International Journal of Intelligent Computing and Information Science

XML ABSTRACTIVE SUMMARY APPROACH

H. A. Elmadany M. Alfonse M. Aref

Computer Science Department, Faculty of Computer and Information Sciences,
Ain Shams University, Cairo, Egyp

Hassanelmadany@cis.asu.edu.eg Marco@fcis.asu.edu.eg Mostafa.Aref@cis.asu.edu.eg

Abstract: *Text summarization saves both time and effort required to manage a vast amount of information. The need to summarize text is increased. This paper introduces a XML Abstractive Summary (XAS) approach to summarize text in the format of XML document that is called XML summarization. XAS approach is considered a new attempt to produce abstractive summary for the xml document regarding to performance, size and accuracy. The output document is a concise and readable version for the original one.*

Keywords: *XML Summarization, Abstractive Summarization, Ranking, Rich Semantic Graph, Functional Dependancy .*

1. Introduction

Text Summarization is used to manage information by condensing the content of the documents and extracting the facts and topics included which are the most relevant. It can be classified according to the type of summary: extractive and abstractive. However, The Extractive summary attempts to identify the important sections of the text and producing them verbatim. On the other hand, abstractive summary attempts to produce a generalized summary in which conveying in information in a concise way.

eXtensible Markup Language (XML) is one of the standard data representation nowadays. It can be used in various applications as its flexibility and easy to use. So the need to summarize XML document become increasingly an important topic to save time and cost. XML summarization approaches classified into two main categories: Structural summaries and content and structure summaries.

Structural summaries which focus on generating a summary of XML document based on its structural characteristics. On the other hand, content and structural summaries focus on generating XML summary based on the features of the content from the logical structure of the XML document as goal is to provide an XML summary with important information in the original document [1].
XML summarization has challenges due to [1]:

- 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.

This Paper focused on the content and structure XML summarization approaches, looking forward to generate a concise, readable XML summary. So to generate XML summary in a semantic way you must be aware of both its logical and structure or content and structure. The author in [2] 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 our approach, we rely on the ranking approach to summarize the XML document with the use of Rich Semantic Graph 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 work, section 3 presents the proposed approach, section 4 presents a movie case study, and finally the conclusion is reported in section 5.

2. Background& Related Work

In this section we presents the relevant past methodologies that used to summarize the documents to get an abstractive summaries. Ramanath, M., & Kumar, K. S. [٣, 1] develops an automated framework for summarizing XML documents with respect to memory budget. It summarizes 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. [4, 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 summarize, but in 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 [5] uses WordNet to summarize text by extracting subgraph for the document from the WordNet.

I. Fathy, D. Fadl, M. Aref [6] presents a new semantic representation called Rich Semantic Graph (RSG). The method uses a domain ontology in which the information needed in same domain of RSG included.

3. Proposed Approach

XAS Approach stands for XML Abstractive Summary. It generates a concise, readable XML summary. Figure (3.1) illustrates the processes of generating the semantic XML summary from original one.
XAS approach consists of 5 processes:

1) Remove Data Redundancies process
2) Ranking Process
3) Summarization Process
4) Evaluation Process
5) Refinement 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 that contains no redundant data. XML document contains redundant information due to bad schemas which includes 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: Functional dependencies [7] (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 [7].
2) Removing XML data redundancies by Structure [4].

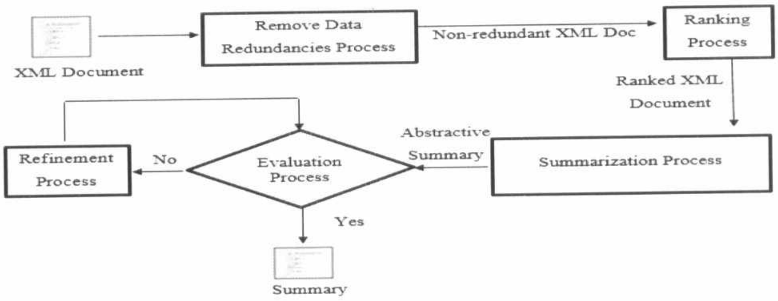


Figure. 3.1: XAS Methodology

The second Process is ranking process in which rank the tags according to their frequencies that describing how many times the tag occurred in the document. There are many methods which used to rank tags. The author uses diverse text value [3] 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 been occurred. It is useful for some kind of text values such as names. This method can be viewed under either corpus or the document belong. 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.

Summarization Process aims to generate 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. Figure (3.2) illustrates the phases for the summarization process

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

The first phase is to create a rich semantic graph. This phase include main steps such as
1) Pre-processing Step
2) Merging sub-graphs Step

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

1) Tokenization
2) Filtration
3) Name Entity Recognition
4) Syntax Analysis

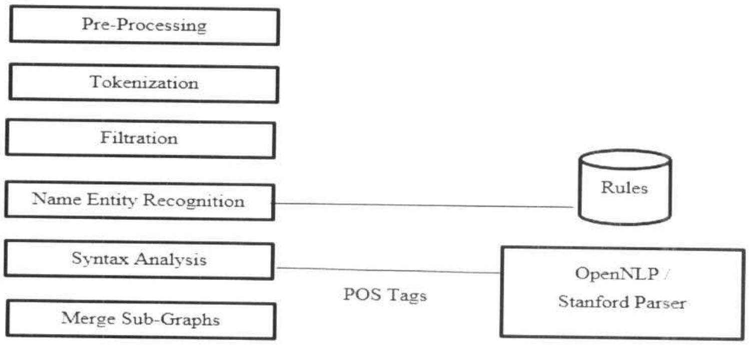


Figure. 3.2: Creation of Rich Semantic Graph

Tokenization gets the text and generates the token for each sentence then do filtration to filter out special character. This step performs by dividing the sentence into tokens by searching for a space after each word. The filtration process is filter out the special characters e.g. *&^/$.#@,';+{}[].

Name Entity Recognition locates the tokens into a categories that are predefined e.g. location, name….etc. using Stanford Parser tool. This tool is available for free or by using Open NLP tool. Syntax Analysis with the use of Stanford parser tool to parse a sentence to get its syntax analysis and for each word, produces POS Tag or Part Of Speech Tagger which is a software that accept text and assign part of speech for each word e.g. Noun, Adjective, and Verb…etc.

The last step in pre-processing modules is to generate the rich semantic graph for each sentence .now merge the sub-graphs to create the rich semantic graph for the whole document. Sentences are represented as nodes.to connect these nodes with each other edges are used. These edges related to similarity relation. Then using a different similarity criteria. The two sentence similarity is calculated and scored. For each sentence, sub-graphs are merged to construct the final rich semantic graph for the whole document. The second phase is done based on some rules (see Figure (3.3)) to generate the reduced RSG by merging, deleting graph nodes. These rules based on some factors

1) Graph node type
2) Sematic relation
3) Similarity or dissimilarity between graph nodes

Using the rules shown at figure (3.3) the graph is reduced then final summary is generated from reduced graph. This phase generates the abstractive summary from the reduced RSG. The sentences are merging with the help of rules and final summary can be generated.

The fourth process in XAS approach is evaluation process. It is the main step in our approach. However, generating a good summary involves satisfying two important goals which are the Maximum coverage and the Minimum space which are contradictory: the larger amount of content have to be included in the generated summary the larger size of that summary, The larger size of the generated summary the lesser its utility to the user so a good balance between the summary size and its coverage is required. A good summarized XML document can be evaluated by the following three standards [4]:

1) 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 comparing with the original one so an XML document of smaller size is more readable and useful than a larger one for a human being.
2) 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.
3) Information Importance: It is necessary to contain the most important information of the original XML document.

So if the generated summary fails to achieve the evaluation methods if will go the next step to be refine otherwise it will accept as the summary. Refinement process is the last process in XAS approach. The summary is iteratively refined by eliminating the least important text value of the least important tag from the current summary until the desired size is reached. The summary is refined to get smaller summaries that can fit the available memory budget. There are two special cases:

1) The tag and its text value which is a correlated value cannot be removed in isolation.
2) The text values proportions that corresponding to the tags need to be maintained aiming for higher coverage of tags and text.

Rule 1. IF	SN1 is instance of noun N	And
	SN2 is instance of noun N	And
	MV1 is similar to MV2	And
	ON1 is similar to ON2	
THEN		
	Merge both MV1 and MV2	And
	Merge both ON1 and ON2	
Rule 2. IF	SN1 is instance of subclass of noun N	And
	SN2 is instance of subclass of noun N	And
	{[MV11, ON11],..[MV1n, ON1n]}	is similar to
	{[MV21, ON21],..[MV2n, ON2n]}	
THEN		
	Replace SN1 by N1 (instance N)	And
	Replace SN2 by N2 (instance N)	And
	Merge both N1 and N2	
Rule 3. IF	SN1 and SN2 are instance of noun N	And
	MV1 is instance of subclass of verb V	And
	MV2 is instance of subclass of verb V	And
	ON1 is similar to ON2	
THEN		
	Replace MV1 by V1 (instance V)	And
	Replace MV2 by V2 (instance V)	And
	Merge both V1 and V2	And
	Merge both ON1 and ON2	

Figure 3.3 : Examples for Reduction Rules [8]

4. A Case Study

A case study about "Students" is presented here in details. To get an abstractive summary for this case study to make it more readable and concise. Figure (٤.1) illustrates the input xml document with redundant data such as class, takenby, student, teacher tags and figure (4.2) illustrates the XML tree for the input. Firstly, check for redundancy to be removed. So the xml input will be redesign to be ready for the next step. Figure (4.3) illustrates the XML document after removing functional data redundancies [2].

Ranking step is used in case of corpus so tag may be occurred many times, so tags have been ranked according there frequencies. The main step in our approach is the summarization step where the output of this step is the abstractive summary needed so the length of text inside each tag must be checked. The <Info> tag needed to be summary to get its abstractive meaning. It consists of 8 sentences and contains 55 words. The preprocessing module consists of four (4) steps:

1) Tokenization
2) Filtration
3) Name Entity Recognition
4) Syntax Analysis

Applying the preprocessing module and creating semantic graph for each sentence then merge the graphs to create the sub graph for the whole input. For single sentence such as the following one:
Hassan ElMadany is a master student. The output for the tokenization and filtration steps is: "Hassan", "ElMadany", "is", "a", "Master", "Student".

The approach breaks sentence into tokens by searching for space after each word the filtration step removes the special characters in the sentence such as (.) Now it's the time to apply the Name Entity Recognition step which locates the tokens into a categories that are predefined e.g. location, name….etc. using Stanford Parser tool.

The output for the NER step is: "Hassan Person", "ElMadany Person". Syntax analysis is done with the using of Stanford parser tool to parse a sentence to get its syntax analysis and for each word, produces POS Tag or Part Of Speech Tagger which is a software that accept text and assign part of speech for each word e.g. Noun, Adjective, and Verb…etc. The output of syntax analysis is:
Hassan/NNP ElMadany/NNP is/VBZ a/DT master/NN student/NN

Creating a semantic graph for the whole document based on the logical form triples subject– predicate– object (SPO). There are two steps to generate the semantic graph [9]:

1) Apply deep syntactic analysis to document sentences and extract logical form triples this step is called Syntactic analysis.
2) Merge the resulting logical form triples into a semantic graph and analyze the graph properties.

Now applying the reduction rules (Figure (3.3)) on the rich semantic graph to get the abstractive summary as the summary for info tag is:

<Info> Hassan ElMadany and Hanan Hasan are master students. They published papers at international conferences. Hanan also published paper at a journal. Hassan engaged with Hanan Hasan. </Info>

The evaluation process plays a vital role in our approach. It makes the decision to generate the summary or try to get a smaller one. Here to evaluate the approach according to its size. To achieve this goal, the ratio between the size of the summarized document and the size of the original one is calculated. If it passes 60% so it will be accepted as a summary. Otherwise, the generated summary must be refined to enhance it. In this case study the size of the original document is 1.49 KB (1,533 bytes) and the size of the summarized one is 957 bytes (957 bytes). The ratio is calculated according to the equation [1]

$$R_S = S_{Summarized} / S_{Original} \qquad (1)$$

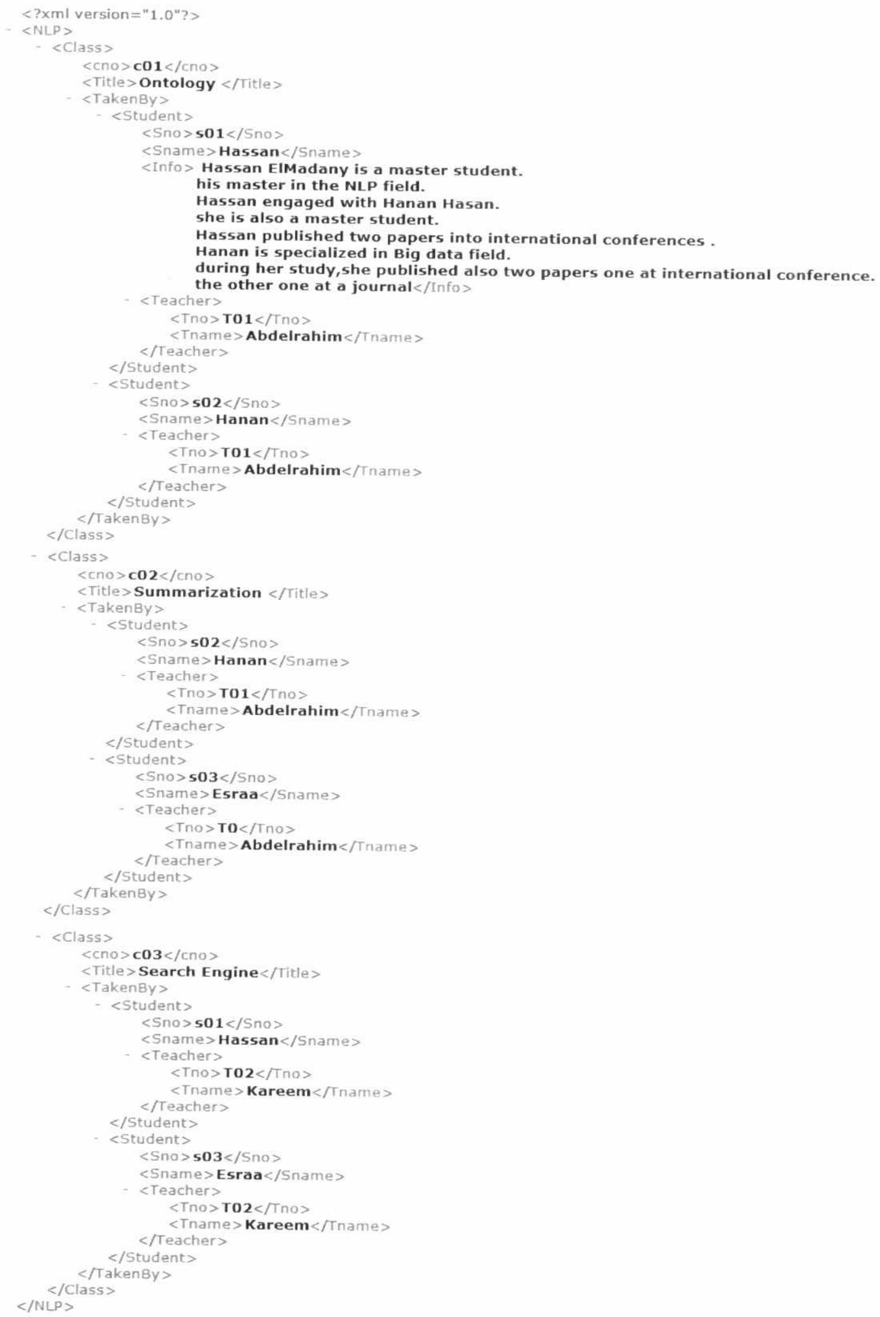
and the ratio will be 62.43%. In case of failure the refinement process has been used to eliminate the tags with low frequencies. The output of the refinement process is an XML document with smaller size than the failed one. This XML document will be evaluated as discussed above.

5. Conclusion

We presented this paper to highlight 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 .Also it achieves the Non-redundancy and Coherence goals by removing data redundancies in form of Functional dependencies and Structure redundancies.

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. 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.
3. 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.
4. Lv, T., & Yan, P. (2013). "A framework of summarizing XML documents with schemas". Int. Arab J. Inf. Technol., 10(1), 18-27.
5. 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.
6. 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.
7. Lv T., Gu N., and Yan P., "Normal forms for XML Documents," Information and Software Technology, vol. 46, no. 12, pp. 839-846, 2004.
8. Ibrahim F. Moawad, Mostafa Aref," Semantic Graph Reduction Approach for Abstractive Text Summarization" 2012 IEEE
9. Munot, Nikita and Govilkar, Sharvari S. "Conceptual Framework for Abstractive Text Summarization ".In International Journal on Natural Language Computing (IJNLC) Vol. 4, No.1, February 2015



```
  <?xml version="1.0"?>
- <NLP>
  - <Class>
      <cno>c01</cno>
      <Title>Ontology </Title>
    - <TakenBy>
      - <Student>
          <Sno>s01</Sno>
          <Sname>Hassan</Sname>
          <Info> Hassan ElMadany is a master student.
                 his master in the NLP field.
                 Hassan engaged with Hanan Hasan.
                 she is also a master student.
                 Hassan published two papers into international conferences .
                 Hanan is specialized in Big data field.
                 during her study,she published also two papers one at international conference.
                 the other one at a journal</Info>
        - <Teacher>
            <Tno>T01</Tno>
            <Tname>Abdelrahim</Tname>
          </Teacher>
        </Student>
      - <Student>
          <Sno>s02</Sno>
          <Sname>Hanan</Sname>
        - <Teacher>
            <Tno>T01</Tno>
            <Tname>Abdelrahim</Tname>
          </Teacher>
        </Student>
      </TakenBy>
    </Class>
  - <Class>
      <cno>c02</cno>
      <Title>Summarization </Title>
    - <TakenBy>
      - <Student>
          <Sno>s02</Sno>
          <Sname>Hanan</Sname>
        - <Teacher>
            <Tno>T01</Tno>
            <Tname>Abdelrahim</Tname>
          </Teacher>
        </Student>
      - <Student>
          <Sno>s03</Sno>
          <Sname>Esraa</Sname>
        - <Teacher>
            <Tno>T0</Tno>
            <Tname>Abdelrahim</Tname>
          </Teacher>
        </Student>
      </TakenBy>
    </Class>
  - <Class>
      <cno>c03</cno>
      <Title>Search Engine</Title>
    - <TakenBy>
      - <Student>
          <Sno>s01</Sno>
          <Sname>Hassan</Sname>
        - <Teacher>
            <Tno>T02</Tno>
            <Tname>Kareem</Tname>
          </Teacher>
        </Student>
      - <Student>
          <Sno>s03</Sno>
          <Sname>Esraa</Sname>
        - <Teacher>
            <Tno>T02</Tno>
            <Tname>Kareem</Tname>
          </Teacher>
        </Student>
      </TakenBy>
    </Class>
  </NLP>
```

Figure. 4.1: Input XML Document

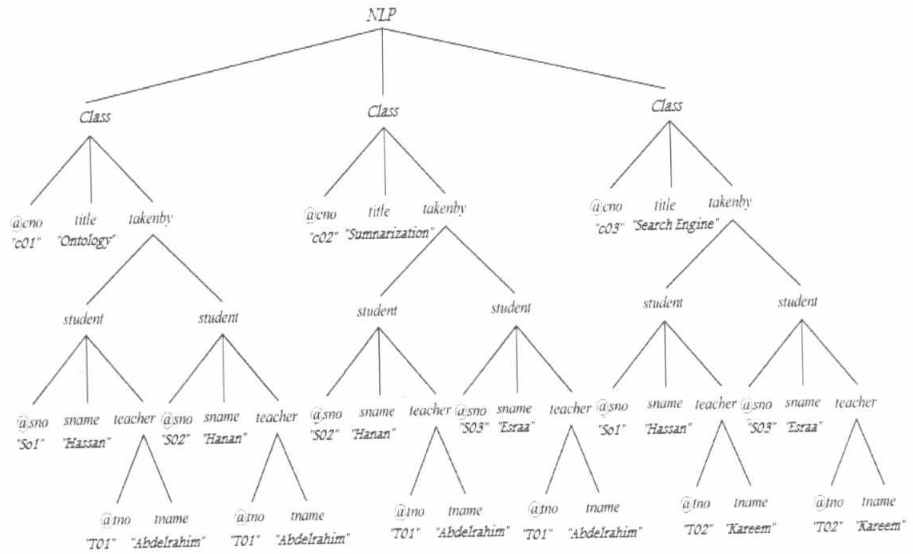


Figure. 4.2: XML Tree

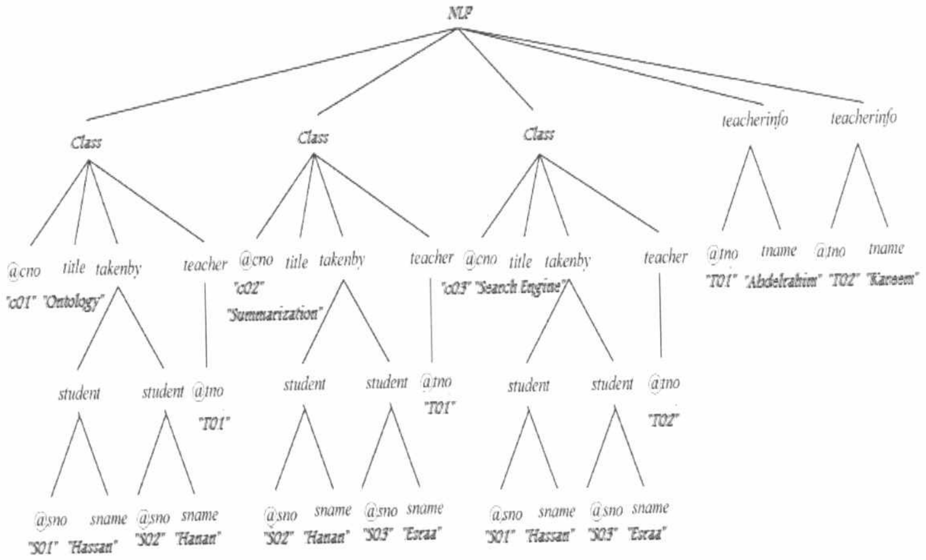


Figure. 4.3: XML document after Removing data redundancies