ISSN 2278-3091
Volume 6, No.4, July - August 2017
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse03642017.pdf

A Semantic Framework for Summarizing XML Documents

Hassan A. Elmadany[1], Marco Alfonse[2], Mostafa Aref[3]
Department of Computer Science, Faculty of Computer and Information Sciences,
Ain Shams University, Cairo, Egypt
[1]Hassanelmadany@cis.asu.edu.eg, Enghassanelmadany@gmail.com
[2]Marco@fcis.asu.edu.eg
[3]Mostafa.Aref@cis.asu.edu.eg

ABSTRACT

eXtensible Markup Language (XML) represents data in an efficient way due to its flexibility and the availability to use in various applications. The need to summarize XML documents is increased due to the increasing use of XML in data exchange and representation also due to its difficulty to read and understand. This paper presents an XML Abstractive Summary (XAS) approach to summarize the XML document in a semantic and concise way. The experiments are done using two dataset: IMDB and DBLP. The results has been tested with more than 300000 XML documents. XAS approach decreases the size of the document up to 50 % with average precision and recall 76.5% and 45% respectively.

Key words: XML Summarization, Abstractive Summarization, Ranking, Rich Semantic Graph

1. INTRODUCTION

eXtensible Markup Language (XML) represents a different data in an efficient way due to its flexibility as it can be supported in various applications. With the increasing use of XML in data exchange and representation and difficult to read and understand large and complex XML documents. It is necessary to provide approaches that summarize XML document in a semantic manner. XML summarization has challenges due to [1, 2]:

- **Informativeness:** a unit of information, e.g. tags and text must be informative to the user as its importance in the document as it must be presented concisely to the user.
- **Non-redundancy:** a tag could occur multiple times in a document and each tag is associated with a distinct value. Clearly, it is not important to repeat all occurrences of the tag in the generated summary, but represent it concisely using a single tag.
- **Coverage:** referring to the amount of information rather than data in the XML summary.
- **Coherence:** the context of a tag in terms of its parents or siblings may be important.

The author in [3] categorizes the XML summarization approach based on its content and structure into three (3) main categories:

1) Ranking Approach
2) Schema Approach
3) Compression Approach

In XAS approach, it relies on the ranking approach to summarize the XML document with the help of Rich Semantic Graph reduction technique [4] to get an abstractive summary for the text in each tag to get a concise summary. This paper is organized as follows: section 2 presents Background and related works, section 3 presents the proposed approach, section 4 presents results and finally, the conclusion is reported in section 5.

2. BACKGROUND

In this section, the relevant past methodologies that used to summarize documents to get an abstractive summary are presented. Ramanath, M., & Kumar, K. S. [5, 1] develops an automated framework for summarizing XML documents with respect to memory budget. It summarizes the XML document using two main processes: First, rank the tags and values according to their frequencies that describing how many times the tag occurred in the document. Second, rewrite the selected tags and values to make a readable summary.

Lv, T., & Yan, P. [6, 1] allows another concept in summarizing XML documents based on a predefined schema. The process of summarizing XML document can be done as First, remove the redundant data using both abnormal functional dependencies and a given schema structure. The second step is to classify the tags into two categories: key or non-key. For key tag and its value will remain as it in the generated summary, but for the other category, it will be summarized according to their occurrence in the original document. Finally, the value in tags will be summarized, but in the case of the same tag with multiple values it only uses the first tag value and for long tag values it will be summarized with respect to a given length. This approach provides a semi-structured summary that allows the help of the user to get some parameter that must be given.

Pushpak Bhattacharyya [7] uses WordNet to summarize a text by extracting sub graph for the document from the WordNet.

I. Fathy, D. Fadl, M. Aref [8, 9] presents a new semantic representation called Rich Semantic Graph (RSG). The method uses a domain ontology in which the information needed in the same domain of RSG included.

3. PROPOSED APPROACH

XAS Approach stands for XML Abstractive Summary. It generates a concise and readable XML summary [10]. Figure(1) illustrates the processes of generating the semantic XML summary from the original one using XAS approach. XAS approach consists of four (4) main processes:

1) Remove Data Redundancies Process
2) Ranking Process
3) Threshold Process
4) Summarization Process

A. Remove Data Redundancies Process

Redundancy means that a tag could occur multiple times in a document and each tag is associated with a distinct value. Clearly, it is not important to repeat all occurrences of the tag in the generated summary but represents it concisely using a single tag. The output of removing data redundancies is non-redundant XML document. XML document contains redundant information due to bad schemes which include XML Schema and Document Type Definition (DTD). Redundancies may cause waste storage space also operation anomalies in XML datasets.

There are two types that cause XML data redundancies [6]: Functional dependencies [11] (Normalization Theory, which determines if the XML schema is good or not) and Structure which refers to dataset itself. So the process can be divided into two main sub process:

1) Removing XML data redundancies by Functional dependencies [11].
2) Removing XML data redundancies by Structure [6].

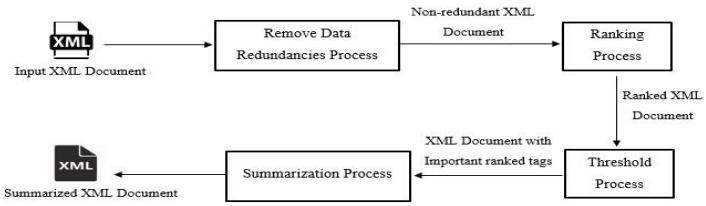

Figure 1: XAS Methodology

B. Ranking process

This is the second process in XAS approach in which the tags are ranked according to their frequencies that describing how many times the tag occurred in the XML document. There are many methods which used to rank tags. The author uses diverse text value [5] method which ranks the text values due to its importance in the document according to their occurrence or frequency that is how many times the value has occurred. It is useful for some kind of text values such as names. This method can be viewed under either corpus or the document belongs. If multiple text values that need to be ranked and only a few occur more than once in the document, then rank these few using their occurrence counts in the document. However, to rank the remaining text values, make use of their counts in the corpus.

C. Threshold Process

The input of this process is the ranked XML document and the output is the XML document with important tags. Here there is a threshold to calculate the importance of the tag that is the tag is important if its ranking frequency is greater than or equal to the half of the total number of XML documents tested.

D. Summarization Process

It aims to generate an abstractive summary. It is the main core process in XAS approach. The input is the text data inside the tag to be summarized and the output is an abstractive summary for these inputs. The summarization module includes three (3) main phases:

1) Creation of rich semantic graph phase.
2) Reduction of the graph phase.
3) Generate summary for the reduced graph phrase.

The first phase is to create a rich semantic graph (RSG) [4] in which it can catch the semantics that lies behind the words, sentences, and paragraphs. It creates nodes for each concept. Each node is enhanced with its attributes. It includes the node's value and the type of the node such as noun and verb. E.g. in the case of verb node type, the attributes could by its subject, objects, place, adverb, etc.

This phase accepts input text, analyze it, and apply pre-processing steps such as tokenization, filtration, POS tags [14], Name Entity Recognition (NER) [15], and syntax analysis. Then it builds a rich semantic graph (sub-graph) for each sentence. These subgraphs are merged into one rich semantic graph which represents the semantics of the whole text. This phase includes main steps as Moawad et al's design [12] [13]:

1) Pre-processing module
2) Word Sense Instantiation.
3) Concept Validation.
4) Sub-Graph Ranking
5) RSG Generation.

The pre-processing step analyses the input. It generates the tokens and POS Tags. It also locates words into categories that are a predefined e.g. name, location…etc. It creates a graph for each sentence individually. The other step is to merge the sub graphs to create the graph that represents the document as a whole. This is the first step in our approach. The input in this step is the text to be summarized and the output is a pre-processed sentence.

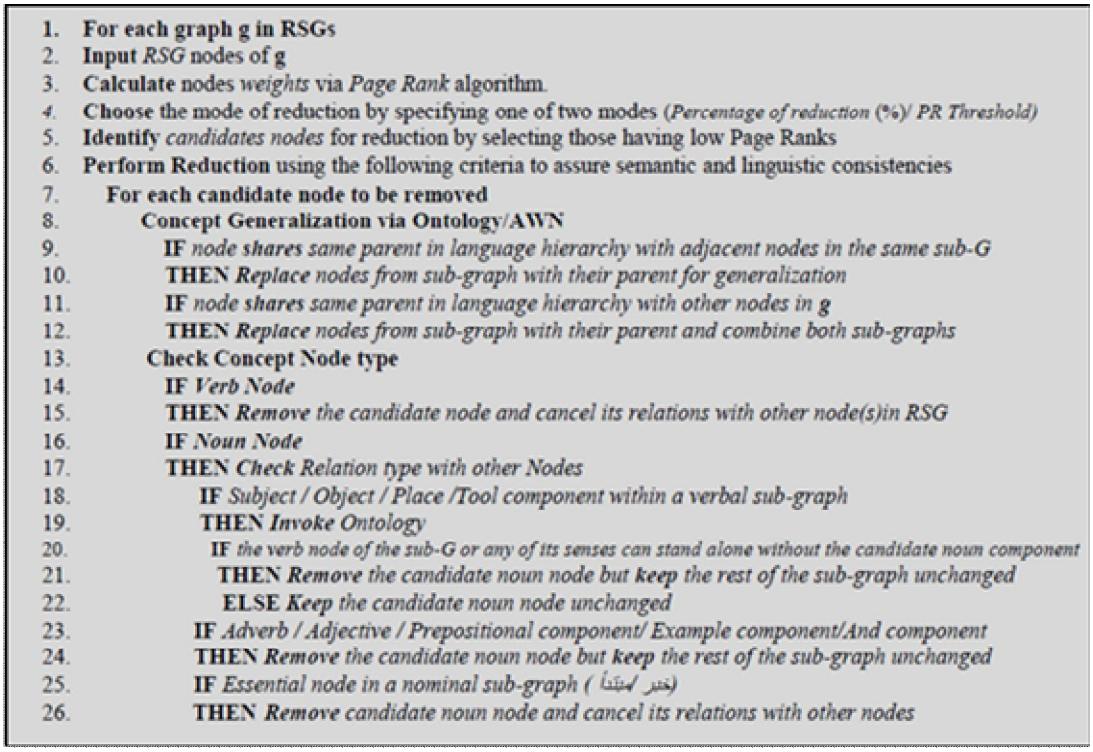

Figure 2: RSG reduction algorithm

After pre-processing is completed, a Word Sense Instantiation process accepts a pre-processed sentence and instantiates senses for each concept using WordNet ontology. The number of senses that can be fetched is reduced by considering only the senses with matching type for each concept.

In the Concept validation process, each concept is validated to reduce the number of the valid senses using the semantic and syntactic relationships which are generated in the pre-processing module.

Sub-Graph Ranking process accepts the valid senses, then rank them according to their relevance from 1 to n, where n is the number of valid senses using equation 1 where $SR_{i,j}$ represents the rank of Sense number j for Concept number i, n stands for the Total number of valid senses for this concept. For the sense rank, a threshold of value 8.5 is chosen to be eliminated any sense with value less than this threshold. The average sense rank, is calculated using equation 2 where ASR_k stands for Average Sub-graph Ranking number k for the sentence, and N is the Total number of concepts in the sentence.

$$SR_{i,j} = \frac{n - j + 1}{n} * 10 \quad (1)$$

$$ASR_k = \frac{\sum_{i=1}^{N} SR_{i,j}}{N} \quad (2)$$

The RSG generation process creates a rich semantic graph from the highest ranked rich semantic sub graphs. It creates a graph for each sentence individually, then merges the sub graphs to create the graph that represents the document as a whole.

The next phase is reduction of the graph. This phase reduced the created rich semantic graph in the previous phase. Figure 2 illustrates the reduction algorithm used. It accepts a rich semantic graph as an input and produces a reduced rich semantic graph as an output. Firstly, the page rank (PR) [4] is calculated for all nodes. The page rank is calculated using equation 3. It evaluates the nodes according to their significance and consequently the importance of its other connected nodes. Then after PR is calculated, the reduction rate is chosen as a PR threshold. Based on the PR threshold, the candidate nodes with PR less than the threshold is calculated and must be removed. However, to accept or reject the removal of the candidate nodes, heuristic rules are applied to ensure the semantic and linguistic consistency. These rules are:

- Generalize the candidate node and the other nodes in case of the same category in the language hierarchy of the WordNet
- Remove the whole sub-graph in case of a verb candidate node
- In the case of a noun candidate, must check its relation with other nodes.
  - If it is a part of a verbal sub-graph and If the main verb of the graph or any of its synonyms can stand alone without the candidate node, then remove the node and

its relation with other nodes, or else keep the node unchanged.

- Otherwise, if it is a part of a nominal sub-graph and if it is an essential part then remove the whole sub-graph, or else remove only the candidate node and its relation with other nodes.

$$PR(N_i) = 1 - d + d * \sum N_j \in Related(N_i)\left(\frac{PR(N_j)}{|Related(N_j)|}\right) \quad (1)$$

where the sets In(Ni)and Out(Nj)are replaced by sets called Related(Ni) and Related(Nj),pointing out the related nodes to Ni and Nj respectively

A good summarized XML document evaluated by the following three standards [6]:

- **Document Size:** the size of the document is considered an important evaluation standard for the generated XML summary. The goal of summarizing XML document is to generate an XML document with an acceptable size compared with the original one so an XML document of smaller size is more readable and useful than a larger one for a human being.
- **Information Content:** a good summary should contain the entire content of the information of the original one. But, it is impossible for the summary document with less size to contain the entire content of the information of the original document which has no redundant information. Although it is difficult to generate perfect XML summarized document as a good summarized document should contain more information in a given size than a bad one.
- **Information Importance:** It is necessary to contain the most important information of the original XML document.

Here to evaluate the approach, according to its size. To achieve this goal, the ratio between the size of the summarized document ($S_{Summarized}$) and the size of the original one ($S_{Original}$) is calculated. This ratio is called Compressed Ratio (CR).

CR is calculated according to the equation 4:

$$CR = 1 - \frac{S_{Summarized}}{S_{Original}} \quad (2)$$

To evaluate the text, there is a problem in establishing what an ideal summary is. The best way to get an ideal summary is to have an expert. To evaluate the text, the standard metrics are used. These metrics are precision and recall. Precision refers to how system summary is correct. Recall measure how the system generates corrected summary with respect to the human model. The recall is defined in [4] as the ratio between the common sentences between human summary and system summary and the total number of sentences in human summary see equation (5). The author in [4] defines the precision as the ratio between the common sentences between human summary and system summary and the total number of sentences in system summary see equation (6).

$$Recall(R) = \frac{no\ of\ common\ sentences}{total\ no\ of\ sentences\ in\ Human\ summa} \quad (5)$$

$$Precision\ (Pr) = \frac{no\ of\ common\ sentences}{total\ no\ of\ sentences\ in\ system\ summa} \quad (6)$$

4. RESULTS

There are two datasets had been used during the implementation. They derived from two sources: DBLP [16] and IMDB [17]. The DBLP stands for Digital Bibliography and Library Project that is a corpus consisted of around 160,000 XML files, each describing a single publication. The IMDB stands for Internet Movies Database that is corpus consisted of approximately 50,000 XML files, each describing a single movie. The only criteria for selecting XML document is to be well-known movies-publication otherwise there were no particular criteria [5]. The other dataset used is IMDB that stands for Internet Movies Data Base. It is an online database of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews, operated by IMDb.com. Table 1 illustrates a description of the data set used in the implementation

Datasets	No of XML Documents	Example Tags
DBLP	323882	Title, Author, year, Journal
IMDB	20,000	Title, director, actor, role, plot

Table 1: Description of Datasets

The experiments are done using XAS system to generate an abstractive summary for the input XML document. The experiments were done using 10% of DBLP dataset. However, the IMDB dataset without text units forms 23% from the total IMDB dataset. Table 2 and table 3 illustrates the result of DBLP dataset and IMDB without text unit dataset where:

- **#XML Docs:** it refers to the number of the input XML documents
- **Size Before:** it refers to the size of the input XML documents in Megabytes
- **Size After:** it refers the size of the summarized XML documents in Megabytes.
- **Total CR**: it stands for the total Compressed Rate which is the ratio of the total size of the summarized XML documents to the total size of the input one. The CR can be calculated according to the equation (3.4)
- **Time**: it refers to the total executed time in minutes.

#XML Docs	Size (MB)		Total Compressed Rate (CR)	Time (Min)
	Before	After		
32000	15.22	10.12	34	20.69

Table 2: DBLP test results

#XML Docs	Size (MB)		Total Compressed Rate (CR)	Time (Min)
	Before	After		
4600	3.04	2.25	26	4.60

Table 3: IMDB without text units test results

Table 4, table 5 and table 6 illustrates the result of IMDB dataset with text unit with respect to the reduction rate of 30%, 40%, and 50% respectively where:

- **#XML Docs:** it refers to the number of the input XML documents
- **Size Before:** it refers to the size of the input XML documents in kilobytes
- **Size After:** it refers the size of the summarized XML documents in kilobytes.
- **Total CR**: it stands for the total Compressed Rate which is the ratio of the total size of the summarized XML documents to the total size of the input one. The CR can be calculated according to the equation (3-4)
- **Tag CR:** it refers to the compressed rate with respect to tags
- **Text CR:** it refers to the compressed rate of the text in text units.
- **Precision (Pr):** a factor to calculate the correctness of the system to generate summary from a system point of view using equation (5)
- **Recall (R):** a factor to calculate the correctness of the system to generate summary with respect to human model using equation (6)

Figure 3 illustrates the evaluation metrics which include precision and recall from the table (4), table (5) and table (6) respectively. The precision and recall are major factors to evaluate the system correctness. The figure shows that whenever the reduction rate is small, the recall is high. The recall is calculated based on the number of common sentences in both system and human summary. So whenever the system can generate the correct summary, the reduction rate decreased. There is an irregular relation between the precision and reduction rate. The precision did not fall quickly when the reduction rate increased.

Figure 4 shows the compressed rate for both tag and text. This factor is the main evaluation factor in XAS system. The compressed rate will be measured to indicate if the XML document has been summarized with respect to its size. It shows that the compressed ratio has a range between 10% and 50%. The figure shows that the text CR is directly related to the reduction rate. Whenever the text CR is increased the reduction rate increased. The tag CR is constant with reduction rate because reduction rate is responsible to reduce the meaning of the text in the text units.

5. Conclusion

This paper presented a new XML summarization approach is called XML Abstractive Summary (XAS) Approach to generate an abstractive summary based on both its structure and data content. The XML Summarization process helps the user to understand the large and complex XML documents by generating a concise summary in less size. The approach discussed in this paper tries to fit the available memory in small size with respect to the size of the original one. It overcomes the XML challenges such as the informativeness as the output summary is an abstractive summary that is a concise and readable to the user with average compressed ratio 50%. Also, it achieves the Non-redundant and Coherence goals by removing data redundancies in form of Functional dependencies and Structure redundancies. XAS approach has an average precision and recall percentage 76.5% and 45% respectively. XAS has been developed using Microsoft Visual Studio 2015 integrated development environment (IDE) and coded with a C# programming language, integrated with a WordNet Library. The performance of XAS approach was measured using a system with CPU 2.40 GHz and 8.00 GB RAM.

#XML Docs	Total Size (MB)		Tag Size (MB)		Tag CR	Text CR	Total CR	PR	R
	Before	After	Before	After					
15400	57.7	37.1	26.3	19.7	25	11	36	33	61

Table 4: IMDB with text units test results with 30% reduction rate

#XML Docs	Total Size (MB)		Tag Size (MB)		Tag CR	Text CR	Total CR	PR	R
	Before	After	Before	After					
15400	57.7	33.3	26.3	19.7	25	17	42	59	46

Table 5: IMDB with text units test results with 40% reduction rate

#XML Docs	Total Size (MB)		Tag Size (MB)		Tag CR	Text CR	Total CR	PR	R
	Before	After	Before	After					
15400	57.7	29.3	26.3	19.7	25	24	49	67	28

Table 6: IMDB with text units test results with 50% reduction rate

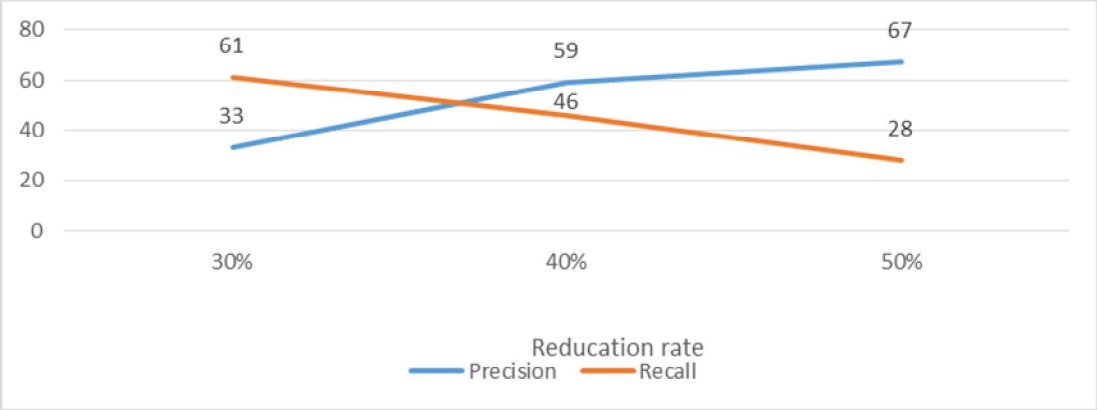

Figure 3: Precision and Recall

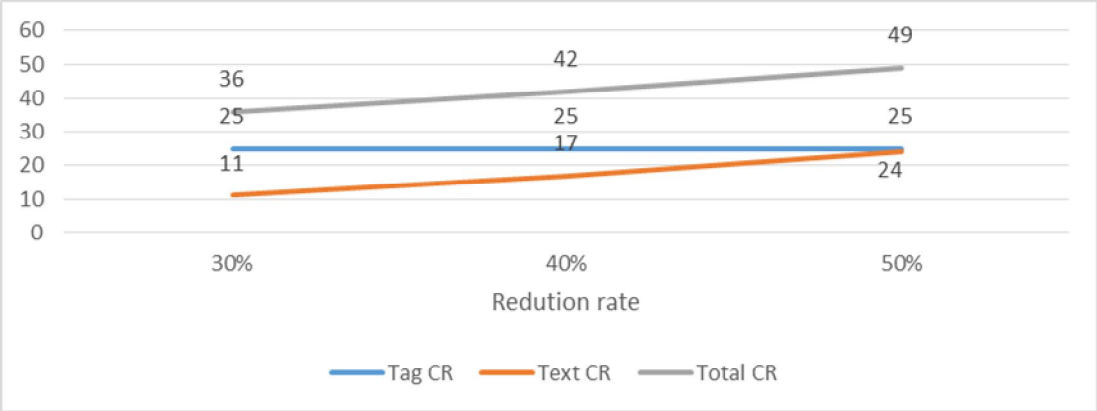

Figure 4: Compressed Rate (CR)

REFERENCES

1. Elmadany, Hassan A., Marco Alfonse, and Mostafa Aref. "XML summarization: A survey." In 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 537-541. IEEE, 2015.
2. Hahn, Udo, and Inderjeet Mani. "The challenges of automatic summarization ", Computer 33.11 (2000): 29-36.
3. Elmadany, Hassan A., Marco Alfonse, and Mostafa Aref. "Semantic-based approaches for XML Summarization." In The Fifteenth Conference. On Language Engineering Organized by Egyptian Society of Language Engineering (ESOLEC'2015). 2015.
4. Sally Saad, Ibrahim Fathy and Mostafa Aref, , "Ontology-Based Approach for Arabic Text summarization", the proceeding of international conference on intelligent computing and information system (ICICIS) (3)
5. Ramanath, M., & Kumar, K. S. (2008, April). "A rank-rewrite framework for summarizing XML documents". In Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on (pp. 540-547). IEEE.
6. Lv, T., & Yan, P. (2013). "A framework of summarizing XML documents with schemas". Int. Arab J. Inf. Technol., 10(1), 18-27.

7. Bellare, Kedar, Anish Das Sarma, Atish Das Sarma, Navneet Loiwal, Vaibhav Mehta, Ganesh Ramakrishnan, and Pushpak Bhattacharyya. "Generic Text Summarization Using WordNet." In LREC. 2004.
8. I. Fathy, D. Fadl, M. Aref, "Rich Semantic Representation Based Approach for Text Generation", the 8th International Conference on Informatics and Systems (INFOS2012), Egypt, 2012.
9. Ibrahim F. Moawad, Mostafa Aref," Semantic Graph Reduction Approach for Abstractive Text Summarization",2012, IEEE
10. Elmadany, Hassan A., Marco Alfonse, and Mostafa Aref. "XML Abstractive Summary Approach". International Journal of Computing & Information Sciences. April 2016.
11. Lv T., Gu N., and Yan P., "Normal forms for XML Documents", Information and Software Technology, vol. 46, no. 12, pp. 839-846, 2004.
12. WordNet [Online] Available at https://wordnet.princeton.edu [Accessed on June 2017]
13. Mostafa Aref, Ibrahim Fathy, Dalia Sayed "Natural language Generation Sentence Planner", International Conference on intelligent computing and information systems ICICIS 2011 p22, Faculty of computer and information science Ain shams University, Cairo, Egypt, 2011.
14. Stanford Parser, Available at http://nlp.stanford.edu:8080/parser/index. [Accessed on March 2017]
15. NER, Available at http://nlp.stanford.edu/software/CRF-NER.shtml [Accessed on March 2017]
16. DBLP, Available at https://kdl.cs.umass.edu/display/public/DBLP [accessed March 2017]
17. IMDB, Available at https://en.wikipedia.org/wiki/IMDb. [Accessed March 2017]