	.026

Figure 4 Screenshot of the output of paired T-test for <*o: p*> tags.

The Sig. (2-Tailed) value in the Paired Samples Test Box of figure 4 is 0.026 which is less than 0.05. Therefore the Hypothesis is rejected and it is concluded that there is a statistically significant difference in the number of <*o: p*> tags before and after noise removal.

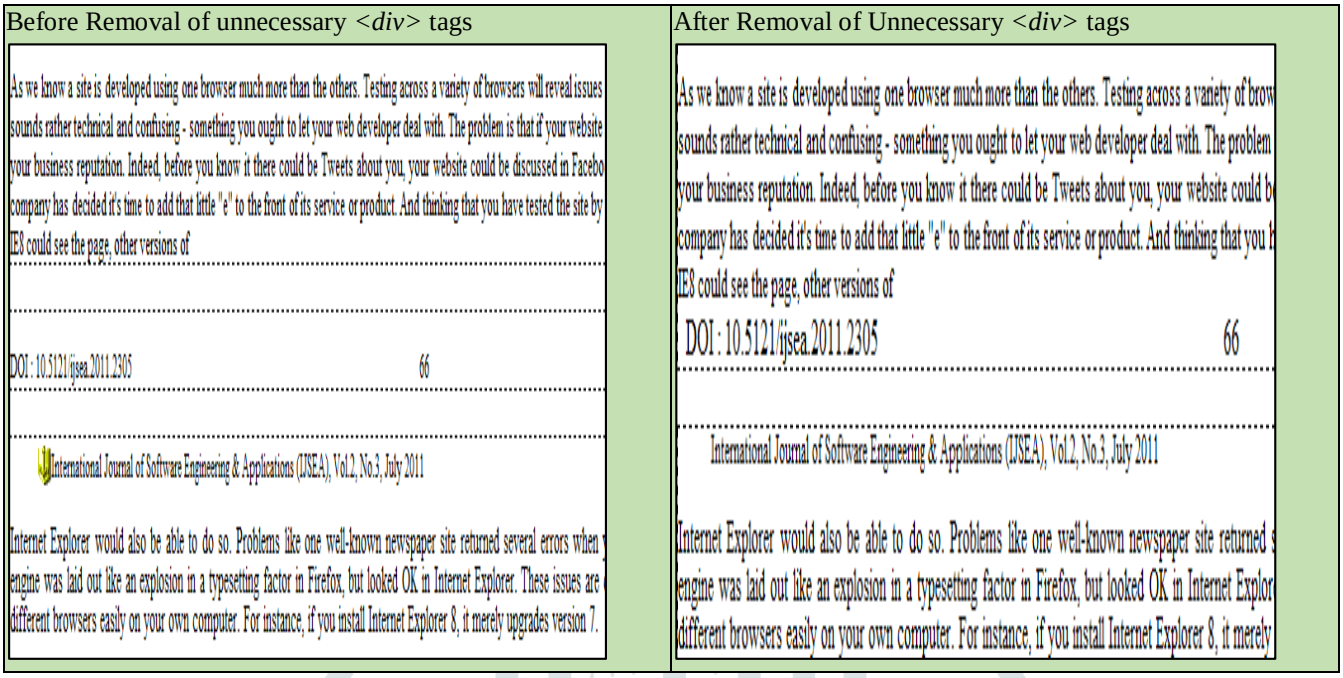


Figure 5 Screenshot of a web page before and after the removal of unnecessary <*div*>tags.

Figure 5 shows a web page before and after removal of unnecessary <*div*> tags. It is quite visible that the <*div*> element containing only a single line of text has been removed and its text has been merged with previous <*div*> element.

3.2 RESULT OF REMOVING <*a name*>ATTRIBUTES.

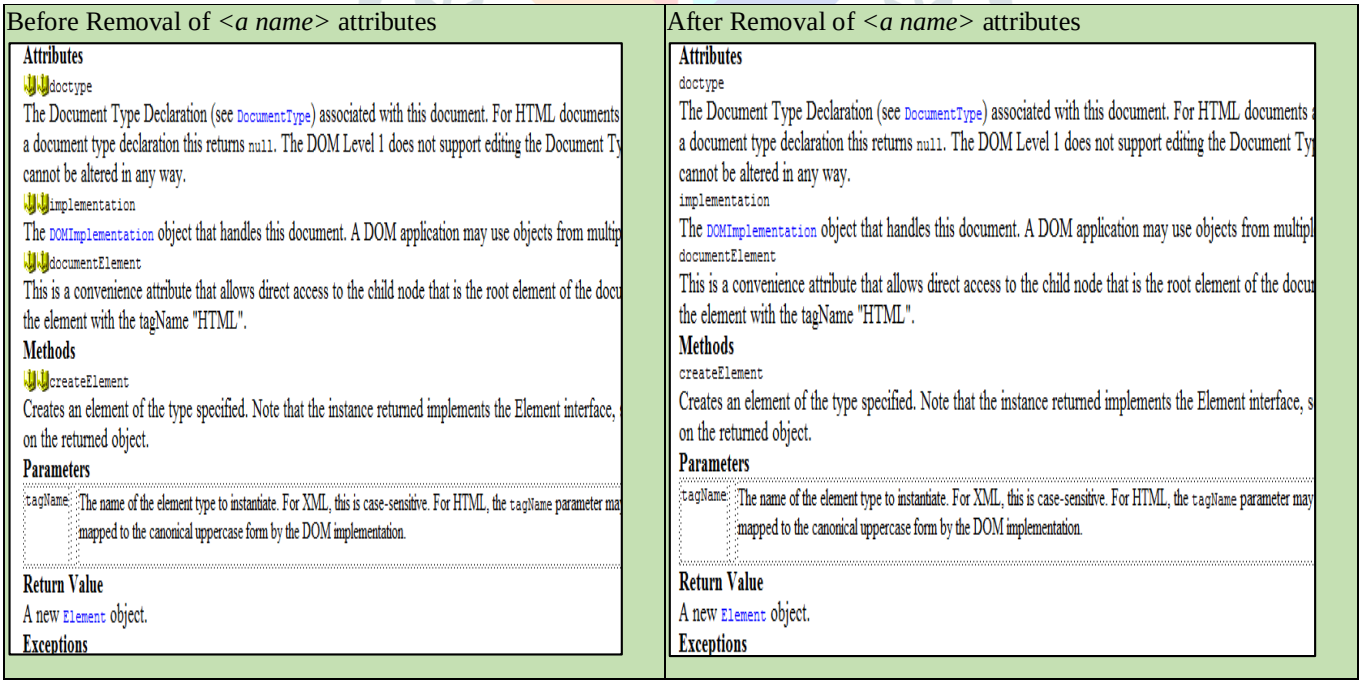


Figure 6 Screenshot of a web page before and after the removal of <*a name*> attributes.

Figure 6 represents a web page before and after removing <*a name*> attributes. It can been seen that noise inserted in the web page by <*a name*>attributes has been removed.

3.3 RESULT OF CHANGING POSITION PROPERTY OF CSS.

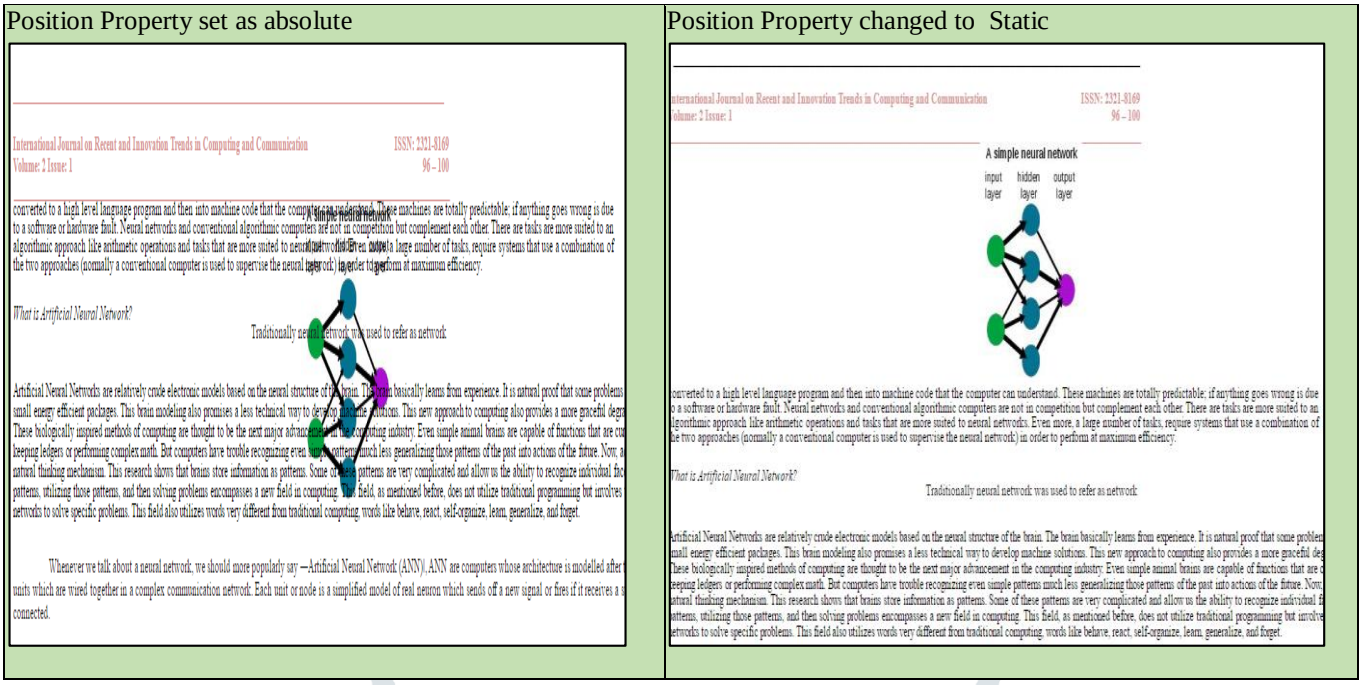


Figure 7 Screenshot of a web page with position property set as absolute and the same changed to static.

Figure 7 represents the web page with position property of images set as absolute and the same web page in which position property has been changed to static. It is quite visible how absolute value of position property affects the web page presentation taking away its look and feel. Changing value of position property to static helps to get rid of this problem.

4. Conclusion and Future Work

4.1 Conclusion

In this paper, noise that arises in web pages created using MS word was identified by analyzing the code associated with these web pages. The identified noise was then classified into three different types based on the source of word document and removed by using the DOM of a web page. The experimental results show that the noise arise because MS word adds some additional tags like *<o: p>*and properties like *mso-spacerun, tab-interval, mso-tab-count, Text-underline* which are not supported by other browsers resulting in noisy elements on web page. It also adds unnecessary, redundant tags like * * tags, *<a name>*attributes, unnecessary hyperlinks which result in code redundancy and make it difficult to read the contents of Web page hence are considered as noise. The position property of all the images present in the web pages was set asabsolute by MS word which results in overlapping of text and images hence taking away the look and feel of the web page. Web page noise also arise because each browsers responds differently to DOM.

The code developed in this paper was able to remove the noise from the web pages successfully. The results of Paired or Dependent T-test for * * and *<o: p>* tags show that there is a significant difference in the number of * * and *<o: p>* tags before and after the removal of noise from the web pages.

In the end, it is concluded that MS-Word is not a good web-authoring tool and the DOM Guidelines followed by MS-Word are not fully supported by other web browsers like Google Chrome, Mozilla Firefox and Opera.

4.2 Future Work

In the future, this work would like to be extended to identify and study noise present in web pages created using other web authoring tools like Microsoft Expression web 4 and Dreamweaver and we would also like to undertake the research on how to classify and remove the noise using neural network techniques so as to make the system more efficient and dynamic.

References

[1] Lan Yi, Bing Liu and Xiaoli Li, "Eliminating Noisy Information in Web Pages for Data Mining," Proceedings of ninth ACM SIGKDD international conference on knowledge discovery and data mining.

[2] https://support.microsoft.com/en-us/help/212270/limitations-when-you-save-a-word-document-as-a-web-page

[3] https://www.w3schools.com/cssref/pr_class_position.asp

[4] Thanda htwe and Khin Haymar Saw Hla, "*Noise removing from web pages using neural network*", IEEE '10: Proceedings of the International Conference on computer and automatic engineering (ICCAE 2010), 2010 pp. 281-285.

[5] Bisma Sultan and Lalit Sen Sharma, "*Identification and Classification of Noise in Converting a Document in Web Page*", International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653 (Online) volume 5 Issue 6, June 2017 pp. 1157-1162.

[6] Maya John and Dr. Jayasudha J.S," *Methods for Removing Noise from Web Pages: A Review",* International Research Journal of Engineering and Technology (IRJET)Volume: 03 Issue: 08 | August-2016.

[7] L. Yi, B. Liu, and X. Li, "Eliminating Noisy Information in Web Pages for Data Mining," in Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2003, pp. 296–305.

[8] C. Ramya, G. Kavitha, and D. K. Shreedhara, "Preprocessing: A Prerequisite for Discovering Patterns in Web Usage Mining Process," ArXiv Prepr. ArXiv11050350, 2011.

[9] S. Dias and J. Gadge, "Identifying Informative Web Content Blocks using Web Page Segmentation," entropy, vol. 1, p. 2, 2014.

[10] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data," SIGKDD Explor Newsl, vol. 1, no. 2, pp. 12–23, Jan. 2000.

[11] M. Jafari, F. SoleymaniSabzchi, and S. Jamali, "Extracting Users'Navigational Behavior from Web Log Data: a Survey," J. Comput. Sci. Appl. J. Comput. Sci. Appl., vol. 1, no. 3, pp. 39–45, Jan. 2013.

[12] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, "User profiles for personalized information access," in The adaptive web, Springer, 2007, pp. 54–89.

[13] P. Peñas, R. del Hoyo, J. Vea-Murguía, C. González, and S. Mayo, "Collective Knowledge Ontology User Profiling for Twitter – Automatic User Profiling," in 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013, vol. 1, pp. 439–444.

[14] S. Kanoje, S. Girase, and D. Mukhopadhyay, "User profiling trends, techniques and applications," ArXiv Prepr. ArXiv150307474, 2015.

[15] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, "Enhancing data analysis with noise removal," IEEE Trans. Knowl. Data Eng., vol. 18, no. 3, pp. 304–319, Mar. 2006.

[16] H. Liu and V. Kešelj, "Combined Mining of Web Server Logs and Web Contents for Classifying User Navigation Patterns and Predicting Users' Future Requests," Data Knowl Eng, vol. 61, no. 2, pp. 304–330, May 2007.

[17] A. K. Santra and S. Jayasudha, "Classification of web log data to identify interested users using Naïve Bayesian classification," Int. J. Comput. Sci. Issues, vol. 9, no. 1, pp. 381–387, 2012.

[18] J. Sripriya and E. S. Samundeeswari, "Comparison of Neural Networks and Support Vector Machines using PCA and ICA for Feature Reduction," Int. J. Comput. Appl., vol. 40, no. 16, pp. 31–36, Feb. 2012.

[19] S. P. Malarvizhi and B. Sathiyabhama, "Enhanced reconfigurable weighted association rule mining for frequent patterns of web logs," Int. J. Comput., vol. 13, no. 2, pp. 97–105, 2014.

[20] X. Wang, B. Chen, and F. Chang, "A Classification Algorithm for Noisy Data Streams with Concept-Drifting," J. Comput. Inf. Syst., vol. 7, no. 12, pp. 4392–4399, 2011.

[21] H. Wang, Q. Xu, and L. Zhou, "Deep Web Search Interface Identification: A Semi-Supervised Ensemble Approach," Information, vol. 5, no. 4, pp. 634–651, Dec. 2014.

[22] F. Hu, M. Li, Y. N. Zhang, T. Peng, and Y. Lei, "A Non-Template Approach to Purify Web Pages Based on Word Density," in Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012, Springer, London, 2013, pp. 221–228