I.J.Modern Education and Computer Science, **2012, 4, 28-35**
Published Online May 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2012.04.04

Domain Based Ontology and Automated Text Categorization Based on Improved Term Frequency – Inverse Document Frequency

Sukanya Ray
Amity School of Engineering & Technology, Amity University, Noida (U.P.), India
sukanyaray007@gmail.com
Nidhi Chandra
Amity School of Engineering & Technology, Amity University, Noida (U.P.), India
nsrivastava5@amity.edu

***Abstract*—** In recent years there has been a massive growth in textual information in textual information especially in the internet. People now tend to read more e-books than hard copies of the books. While searching for some topic especially some new topic in the internet it will be easier if someone knows the pre-requisites and post- requisites of that topic. It will be easier for someone searching a new topic. Often the topics are found without any proper title and it becomes difficult later on to find which document was for which topic. A text categorization method can provide solution to this problem. In this paper domain based ontology is created so that users can relate to different topics of a domain and an automated text categorization technique is proposed that will categorize the uncategorized documents. The proposed idea is based on Term Frequency – Inverse Document Frequency (tf -idf) method and a dependency graph is also provided in the domain based ontology so that the users can visualize the relations among the terms.

***Index Terms*—**Term Frequency – Inverse Document Frequency, Ontology, Dependency Graph, Text Categorization

I. INTRODUCTION

In the internet various course wares, e-books, documentations, guide books are available but often they are not properly organized or arranged. Often it is found that the search results returned on searching on a domain are not in the sequential order in which the terms or topics of the domain are related with one another. So it becomes difficult especially for a user in a new domain to get proper domain knowledge easily.

In this paper domain based ontology is constructed and users can visualize the dependency graph of the ontology constructed. The dependency graph helps a user to acquire knowledge about the super class, sub classes and also about the object properties of each term. User can have more clear idea by visualizing the relationship between the terms than reading from the hard copy of a book. [1]

The documents found on the topics often do not have any proper title. Later it is found that there are many documents having title say Chapter 1. It becomes difficult for user to identify which document is of which category. The automated text categorization based on the term frequency – inverse document frequency method is used to categorize uncategorized documents. User can categorize the documents using this application and then matching the category with the domain based ontology graph it will be easy for them to study in a proper channel.

This paper is divided into 5 sections. The second section focuses on the background study. The third section gives the proposed methodology to be adopted, the fourth section discusses about the experimental setup and results. The last fifth section will give summary and direction for future research.

II. BACKGROUND STUDY

In this section a brief introduction is given on Ontology, Dependency Graph, Text Categorization and Term Frequency – Inverse Document frequency approach.

A. Ontology:

Ontology is the mechanism that gives idea about the concept, relationships between the terms in each domain. [2] The main aim of domain based ontology is to represent knowledge in such a way such that it can be shared and reused. Domain based ontology apply specific meaning to each term as they apply in that domain. [3]

For example the word ball has got different meanings. Let do ball dance means it is a type of dance from the dance category, while in let us have a game of ball, the term ball means playing equipment. Main applications of ontology are knowledge management, web commerce, e- learning. [4]

The main constraints in construction of domain based ontology are:

i) Sufficient domain knowledge is required.
ii) Construction of domain based ontology is very time consuming.

B. Dependency Graph:

A dependency graph is a directed graph which shows relationships between different objects with each other. "Given a set of object S and a transitive relation R = S x S with (a, b) ∈ R showing a dependency ' a needs b evaluated first' then the dependency graph is a graph G = (S, T) with T ∈ R and R being the transitive closure of T." [5]

Dependency graphs are generally represented in hierarchical order with the root being the most general term and the leaves being specific terms. Using this dependency graph the relationship and dependencies between different concepts can be shown and it also becomes easy for the people to visualize and understand. The concepts are depicted by ellipses and the dependencies by the directed arrows and are shown in Fig. 1 below:

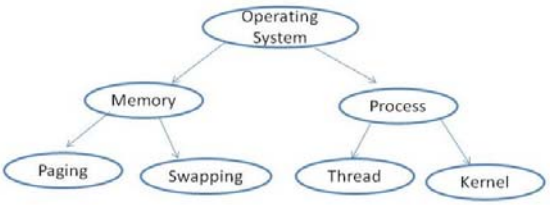

Figure 1: A dependency Graph of the Operating System domain

A dependency graph is almost similar to a concept map except for the fact that in concept map can have any number of relationships between two objects but in a dependency graph two objects can have only one directed relation between them.

C. Text Categorization:

Text categorization means assigning an uncategorized document into one or more pre-defined categories. Text categorization is being widely used for document text retrieval, information retrieval, indexing, classifying and cataloging documents, and filtering web resources. With the huge growth of textual information in digital form text categorization is gaining immense importance. [6]

For example during the organization of a national or international conference thousands of research papers are being submitted for publication and categorization of those research papers manually is very tedious and erroneous. So automated text categorization can provide a solution to this. If we have pre-defined category according to the topics of the conference then we can categorize the digital documents in these categories. This will save time and is most likely to be error free. The same process can be applied during the time of admission where the admission forms submitted can be categorized according to marks obtained or degree needed. In industry also text categorization can be used to filter resumes sent to them. The resumes that do not meet the criteria can be easily ignored using this method without wasting much time.

Text categorization may also provide help to any person new in any domain to gain knowledge about that domain by listing out all the documents available. The person need not have any prior knowledge about that domain for that search, only the domain name will be sufficient for such searching. To sum it up, it can be said that text categorization will help in fast and easy access of proper documents in every domain. [7]

A major drawback of this approach is that the categorization depends a lot on the proper feature selection during the construction of pre- defined data sets. If incorrect features are selected in a data set then the categorization result will be incorrect.

D. Term Frequency – Inverse Document Frequency:

The first step in Term Frequency- Inverse Document Frequency (tf-idf) is the calculation of term frequency. Term frequency is calculated by the total frequency of each individual term appearing in the document. But before calculating the frequency of the individual terms the document need to be processed, that is the stop words need to be removed and them stemming has to be performed on the rest of the words in the document. Stemming moves the words back to their root words and as all the words are of same context then it increases frequency in that context and thus helps in better categorization result. Inverse Document Frequency is calculated by counting the number of documents in which the term appears in the total number of documents in the corpus. The formula of inverse Document Frequency calculation is shown in Equation 1 below:

$$\text{idf}(t) = \log \frac{|D|}{|\{d : t \in d\}|}$$

Equation 1: Equation for calculating the Inverse Document Frequency

where D is the total number of documents in the corpus and $|\{d : t \in d\}|$ is the number of documents where term t appears. If the term is not in the corpora then it will be division by zero which is not mathematically possible. So in that case the term will be adjusted by $1+|\{d : t \in d\}|$. Thus tf-idf is calculated by the Equation 2 shown below:

$$\text{tf-idf}(t,d) = \text{tf}(t,d) * \text{idf}(t)$$

Equation 2: Formula for calculating the Term Frequency-Inverse Document Frequency

where d is number of documents and t in the number of individual terms.

The Tf - Idf approach is derived from the theory of language modeling which states that the terms in a given document can be divided into with and without the property of eliteness, which is the term is about a topic given in the document or not. [7]

But Tf – Idf approach has some major drawbacks like it is sometimes treated as ad – hoc as it has not derived from any mathematical model of relevancy analysis and term distribution. Another drawback is the dimensionality of the text data is the size of the vocabulary across the entire data set present.

III. PROPOSED METHODOLOGY

In this paper a new methodology is proposed such that a user can keep all his/ her documents in an organized manner with the help of automated text categorization and any user who is new in a particular domain can get well acquainted with the domain with the help of domain based ontology.

The architecture for this proposed approach is shown in the figure below:

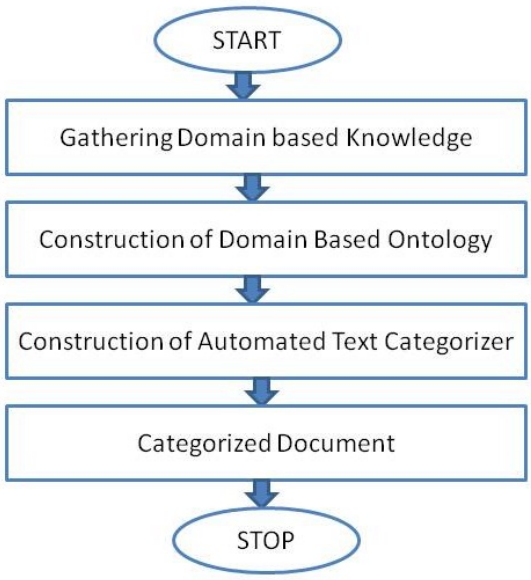

Figure 2: Architecture of Proposed Methodology

The architecture of the proposed Automated Text Categorization method is shown in figure below:

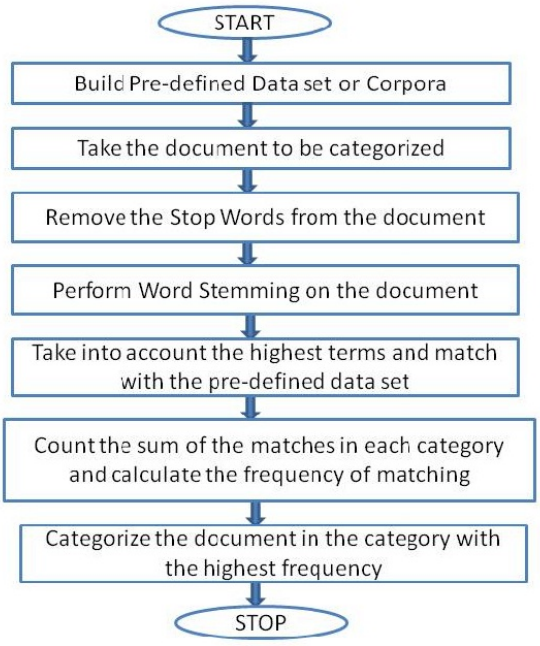

Figure 3: Architecture of the Proposed Automated Text Categorization method.

Each step of the proposed approach need to be discussed in details and also the detailing of using that step is discussed here:

Gather domain based knowledge: Construction of ontology is a very difficult task. For the proper construction of ontology one need to have proper domain knowledge about all the terms and how they are related to one another, their dependency on one another and constraints if any. So gathering domain knowledge is a very important step in the approach.

Construction of domain based ontology: After the gathering of sufficient domain based knowledge it is important to execute it in a proper manner. So in the construction of the ontology based on the domain knowledge the ontology is built. The dependencies and restrictions are put on the different classes and their sub classes. The construction is made in such a way that on clicking on a particular term all the relevant information related to that term like its super-class, sub-class, domain, disjoint classes, usage of domain are shown and it becomes easy for the user to relate all the terms properly.

Construction of automated text categorizer: For automatic text categorization we need to build an automated text categorizer. For the construction of this automated text categorizer the following approach is to be followed as shown in Fig. 3.

Construction of pre-defined data sets: One of the most important and vital step towards the construction of automated text categorizer is the building of pre-

defined data sets. These data sets can be made according to the need of the categorization. For example if the categorization is based on whether students have passed or failed then the data sets will have to contain marks, if the categorization is based on diseases then the data sets will contain the symptoms of various diseases. This data set construction is very time consuming and a lot of care must be taken for the proper construction of these data sets because improper construction of data sets will lead to wrong categorization result. Another thing that has to be kept in mind during the construction of data sets is proper feature selection. Good categorization result depends on proper feature selection. For example one need to categorize documents in computer science field and has data set support of electronics field then correct categorization will never occur.

Submission of document for categorization: For categorization of the uncategorized document the user needs to submit the document in the application interface. The document to be submitted can be in any form but so far support is provided for .doc and .docx form. After the submission of the document the processing of the document starts and the words are tokenized that is all the spaces, punctuations are trimmed and the whole document is treated as a list of words.

Removal of Stop- Words from the document: In English some words appear very commonly but they are not related to any particular domain or category. These are called the Stop-Words, like a, an, the , is, was, are, were, they, their etc. The is the most common word in English and if you count the terms in any document the word 'the' is most likely to have the highest count always. This will lead to wrong categorization result. Moreover even if they do not affect the categorization, they will increase the compilation time of the categorization result as there are almost 400 Stop-Words in English. So it is better to remove all these stop words from the document.

Perform Stemming: Stemming is the process to stem each word back to its root. After removal of the Stop-Words there will remain many words like going, gone, go, goes but all these words come from the root word go and in case of categorization as the highest count matters so it is better to move all the words to their root word as the main context of the document will still remain the same and this will lead to better categorization result. This will also lead to better compilation time. For example let us assume there are words in the document like networks, network, networking, networked. If the words are not stemmed to their root word that is network then the count of each of networks, network, networking, networked will be 1 instead of the count of network as 4. Now having the count of network as 4 it is easier to tell that the document is of network category but before performing stemming as the count of the words network, networks, networked, networking are all 1 and also there will be other less significant words having count as 1, so categorization will be very difficult. So performing stemming of the words is a very necessary step in the automatic text categorization part.

Matching with the pre-defined data-sets: Now with all the processing of the document is done the categorization of the document is to be done. The terms in each of the data set is taken in a list and the word in the document which is taken in a list is matched against each of the data set list.

Count the frequency of the individual term of the document: The number of matches is counted and the frequency of match with each data set is calculated by dividing the sum of matches by the total number of words present in that data set.

Determining the category: The frequency of matching of each data set is counted and the category whose data set has the highest frequency count is said to be the category of the document. The result is shown as the category name to which the document belongs.

The algorithm for each of the above steps of the proposed method discussed is given below:

Construction of Domain based Ontology: Gathering of sufficient domain knowledge is required for the construction of domain based ontology. Based on that knowledge construct the domain based Ontology. The dependencies, sub-classes, super-classes, restrictions, domains and ranges of each class are shown in the Ontology constructed.

Submitting a document: A form is designed having a Browse button. On clicking the Browse button the files present in the system will be displayed. Then select the document of your choice that user want to categorize and click OK button. On clicking 'OK' the document will be submitted and then text will be extracted from the submitted document.

Tokenizing: An array with dynamically increasing size is taken and all inputs to the array are type casted into string. Regular expression is used to determine the pattern according to which the document will be partitioned. All the individual terms are converted into lower case. Now the input strings are split at position defined by regular expression pattern. All occurrences of the regular expression is searched and a set of successful matches found is made. Finally all leading and trailing white space character from the string objects is removed.

Stop Word Removal: A List is being created with the Stop Words. Till the end of the document is not reached each word from the document is taken and

converted to lower case. Each word from the document is matched with the Stop Words in the List. The words that do not match with the Stop Words are put into a new List.

Stemming of the remaining words: At first the different cases which are the possible modification of a root word in English are written. Now each word from the List left after the Stop Word removal is taken up and matched with the cases. If the word is already a root word then it is returned as it is else the steps in the case with which the word matches are followed. The modified word are returned, that is the word stemmed back to its root is returned.

Match the words with the data sets: For each data set a List is created containing the words from each data set, which means there will be as many Lists as there are categories. Now match the words list from the document with each of the data set List and if a match occurs increase the count of that word. Then add the words whose counts are greater than 0 in a List. The remaining words that do not match are ignored.

Categorization: In Categorization part the frequency of matching of each Data Set is calculated. The formula for the frequency of matching is given in equation 3:

Frequency of matching = (Number of matched words from one Data Set) / (Total number of words in each Data Set)

Equation 3: Formula for calculating the Frequency of matching

From the result obtained by using the above formula, the category having the highest frequency of matching is declared to be the category of the document.

IV. SIMULATION RESULT

The domain based ontology is constructed with the help of domain knowledge and then based on the proposed algorithm the automated text categorization was executed on some research compilations from various categories of computer science engineering. The domain based ontology was carried out in Protégé 3.5 Alpha software in OWL language. The visualization of the dependency graph is done with the help of this software and the result of the ontology is displayed in .html format. The simulation of the above proposed idea for automated text categorization is carried out in C# language using Microsoft Visual Studio 2008 software to verify the effectiveness of the proposed idea. Pre-defined data sets on Operating System, Networking, Wireless Sensor Networks, Software Engineering, Software Testing and Digital Image Processing were built and the submitted document was checked against those data sets. The documents to be submitted can be in .txt or .doc or .docx format.

Based on the number of pages and the contents of each research compilation the execution time varies and it is shown in the table below:

TABLE I: EXECUTION TIME DEPENDING ON DIFFERENT TYPES OF DOCUMENT

Serial Number	Size of Document (KB)	Number of Pages	Contains Figures, Tables	Execution Time (sec)
1	48	2	No	1.78
2	60	3	No	2.70
3	84	2	No	2.19
4	124	4	Yes	3.10
5	149	4	Yes	4.15
6	200	8	Yes	13.00
7	254	9	Yes	13.05
8	324	4	Yes	3.96
9	517	5	Yes	3.98
10	600	4	Yes	3.98
11	745	6	Yes	4.00
12	841	5	Yes	3.84
13	941	5	Yes	4.00
14	1028	5	Yes	5.74
15	1151	6	Yes	5.85
16	1243	6	Yes	6.60

The following screenshots displays the result obtained from the proposed methodology:

Figure 4: Screenshot of the dependency graph of domain based ontology.

In Fig. 4 the dependency graph of domain based ontology is shown where the purple arrows shows the sub classes of the selected term and the green arrow shows the super class of the selected term.

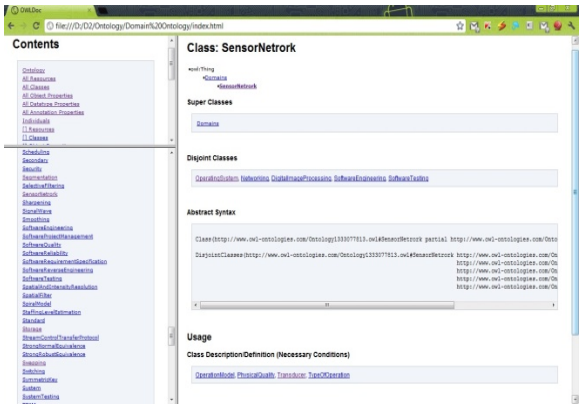

Figure 5: Screenshot of the .html pages generated from the Ontology

In Fig. 5, the .html page shows ontology of term Sensor Network. It displays the super class of the class, the disjoint classes, the sub classes and also the functionality of that term.

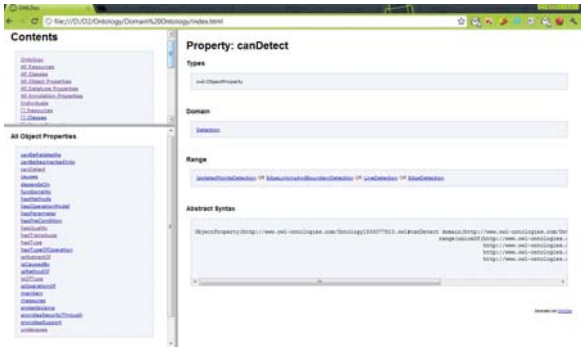

Figure 6: Screenshot of .html page generated from the Ontology

Fig. 6 shows the ontology of an object property. It gives the details of the domain and range of that object property. So it becomes very easy for an user who is new in a domain to gather enough domain knowledge based on this domain based ontology.

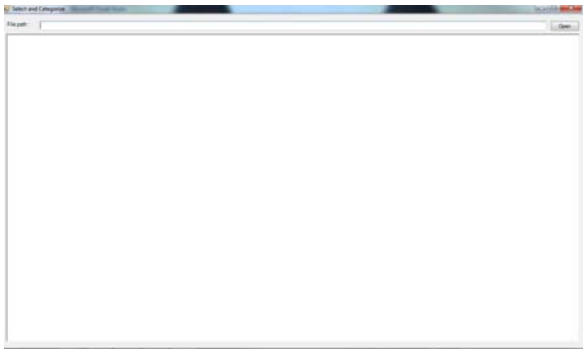

Figure 7: Screenshot of dialog box asking to submit a document for categorization

Figure 8: Screenshot of a document being selected for submission

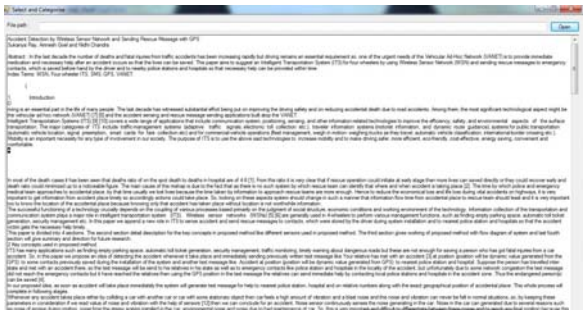

Figure 9: Screenshot of the document submitted is displayed before the categorization result is shown

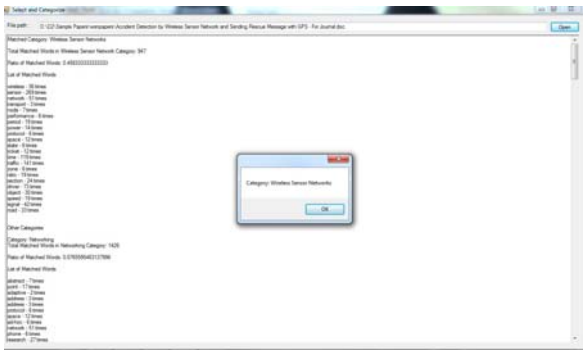

Figure 10: Screenshot displaying the categorization result

Figure 11: Screen shot of the result of categorization

Fig. 10 and Fig. 11 shows the result of the categorization of the uncategorized document submitted in Fig. 8 and is being displayed in Fig. 9.

V. DISCUSSION AND CONCLUSION

The main aim of this dissertation was the construction of domain based ontology and automated text categorization and the proposed method successfully shows the domain based ontology and also categorizes uncategorized documents into proper categories. The compilation time of the result depends on the size and content of the document submitted.

From the above table, Table I showing the execution time of the proposed method, it is clear that the execution time depends on the content of the document, its size and also on the length it has covered in terms of pages. The document having more content in terms of words rather than figures, tables, flow charts are taking more time in completing the execution process and giving the result than document whose size in terms of KB is more as it contains more figures, tables and flow charts.

For instance the document having size 200 KB and 254 KB are 8 pages and 9 pages long respectively and contain one figure each. The time needed to process these two documents and show result is the much higher than the document which is 1243 KB in size and contains many figures, tables, and flow diagrams. The time taken to process the former documents which are almost $1/6^{th}$ the size of the later and display the result is almost twice. It can be said that the execution time of the proposed method is directly proportional to the word content of the document. The more the word content the more time will it take to display the result. So if there are two documents of exactly same size then the one having more word content will take more time to execute.

Based on the results derived after execution of the proposed method it can be concluded that the main aim of this paper is met. Users can categorize any uncategorized documents using this application provided the data set of that category is present at the back end or in the third tier of the model if we consider it as 3-tier architecture with user or viewer at the top tier and proposed method in the middle tier.

The aims of this paper were construction of domain based ontology so that user can relate the terms of a category based on their dependencies, domain and range. This part is successfully done using Protégé software and Web Ontology Language (OWL) and the results are displayed in screenshots. The next aim was the Automated Text Categorization and the execution of the proposed method for that too is giving the desired output which is displayed in the screenshots in the Simulation Result section.

REFERENCES

[1] J. D. Novak and A. J. C. Nas, “The theory underlying concept maps and how to construct and use them,” in Technical Report IHMC C map Tools 2006-01 Rev 01-2008. Florida Institute for Human and Machine Cognition, 2008.

[2] W. M.-j. YUN Hong- yan, XU Jian-liang and X. Jing, “Development of domain ontology for e-learning course,” in ITIME-09 IEEE international symposium, 2009.

[3] T.R.Guber, “Towards principles for the design of ontologies used for knowledge sharing,” in Int..J.Human-Computer Studies. Florida Institute for Human and Machine Cognition,43(5-6), p.p 9.7-928, 1993.

[4] D. Fensel, I. Horrocks, F. van Harmelen, D. L. McGuinness, and P. Patel-Schneider, “Oil: An ontology infrastructure for the semantic web,” IEEE Intelligent Systems, vol. 16, no. 2, 2001.

[5] Wikipedia, “Dependency graph — wikipedia, the free encyclopedia,” 2011, [Online; accessed 16-February-2011]. [Online]Available:http://en.wikipedia.org/w/index.php?title=Dependency_graph&oldid=408804604

[6] Ma Zhanguo, Feng Jing, Chen Liang, Hu Xiangyi, Shi Yanqin, Ma Zhanguo “An Improved Approach to Terms Weighting in Text Classification” 978-1-4244-9283-1/11 2011 IEEE

[7] Sukanya Ray and Nidhi Chandra “A Term Frequency-Inverse Document Frequency Based Prototype Model for Easing Text Categorization Effort for Conference Organizing Committee” International Journal of Computational Intelligence and Information Security, February 2012 Vol. 3, No. 2 pp 33 – 37

Sukanya Ray is currently pursuing M.Tech in Computer Science and Engineering at Amity School of Engineering & Technology, Amity University, Noida. She has received B.Tech degree from IMPS College of Engineering and Technology, WBUT in 2010. Her current research includes in MANET, sensor network and NLP.

Ms. Nidhi Chandra has more than 7 years' experience in academic and Software Development. She is M.Tech from CDAC NOIDA, affiliated from GGSIPU, Delhi. Presently she is working as Assistant Professor at Amity University Noida. She has worked with Tata Unisys and CDAC Noida. Her research interest includes Natural Language Processing, Assistive Technology and Semantic Web Based Application.

