CHAPTER 5

TEXT CLASSIFICATION FOR DIGITAL LIBRARIES

In chapter 2, the problem of distributed text classification in a federated environment is explained. It also discussed various conventional text classification methods. In this present work a multi-agent based approach is used to solve the problem of distributed text classification and the introduction of a new character self-proclamation in the MAS is yet another innovative step.

In this work Section 5.1 gives the outline of the nature of ACM CR pre-specified hierarchy and how the concept matrix is formulated at every specific level. Section 5.2 explains the architecture design of the domain specific multi-agent based document classification system. Section 5.3 explains the mathematical/empirical modeling for this concept-document relativity analysis and its methods. Section 5.4 gives implementation as well as details of the results of this empirical experiment. Section 5.5 presents the summary of this chapter.

5.1 ACM CR PRE-SPECIFIED CLASSIFICATION AND AUTOMATED CONCEPT-MATRIX FORMULATION.

ACM full classification scheme (http://www.acm.org/class/1998/) involves four level concept hierarchies. (Containing three coded levels and a
fourth uncoded level). The tree consists of 11 first-level nodes. The set of children of all first and second-level nodes begins with a node General and ends with a node Miscellaneous. The first-level nodes have letter designations (A through K). The second and third levels have combination of letter-and-numerical designations.

Table 5.1 A Sample Experiment Document

<table>
<thead>
<tr>
<th>Object Oriented Units of Measurement</th>
</tr>
</thead>
</table>

ABSTRACT. Programs that manipulate physical quantities typically represent these quantities as raw numbers corresponding to the quantities’ measurements in particular units (e.g., a length represented as a number of meters). This approach eliminates the possibility of catching errors resulting from adding or comparing quantities expressed in different units (as in the Mars Climate Orbiter error), and does not support the safe comparison and addition of quantities of the same dimension. We show how to formulate dimensions and units as classes in a nominally typed object-oriented language through the use of statically typed metaclasses. Our formulation allows both parametric and inheritance polymorphism with respect to both dimension and unit types. It also allows for integration of encapsulated measurement systems, dynamic conversion factors, declarations of scales (including nonlinear scales) with defined zeros, and nonconstant exponents on dimension types. We also show how to encapsulate most of the “magic machinery” that handles the algebraic nature of dimensions and units in a single metaclass that allows us to treat select static types as generators of a free abelian group. General Terms: Design Languages.
Table 5.2. A Concept Matrix for the Given Sample Experiment Document

<table>
<thead>
<tr>
<th>S.No.</th>
<th>List of Phrases</th>
<th>Freq</th>
<th>S.No</th>
<th>List of Phrases</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Object Oriented Units</td>
<td>1</td>
<td>16</td>
<td>Nominally typed object-oriented Language</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Measurement</td>
<td>1</td>
<td>17</td>
<td>Statically typed metaclasses</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>Programs</td>
<td>1</td>
<td>18</td>
<td>Parametric</td>
<td>1</td>
</tr>
<tr>
<td>4.</td>
<td>Raw numbers corresponding</td>
<td>1</td>
<td>19</td>
<td>Inheritance polymorphism</td>
<td>1</td>
</tr>
<tr>
<td>5.</td>
<td>Quantities’ measurements</td>
<td>1</td>
<td>20</td>
<td>Unit types</td>
<td>1</td>
</tr>
<tr>
<td>6.</td>
<td>Particular units</td>
<td>1</td>
<td>21</td>
<td>Integration</td>
<td>1</td>
</tr>
<tr>
<td>7.</td>
<td>Number</td>
<td>1</td>
<td>22</td>
<td>Encapsulated measurement Systems</td>
<td>1</td>
</tr>
<tr>
<td>8.</td>
<td>Catching errors resulting</td>
<td>1</td>
<td>23</td>
<td>Dynamic conversion factors</td>
<td>1</td>
</tr>
<tr>
<td>9.</td>
<td>Addition</td>
<td>2</td>
<td>24</td>
<td>Dimension types</td>
<td>1</td>
</tr>
<tr>
<td>10.</td>
<td>Comparing quantities expressed</td>
<td>1</td>
<td>25</td>
<td>Encapsulate</td>
<td>1</td>
</tr>
<tr>
<td>11.</td>
<td>Different units</td>
<td>1</td>
<td>26</td>
<td>Algebraic nature</td>
<td>1</td>
</tr>
<tr>
<td>12.</td>
<td>Mars Climate Orbiter error</td>
<td>1</td>
<td>27</td>
<td>Metaclass</td>
<td>1</td>
</tr>
<tr>
<td>13.</td>
<td>Support</td>
<td>1</td>
<td>28</td>
<td>Static types</td>
<td>1</td>
</tr>
<tr>
<td>14.</td>
<td>Units</td>
<td>1</td>
<td>29</td>
<td>Generators</td>
<td>1</td>
</tr>
<tr>
<td>15.</td>
<td>Classes</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In actual classification usage, first-level nodes (like B. Hardware) are never used to classify material. For material at a general level, the General node (in this case B.0) is used. The General node at the first or second level can serve two purposes: it is used for papers that include broad treatments of the topic covered by its parent node (the node immediately preceding it in the
tree), or it may cover several topics related to some (but not necessarily all) of its sibling nodes. For example, under K.7, the Computing Profession, the node K.7.0 General would be used to classify a general article on the computing profession, but also could be used for an article that deals specifically with computing Occupations (K.7.1), Organizations (K.7.2) and Testing, Certification, and Licensing (K.7.3). A set of subject descriptors is associated with most leaves of the tree. The complete classification tree hierarchy is given in http://www.acm.org/class/1998/ccs98.html (1998 Version).

In this present work more number of concepts at every fourth level sub-hierarchy are added. Each and every document is represented as a concept and every concept is represented thorough a concept matrix. Concept matrix contains a list of technical phrases. The system will automatically add any number of concepts at this level. The classification and extraction of technical phrases to construct a concept matrix is the very critical task, for which the list of ACM proper noun index and Keyword index are used. Apart from this a list of words and phrases from Microsoft on-line computer dictionary is also used. The phrase extractor agent automatically extracts using these words and phrases, an additional set of words and phrases. The occurrences of all these phrases are taken and then the relativity between the list of index and newly extracted phrase is taken into account in order to include the particular technical phrase in the concept matrix. These phrases are also stored in the concept dictionary and used for the future usage. For example the independent word and phrase list of D.1.5 Object-oriented Programming consists of a proper noun CORBA (D.1.5) and other related technical words and phrases such as object, class, abstraction, inheritance, polymorphism, dynamic binding, message passing, persistence object and operator overloading are included in the list. Normally while extracting the technical phrases from the list of complete phrases extracted from the document, the occurrence of these
independent words and phrases in those lists and selection of those phrases as the technical phrases are observed. An example of a sample document titled “Object Oriented Units of Measurement” is taken for analysis as shown in Table 5.1. After getting the relevant phrases, the phrase extraction agent formulates the concept matrix with a list of technical phrases along with frequency as shown in Table 5.2.

5.2 ARCHITECTURE OF DOMAIN SPECIFIC MULTI-AGENT BASED DOCUMENT CLASSIFICATION SYSTEM

In this multi-agent system every individual agent is designed to be domain specific (specific-subject) apart from the agent used for the phrase extraction. In this present work it is identified that there exists a multiple server for different departments. If the centralized approach or fully distributed approach is used to solve this problem of distributed classification then it will increase unmanageable traffic or storage space in the network. In order to avoid this situation, the proposed system design in this case takes into account only a single agent for each one of the first-level hierarchies. The domain specific agent is designed to travel to the various servers and identify the document that belongs to the subject-hierarchy. Each server is designed to have a phrase-extraction agent as well as a concept dictionary and this phrase-extraction agent will form a concept matrix after the phrase-extraction. After framing the concept matrix for every new document if a phrase extraction agent finds a new technical phrase, then it will be updated in the centralized dictionary as well as in the Department server. The arrival of a new document, phrase and its category is informed to all servers through a moderator as well as the blackboard. The architecture of this multi-agent system is given in the following Figure 5.1. Every domain-specific agent is designed to travel through the different servers to identify whether the new documents belong to
their own hierarchy or not. Also, every domain specific agent stores a list of phrases related to other domains. At first it performs cosine relativity with the incoming concept-matrix and these specific domain phrase lists. If it predicts relativity with those other agents then it voluntarily proclaims that information to the respective agents. It also transfers the concept matrix to the respective server. A concept relativity table is maintained in all the servers as well as the blackboard and is updated as a concept-dictionary, once every moment. Also, this indicates the movement of the concept-matrix to the different servers. This mechanism reduces the flow of concept-matrices to the different servers. This concept relativity table simply displays category, server number and whether it is related to it or not.

A specific character of a domain specific agent enables the system to learn about the pattern of the relativity among documents. It also maintains the list of phrases for all set of agents. Immediately after extracting the phrases it tries to identify the relative domain and self-proclaims those categories with the respective domain-specific agent. This enables the system to reduce processing load. Thus, instead of directing all the agents to classify the documents, only a related set of agents are given suggestion to classify the documents. Otherwise the deployment of multi-agent system has no meaning in the system design. It simply looks like a fully-distributed system.

5.2.1 Phrase-Extraction Agent

This phrase extraction agent gets the new document coming into the Digital Library and then it performs a list of preprocessing steps. These preprocessing steps involve stop-word elimination, phrase comparison and phrase matching. The phrase comparison through the concept dictionary is provided with the system. It also computes the phrase matching to eliminate
the slight differences. For instance, the ‘information retrieval’ and the ‘retrieval of information’ are two phrases indicating the slight difference. Here, the concept dictionary has recorded the information retrieval. On the other hand, after the arrival of a new phrase ‘retrieval of information’ the phrase extraction agent will locate individual occurrence of the word and compare with the phrases of concept dictionary. If it finds the match with at least with one of the words then the new phrase is identified as the new phrase. This gets recorded in the concept dictionary. After extracting the phrase, the system will frame the concept-matrix. Other than the central server there exists a phrase extraction agent in every individual server.

5.2.2 Domain-specific Agents

In this present design, there are eleven such agents, which are designed to do the concept matching at the individual first level hierarchy. Every subject at the first top level is designed with a specific agent. The dispatcher is informed about the arrival of the new document in the domain-specific server. Immediately the domain specific agents staying in the server actively predict the relativity of that document with other domain specific agents and proclaim the respective agents to perform the concept relativity analysis to identify relativity of that specific concept hierarchy. The concept relativity analysis is performed through the concept relativity algorithm. This involves a mathematical process of identification of semantic relativity through Latent Semantic Analysis and is explained in Section 5.3. After identifying the specific concept, the individual agents take care of recording the concept with the specification of the hierarchy of its own level. It also places the document in the concerned hierarchy.
UIA – User Interface Agent  PEA – Phrase Extraction Agent
LDB – Local Data Base  LCD – Local Concept Dictionary
MDSSPA – Mobile Domain Specific Self-Proclamative Agent

Figure 5.1 Architecture of Domain Specific Multi-Agent Based Document Classification System
5.2.3 Concept Dictionary

A concept dictionary provides a list of concepts/phrases required to run the system. At the initial stages while starting up the system a list of independent concepts are taken from ACM computing review classification index, keywords as well as using the words and phrases of Microsoft on-line computer dictionary. These phrases and words are entered by the user through the user interface agent to provide systems ontology. Later additions of concepts will automatically take-place after the identification of new technical phrases. The concept dictionary is designed through a database. The interaction between the concept-dictionary and the user takes place through the user-interface agent. In this present work the concept dictionary is developed using the MS-Access database. A concept dictionary user interface is shown in the above Figure 5.2. Normally, the concept-dictionary can be accessed in two ways. One is through the concept dictionary user interface by the user. The other one is through the phrase-extraction agent. If the phrase extraction agent gets the new concept-phrases then it will automatically add the same to the concept dictionary.
5.2.4 User-Interface Agent

The user interface agent facilitates the user to enter the intended document into the system for the classification through the input screen. Also, an additional component can be invoked to enter the concept-phrase through the concept dictionary. The user can invoke this user–interface to add, find, update and delete the concept phrase as well as indexes in the concept dictionary. A concept Dictionary user interface is shown in the following Figure 5.2.

5.2.5 Black Board

This is a shared memory, which is able to store and exchange the concept-matrix as well as the messages required for different servers. The individual domain-specific agents are permitted to read the content of this shared memory. The phrase-extraction agent is permitted to write the phrases, concept-relativity table entries and new document-hierarchies in this shared memory.

5.3 MATHEMATICAL MODELING FOR CONCEPT-PATTERN RELATIVITY ANALYSIS

Latent Semantic Analysis (LSA) is a theory and is a method for representing the contextual usage of meaning of phrases and its relatedness by statistical computations applied to a large corpus of text. The phrase and passage meaning representation derived by LSA has been found to be capable of simulating a variety of human cognitive phenomena. After processing a large sample of machine-readable language text, LSA represents the phrases,
either taken from the original corpus or new, as points in a very high dimensional semantic space.

In this case it is represented as a conceptual matrix and it also permits one to infer about the relation of expected contextual usage of phrases. LSA applies a Singular Value Decomposition (SVD) to the matrix; this is a form of a factor or more properly the mathematical generalization of which factor analysis is a special case. In SVD, a rectangular matrix is decomposed into the product of three other matrices. One component matrix describes the original column entries in the same way, and the third is a diagonal matrix containing scaling values such that when the three components are matrix-multiplied, the original matrix is reconstructed. The reconstructed two-dimensional matrix that approximates the original matrix and a few highest values are selected to reconstruct the original matrix.

Each document in the particular sub-hierarchy represents the rows and each phrase with respect to the document is represented as the columns. Learning human like knowledge consists in formulating a bivariate frequency table with row i representing the i\textsuperscript{th} phrase and column j representing the j\textsuperscript{th} document (or between any two entities) and $f_{ij}$ evaluated by the Shannon’s measure of information $\sum p \log p$. This together with the dimension reduction will constitute the constraint satisfaction for prediction between the observed and the expected values to make classification. Actual data pertaining to any two measurable entities (phrases and sentences, text classification in digital libraries, etc.) will have to be collected. Sets of examples pertaining to each of the two entities can be exhibited in a bivariate frequency table for determining the relationships between any two examples. Tables can be formulated for comparison and valid conclusions.
SVD is a powerful technique employed for solving a linear system of equations $AX=B$, in M equations of N unknowns with $M \geq N$ in order to get unique set of solutions; a set of singular solutions, infinite number of solutions; non trivial solutions or trivial solutions based upon the nature of the coefficient matrix A. Whatever maybe the vectors X, B concepts of rank, null space, range space of linear algebra are essential in formulating the computer program for any practical problem in conformity with the decomposition of the matrix $A$

$$[A]=[U][W][V^T] \quad (5.1)$$

When more equations than the unknowns are given, relevant solutions can be obtained by least squares method.

After the reconstruction of the original matrix we find the correlation between the new document and the existing document in the sub-hierarchy. If the correlation is high then it is decided that the new document belongs to the particular sub-hierarchy category. The same process is repeated in all agents, if none of them finds correlation in its sub-hierarchy then it will be put under the miscellaneous category.

The main issue while using the LSA is the size of the matrix and if it is very high than and the systems are sometimes not able to process this. In order to avoid this situation and keep the matrix size under control every time only a set of five to ten documents alone are taken out of entire set of the given sequence for relativity analysis.
5.3.1 An Example is Worked-out to Exhibit the Relating to the Situation

The hierarchy classification that is taking place in the domain-specific agent is as follows. First the domain specific agent extracts all the phrases and formulates the concept-matrix as given in Section 5.1. Five such human-judged document samples (training documents) are taken and put in the form of a bi-variate frequency table. Later the sixth document is added. All the five documents are judged under D.1.5 category by the expert human, but the sixth document category is not yet known. Then the same matrix is applied to Singular Value Decomposition and this results in the product of three matrices. From these matrices the original matrix is reconstructed with a few highest diagonal values. The correlation between these matrixes reflects the relativity of the document with every other five documents. In the present experiment the agent is taking care of Software (D) and is able to identify the relativity of the document under the category of D.1.5.

D. SOFTWARE
D.1 PROGRAMMING TECHNIQUES
D.1.5 OBJECT-ORIENTED PROGRAMMING

After decomposition, the category of D.1.5 finds a good correlation among the training set as well as the new document coming into the system. The average correlation is above 0.50 and so it is decided that the new document belongs to that category. The result of the correlation is given in the following Table 5.3.
Table 5.3 Correlation between Training Samples and new Unclassified Document

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>0.239</td>
<td>1</td>
</tr>
<tr>
<td>D2</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>0.239</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-0.239</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>0.239</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>0.239</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>D6</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0.239</td>
<td></td>
</tr>
</tbody>
</table>

5.4 SIMULATION EXPERIMENT

The system under consideration is simulated using apache web servers. In this present experiment all these servers are hosted with IBM Tehiti server to run the Aglets (Danny Lange 1998) The Aglet is a Java based autonomous agent. The Aglets Workbench, developed at IBM's research labs in Japan, is aimed at producing stand-alone mobile agents. The subject-specific agent, phrase-extraction agent and user interfaces agent are developed using these aglets.

There are four such Apache web servers that are hosted and one of them acts as a central server that runs the moderator, black-board and central concept dictionary. An Agent Transfer Protocol (ATP) is used to communicate with these different agents. Aglet uses a technique called serialization to transmit data on the heap and to migrate the interpretable byte-code. Aglet has a well-defined entry point for itself to re-start computation. Aglet also supports persistence by calling the appropriate base-class functions. Temporarily stored aglets can be stored in secondary storage and later they can be activated. This facility easily permits one to simulate the blackboard system using aglets.
These aglets support message passing and broadcasting. Each aglet is integrated with the functional components of our architecture. The blackboard system is shown as the explicit component and is implemented through using the standard Java serialization. For the domain specific aglets (Agents) initially the user has to specify the training sample document either from the local machine or from the web through an user-interface aglet. Each domain specific aglet is designed to learn the concept-matrix of that specific hierarchy. The phrase extraction aglet is developed to extract the phrases and frame the concept matrix and then the same is placed in the blackboard. After that, the same concept matrix is taken by the domain-specific aglets. Training documents are indicated with the specific category-hierarchy; otherwise it is treated as an experiment document. After the training is over, the new sample document is given to the system that is preprocessed and the concept-matrix is passed to blackboard. Then the moderator broadcasts the message to all the domain-specific aglets about the arrival of new document. Then every domain-specific aglet gets the same and processes it to proclaim the category hierarchy.

5.5 EXPERIMENT RESULTS AND DISCUSSION

In these experiments, two types of experiments are performed to prove the efficiency by means of different metric measurements. In these experiments, the ACM CR classification system at the fourth-level hierarchy consists of nearly 1120 categories and the present system is trained with five documents under each category. Later every sample document which is classified under a certain category is put under the specific category and is taken for further document relativity analysis. If it is not able to decide the category with the given training sample documents then the process is repeated until it finds the relation with the any of the existing set of
documents. If it is not able to find the correlation between the lists of all documents then it is decided that the specific document does not belong to this category.

5.5.1 Data Sets

In these experiments, during the first phase, two data sets were collected from different Internet portals. In the first data set collection a set of 100 papers are collected from ACM Portals by submitting eight queries. For each query at most 10 to 15 documents are taken for experimentation. In the second data set collection, a set of 450 computational research literatures were collected randomly from Google Web Directory by submitting nearly fifty queries on various topics of information technology.

In the second phase the author explicitly adjudicated the ACM CR categories of these research papers. It is taken as the expected standard value for the collection. The categories given in the research literature are taken as the human expert judgment. The Table 5.4 gives the category statistics of the data sets. For each data set, the resultant search documents are divided into 11 subsets, based on the human expert judgment results of primary classification categories. These two data sets are taken for experimentation.

**Table 5.4 Statistics of the Two Data Sets**

<table>
<thead>
<tr>
<th>S.No.</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set 1</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>15</td>
<td>10</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>4</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Data Set 2</td>
<td>39</td>
<td>66</td>
<td>88</td>
<td>43</td>
<td>32</td>
<td>36</td>
<td>40</td>
<td>22</td>
<td>12</td>
<td>16</td>
<td>39</td>
</tr>
</tbody>
</table>
5.5.2 Performance Measure

To evaluate this approach the data sets collected for experimentation are given to the human expert judgment and then the same is compared with machine classified results using phrase based as well as word based approaches. In order to evaluate the effectiveness of classification system two set of metrics are measured. The first set of well known metrics (Precision, Recall, F1 Measure and Chi-Square) (George Forman 2003) are used and are shown by

\[
\text{Precision} = \frac{\text{Number of Documents Classified Under Particular Category}}{\text{Total Number of Documents Taken for Classification}} \quad (5.2)
\]

\[
\text{Recall} = \frac{\text{Number of Documents Classified Under Particular Category Using Word/Phrase Based Method}}{\text{Total Number of Documents Under a Particular Category Under Human Expert Judgment}} \quad (5.3)
\]

\[
\text{F1-Measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (5.4)
\]

\[
\text{Chi – Square} = \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}} \quad (5.5)
\]

F1-Measure is the harmonic mean of the precision and recall. This metric measures the ultimate measure of performance of the classifier.

Chi-Square is the common statistical test that measures divergence from the observed and expected, if one assumes the future occurrence is actually independent of the class value. As a statistical test, it is known to behave erratically for very small expected values, which are common in text classification both because of having rarely occurring word/phrase features, and sometimes because of having few positive training examples for a concept.
The Second set of metrics is the distributed multi-agent system performance measures. The Cougaar Architecture (Aaron Helsinger 2003) measures the communication and storage strengths by various measures. Similarly in this present design the storage and communication cost are measured using the following methods.

\[ SC = OSC + DSC + ASC \]  \hspace{1cm} (5.6)

\[ CC = DTC + MTC \]  \hspace{1cm} (5.7)


### 5.5.3 Experiment 1 Classification Results Metrics:

Based on the definitions of precision and recall, the graph is drawn with the total number of documents classified using different phrases and word based approaches as shown in the Figures 5.3 and 5.4. In all cases of classification performance metric measure F1 represents the harmonic measure of precession and recall. In this experiment, the following Figure 5.5 shows the F1 measure comparison between phrase and word based approach in the machine classification. From the F1 metric measurements, it is clearly observed that the phrase based approach is always better than the word based approach. The Chi-square distribution is applied for the data pertaining to the human classified documents Vs machine document classification using phrase and word based approaches. This is shown in Figure 5.6.
Figure 5.3 Experiment 1: Precision Graph for Classification System

Figure 5.4 Experiment 1: Recall Graph for Classification System
5.5.4 Experiment 2 Classification Results Metrics

In the second experiment another set of 450 documents are given to the human expert judgment and then the same is compared with phrase and word based machine-classified results. The result of this experiment is shown in the following Figures 5.7, 5.8, 5.9, 5.10. Based on the definitions of precision and
recall, the graph is drawn with the total number of documents classified using different phrase based, word based and human-expert judgment approaches as shown in the Figures 5.7 and 5.8. In the second experiment, the following Figure 5.9 shows the F1 measure of the machine classification. Large samples normal distribution is applied instead of the chi-square distribution and this is shown in the Figure 5.10.

Figure 5.7 Experiment 2 : Precision Graph for Classification System

Figure 5.8 Experiment 2 : Recall Graph for Classification System
5.5.5 Experiment 2 Distributed Results Metrics

In this present multi-agent architecture, in order to evaluate the efficiency of the multi-agent approach with the character of self-proclamation, two additional parameters such as communication cost as well as the storage
cost are measured through the central, distributed and the multi-agent approaches. It is assumed that all servers are installed with required agents. The storage and communication cost is measured in terms of KBs (Kilobytes) of information. Each unit of communication message constitutes one KB of information. The result of this analysis is shown in the following Figures 5.11, 5.12, 5.13, 5.14, 5.15, 5.16.

**Figure 5.11 Central Server Approach Storage Space Analysis**

![Central Server Approach Storage Space Analysis](image1)

**Figure 5.12 Central Server Approach Communication Cost Analysis**

![Central Server Approach Communication Cost Analysis](image2)
Figure 5.13 Distributed Server Approach Storage Space Analysis

Figure 5.14 Distributed Server Communication Cost Analysis
From these metric measurements it is evident that compared to the central, distributed approach the multi-agent approach is very much efficient in terms of storage as well as the communication.
5.5.6 Discussion of Results

From the resultant classification metrics of both experiments such as Precision, Recall, F1-Measure and Chi-Square proves that such classification yields good results. While comparing the word and phrase based approach, the phrase based approach is better in terms of four metrics. The Chi-square value in the application of the chi-square test is found in which the data pertaining to the human classified documents against the machine document classification are assumed to be independent. The Chi-Square test indicates that there is a significant difference between human-judged document classification and machine-classified approaches using phrase as well as word based classification at 5% level. In the second experiment instead of using a Chi-Square test, the normal distribution is applied. A phrase based approach compared to word based approach in the distributed document classification environments makes a better classification.

5.6 CHAPTER SUMMARY

An attempt is made in this work to design and develop a multi-agent framework for document classification using ACM CR classification. This work has revealed that the user of Latent Semantic Analysis with Multi-Agent Systems enables us to classify the documents under a specific hierarchy in the distributed computing environment. The initial system is trained with only a few documents. Over a period of time, after acquiring the new set of documents the system will reveal more accurate results. This is measured using different metrics. In this work self-proclamation is identified as the new character in the multi-agent system and is established by such experiments. This character of self-proclamation enables the system to classify the documents automatically in a distributed environment. In this experiment the
system is tested with ACM CR categories. The same may be extended for other categories like DDC and other types of classification system (Science/Engineering) to build the complete subject-specific servers across the different departments.