Object Oriented Intelligent Multi-Agent System Data Cleaning Architecture to clean Preference based Text Data

Dr. G. Arumugam
Department of Computer Science
Madurai Kamaraj University
Madurai 625021

T. Joshva Devadas
Department of Master of Computer Applications
The American College
Madurai 625002

ABSTRACT

Agents are software programs that perform tasks on behalf of others and they are used to clean the text data with their characteristics. Agents are task oriented with the ability to learn by themselves and they react to the situation. Learning characteristics of an agent is done by verifying its previous experience from its knowledgebase. An agent concept is a complementary approach to the Object Oriented paradigm with respect to the design and implementation of the autonomous entities driven by beliefs, goals and plans. Preference based text data cleaning is based on the selection issue. Preferences are given by the user in the form of alphabets, numbers and special characters. Preference based Text data cleaning process transforms the given text data into structured database and extracts the required information using the given keyword. Agents incorporated in the architectural design of a Text data cleaning process combines the features of Multi-Agent System (MAS) Framework, MAS with Learning (MAS-L) Framework. MAS framework reduces the development time and the complexity of implementing the software agents. MAS-L framework incorporates the intelligence and learning properties of agents present in the system. MAS-L Framework makes use of the Decision Tree learning and an evaluation function to decide the next best decision that applies to the machine learning technique. This paper proposes the design for Multi-Agent based Data Cleaning Architecture that incorporates the structural design of agents into object model. The design of an architectural model for an Intelligent Multi-Agent based Data Cleaning inherits the features of the Multi-Agent System (MAS) and uses the MAS-L framework to design the intelligence and learning characteristics.

Keywords

Text data, Preference, Agents, MAS, MAS-L, Architecture, Intelligent, Data Cleaning, IMASDC

1. INTRODUCTION

Text data mining is categorized into document retrieval, processing, cleaning, mining and visualization. In the text data mining process, significant cleaning of extracted free text is typically required in order to accurately portray the prevalence of concepts in the corpus. Cleaning removes the irrelevant material as much as possible and combines words that represent the same concept [5][36]. Issues in Text cleaning are related to the selection and compression of the terms. Selection determines the candidate keyword for analysis and compression is grouping together synonymous terms. Stemming is the most common type of compressions that uses rule based algorithm for combining words. Focus of this paper is to clean the text data using the preference given by the user in the form of alphabets, numbers and special characters. Preference based text data cleaning is based on the selection issue and it transforms the given text data into a structured data. Intelligent Multi-Agent System Data Cleaning (IMASDC) Architecture is used to perform the text data cleaning by extracting information from the give text data file.

Agents present in this architecture are capable of processing independently and react with the system using their knowledgebase [15]. Agents introduced in the cleaning process clean the text data by using the preferences specified by the user and information is extracted by applying transformation principle [25]. These agents make use of the object oriented frameworks MAS and MAS-L to utilize the communication and intelligence components present in it [6]. The proposed architecture along with these two frameworks makes the data cleaning system to have intelligent behaviour. This IMASDC architecture is used to clean the arbitrary text data with the preference given by the user.

In this paper Section 2 describes about the object oriented frameworks MAS and MAS-L. Section 3 describes the Preference based Text Data cleaning. Section 4 describes Multi-Agent based Text data cleaning and preference based text data cleaning using the Multi-Agent Architecture. Section 5 explains the agent learning for preference based text data cleaning and presents learning algorithms based on decision trees. Also this section illustrates the significance of Decision tree construction process with sample training example. Section 6 describes preference based Text data cleaning using IMASDC Architecture and the implementation process with a sample input.. Section 7 analyses normal preference based Text data cleaning method and the method based on IMASDC Architecture.

2. OBJECT ORIENTED FRAMEWORKS

2.1MAS Framework

Objective of MAS framework is to reduce the development time and the complexity of implementing software agents [31]. The design of this framework is based on Gaia methodology, which is used to build models in the analysis and the design phase [34]. This framework is composed of one abstract class Agent, two final classes namely ProcessMessageThread and AgentCommunicationLayer and four interfaces [10]. The

interfaces are AgentMessage, AgentBlackBoard information, InteractionProtocol and AgentInterface [11].

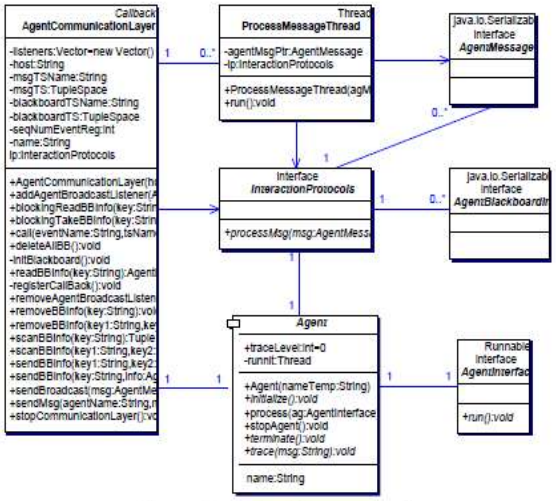
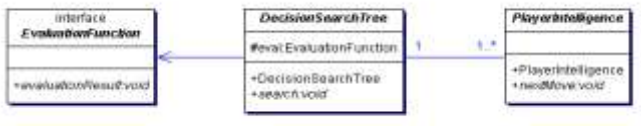


Figure 1 – MAS Framework

Agent class is responsible for providing start up code, ending code, and display of messages [9]. AgentInterface, a sub class inherited from Agent, implements the run method and is responsible for initiating agent's private actions. InteractionProtocol interface defines the interaction among the agents. ProcessMessageThread creates a new control for every incoming message and AgentMessage interface takes care of the message format. AgentBlackboard information specifies the black board message format. The class AgentCommunicationLayer implements the entire communication needed for agents to interact with the distributed system.

2.2 MAS-L Framework

The MAS-L Framework is combined with the MAS framework to implement the learning and intelligence characteristics of software agents [23]. This Framework incorporates two machine learning approaches namely Decision Tree based learning and an Evaluation function. The decision tree approach of software agent learning is based on the current state of environment. The nodes of the tree are intermediate decision points and the leaf nodes are the final states where the searching ends successfully. The search algorithm indicates the best move that leads to successful final state. The Evaluation function is used to determine the next possible move to perform action.



The MAS-L Framework has two abstract classes namely, DecisionSearchTree (DST) and PlayerIntelligence and an interface Evaluation Function [24]. The first class to be extended is Evaluation function that evaluates the intermediate node of the decision tree. This class emphasises well defined data structure along with the properties of the Decision trees to choose a machine learning technique to implement the evaluation function. The next class to be extended is the DecisionSearchTree (DST) that implements the data structure of Decision Tree along with the searching. To reduce the searching time in the DST, pruning techniques are incorporated using the maximum depth search algorithm. This class shall instantiate the class that implements evaluation function to perform the evaluation of intermediate nodes. Finally the PlayerIntelligence class acts as an interface to the software agents. This class shall instantiate the class that extends DecisionSearchTree to decide the next best decision.

3. PROBLEM DEFINITION

Text data cleaning is performed using the preference given by the user [7]. Text data cleaning is made by verifying the data present in the text with the preference given by the user. If the preference matches then the data is extracted and sent as an output of the cleaning process. The description of the Preference based text data cleaning is described as follows.

3.1Preference based Text Data Cleaning

Text file comprises alphabets, numbers, special characters, hyper texts, hyper links etc. Data cleaning is done based on the portion of the data in which the user is interested [14]. Objective of this preference based text data cleaning relies upon the input received from the user. Based on the input received from the user the cleaning is made on the text data.

In the Text Data cleaning process information extraction is done by applying segmentation followed by transformation. Segmentation separates the text data into parts of different categories and those parts are transformed into a structured data [17]. The structured data is then extracted using the user specified keyword. This process is repeated for each of the data and those data are matched with the user preference for information extraction. Major drawback of this Text data cleaning process is that, whenever the source text data file is referred the same process is repeated irrespective of the user.

3.2 Improvements in the Preference based Text Data Cleaning

Preference based Text data cleaning described above does not make any attempt to reduce the time required for the cleaning task. Use of functional dependencies and transformation concepts in the preference based data cleaning process improves the performance of text data cleaning. Preference based Text data cleaning is done by first verifying the existence of the source file. Then the cleaning system checks the preference by matching the user specified preference. The next step is to transform the text data into a record structured form with two attributes namely Data type and Content[21]. Data types are the classifications identified / present in the source data. Once the data is transformed into a text database, data cleaning is done by grouping the user requested preferences into various clusters and using of functional dependency (FD) concepts [16]. In the preference based text data cleaning process, Functional

dependency plays a vital role in removing unwanted information from the text data. The transformed data is stored in a database and is used in the cleaning process.

3.3 Functional Dependencies and Text data cleaning

Once the text data is transformed into a text database (structured database) data cleaning process applies the functional dependency principle to clean the text data. The data transformation also eliminates the inconsistencies by transforming the source data into a form that is more suitable for the current application. This transformation process identifies and defines appropriate functional dependency association with the text data and uses it in the data cleaning process to filter the unwanted data using the key attributes [18]. Use of the FD in the cleaning process improves the performance of the system and the data quality.

3.4 Phases of Preference based Text data cleaning

Transformation, Preference Matching and Cleaning are the three major phases involved in the preference based text data cleaning. Transformation phase receives and transforms the source data into structured data. Each and every character present in the text data is transformed into the text database. Though the transformation phase takes the same amount of time required for cleaning the text data, it improves the performance of data cleaning by using the transformed data in the subsequent cleaning process. The next phase is to find out the data that match the user specified preferences by verifying the transformed structured data. Only those records that match with the preference are extracted as the output. This extracted information forms a cluster and is termed as cleaned data.

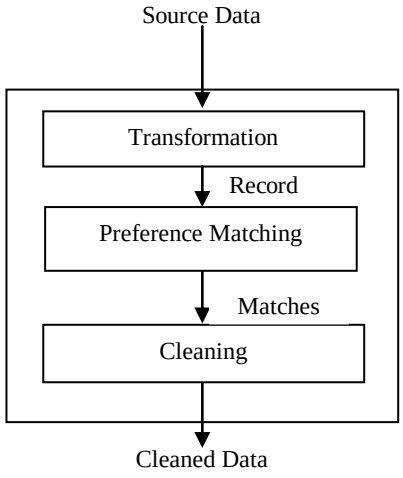


Fig 3 Phases of Preference based data cleaning

3.5 Need of Agent based Text Data Cleaning

Use of Functional Dependencies, Information Extraction and Data transformation improves the performance of the existing system. But the agent based text data cleaning reduces not only the processing time but also improves the performance of the system by using its intelligent behavior in cleaning the text data. The Agent uses its reactive characteristics to react with the environment with its possessed knowledge [15]. Agent's intelligence is stored in the form of actions in the knowledgebase and this knowledgebase is used to react with the system [3]. Agent makes use of the transformed text data (stored in the form of records) to perform the data cleaning task. Before performing the cleaning task the agent makes use of its knowledgebase to verify the existence of cleaned details to avoid repeating the tasks that have been done already by the agent [27]. Such verification process eliminates the repetition process in the data cleaning. The use of knowledgebase and the verification process improves the performance of the system and this leads to have an intelligent agent in the data cleaning task [20].

4 MULTI-AGENT DATA CLEANING ARCHITECTURE

Agents are made available with some initial knowledge in order to take initial decisions [2]. Data cleaning architecture describes the interaction and method of communication between the multi-agents present in the system [8][13]. The Data Cleaning Multi-Agent Architecture requires Interface Agent, Data Collection Agent, Data Cleaning Agent, Knowledge Management Agent and Message Handling Agent to perform different tasks. These agents are described in the subsequent paragraphs.

Interface Agent (IA) present in the top most layer interacts with Data Collection Agent, Data Cleaning Agent, Message Handling Agent and Knowledge Management Agent. This agent submits the user provided details to the Data Collection Agent. Also, IA receives user information from the Data Cleaning Agent, interaction dialogues from the Message Handling Agent and mined results from the Data Cleaning Agent. Finally, it returns the result from the interacted agents to the user.

Data collection agent (DCOA) present in the second layer gets user details from Interface Agent (IA) and sends it to the Knowledge Management Agent (KMA) to verify the user information. The result of the verification process from the KMA is passed to the Interface Agent. Also, DCOA has the responsibility to handover all the collected information, namely the user name, file name, file type, user preference or the filter key to the Data Cleaning Agent to perform the data cleaning task.

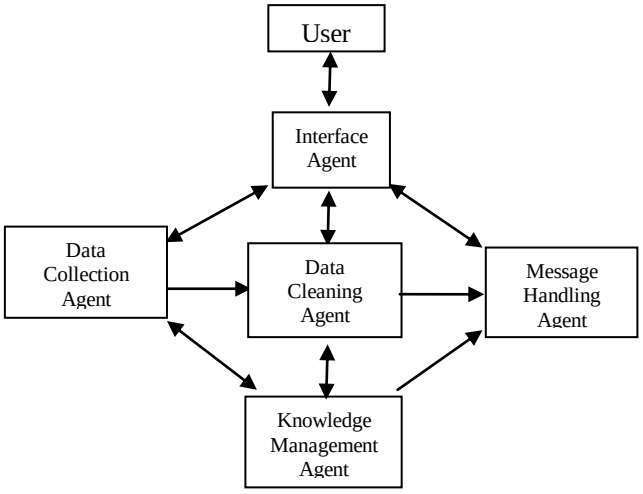


Fig 4. Multi-Agent Architecture for Data Cleaning

Data cleaning agent (DCLA) in the second layer receives the user profile and data cleaning details from the Data collection agent (DCOA). The knowledge required for the data cleaning process is received from the KMA. If the KMA instructs to carryout the cleaning process then the data cleaning is performed by executing appropriate methods with the keyword. If the KMA finds that the cleaning has already been done then the result is handed over to the IA by the DCLA. DCLA uses the keyword and the file name given by the user for cleaning either by clustering or by filtering technique. Irrelevant or the *inconsistent data* are identified by analyzing the data items which are not close to the context and are identified as a group of clusters. These clusters are either omitted or deleted from the database. The remaining data in the database after this process is sent to the IA.

Knowledge Management Agent (KMA) does the process of identification and elimination of user information. KMA possesses some initial knowledge and functions to analyze the knowledgebase. This Agent is capable of learning from the previous experience present in its knowledgebase. The KMA updates with the addition of new knowledge or by the removal of conflicting patterns that may present in the knowledgebase. At this stage either the knowledge is added to the knowledgebase or removed from it. The functionality of the KMA is to receive user details from Data Collection Agent and newly learned knowledge from Data Cleaning Agent. Also, it requests the Message Handling Agent (MHA) to prepare erroneous message or interaction dialogues to handle the situation. With the user profile and preferences it verifies its knowledgebase to decide for initiating the Data Cleaning Agent to perform the cleaning task. If the cleaning process has already been done with the specified preference in a particular file, the KMA decides not to do the cleaning process. Otherwise the cleaning process is initiated in the DCLA by the KMA. The MAS-L framework is designated for KMA-Learning. KMA initiates the Intelligence class of MAS-L Framework by instantiating the Decision Tree that invokes the evaluation function to decide the next best move.

Message Handling Agent (MHA), handles messages among the agents present in the system. KMA initiates the MHA to prepare a message when the requested file is not available in the specified path or directory and/or when the specified preference does not match with the data present in the system. During the cleaning task the Data Cleaning Agent may encounter an erroneous situation or may require the user to provide some additional information. To handle this situation Data Cleaning Agent requests the MHA to prepare either an erroneous message or user interaction dialogue. MHA sends appropriate message to the Interface Agent.

4.1 Preference Text Data Cleaning using Data Cleaning Multi-Agent Architecture

Objective of Text Data Cleaning focuses towards the Preference based data cleaning. Preference based data cleaning process and the phases of data cleaning are described in detail in the previous section. Agents are intelligent processing components that can perform action on behalf of others [35]. Agent chooses the method Pref() to perform Preference based data cleaning by analyzing the current cleaning task [32]. Agent learns from the data cleaning environment and updates its knowledgebase [1]. The learning behavior is defined in the MAS-L framework and agent uses this behavior when necessary. Introduction of an intelligent component in the data cleaning process enables the learning activity in the environment and the reaction to the situation causes improvement in the performance of the system.

5. AGENT LEARNING IN PREFERENCE BASED TEXT DATA CLEANING

5.1 Agent learning process

Machine learning is the area of artificial intelligence that examines how to write programs that can learn. Machine learning is often used for prediction or classification [30]. Prediction is done based on the feed back and an agent learns through examples. When a similar prediction occurs in future the feed back is used to make the same prediction or completely a new prediction. Machine learning applied in the data mining task uses a model to represent the data. The model is a graphical structure such as neural networks or decision trees.

A decision tree is a tree where the root and each internal node are labeled with a question. The arcs emanating from each node represents each possible answer to the associated question. Each leaf node represents a prediction of a solution to the problem under consideration [19]. A Decision Tree is a prediction modeling technique used in classification, clustering, and prediction tasks. Decision Tree uses *divide and conquer* technique to split the problem space into subsets. Searching in a decision tree starts at the root node and ends successfully at the leaf node. Leaf nodes represent the successful guess of the object being predicted.

Decision tree based machine learning approach is introduced in the IMASDC architecture describes the examples in terms of attribute-value pairs that range over finitely many fixed possibilities. Also the concept to be learnt may have discrete and disjunctive description may be required to arrive at a decision. Learning decision trees classifies the training example with a finite set of attributes along with its associated values [19]. Attributes and values are used to learn the structure of the decision trees which can be used to decide the category in an unknown situation.

Learning takes place as a result of the interaction between the agent and the environment, and from observation by the agent of its own decision making processes [12]. Learning is used not only to react but also to improve the agent's ability to act in the future [22]. Agents incorporated in the Text data cleaning process works under unsupervised learning principle and they use machine learning based decision trees as a tool to learn from the environment with its past experience [33].

5.2 Decision Tree based learning algorithm for Preference based Text data cleaning

The Decision Tree based learning focuses on Preference based Text data cleaning. Preference based text data cleaning approach is described in this section.

Algorithm:

```
Preference-Clean ()
Begin
If File-Avail='yes' Then
    // check the existence of the source file taken for the
        // cleaning task
Read-data ();  // read the data from the training example

If Preference = 'Matches' Then
    //user provided preferences are matched with the data
     //cleaning system  preference
       Perform-pref-store();
If DB-Avail = 'Avail' Then
        //  Check the existence of the database file
        // Verify the source file with *.DB extension
    Load-file ();
    Use-pref-store ();
     //  use the preferences stored and checks the source file
      // name along with the first character of the user given
  // preference. [ if 'sample' is the source file then it checks
 // 'sample-a' for alphabet preference 'sample-n' number and
  // 'sample-s' for special characters]

If Cleaning = 'Reqd' Then
        // appropriate extension of the source file is not present
   Clean();//Clean given file with the user specified preference
   Store (); //  Store the cleaned file in knowledgebase along
                //with the first character of the user given preference
Else  {Cleaning not required }
                    // No need to perform cleaning task
  Load-clean();//Cleaned data is present in the knowledgebase
            //  Extract the cleaned data from the knowledgebase
End if
Else  {Database is not available }
   Clean(); // Clean the given file with user specified reference
   Db-Construct ();  // Construct the Database and store
   Store();          // Store cleaned file in the knowledge base
End if
   Else {Preference did not match}
     Send-error ();
 // Agent prepares an error report and sends it to the message
 // handling agent to prepare error message as 'Preference does
// not match'
   End if
Else {File is not available}
     Send-error ();
  // Agent prepares an error report and sends it to the message
 //  handling agent to prepare error message as ' File not found'
End if
```

End.

The decision tree based learning for the preference based text data cleaning learns the cleaning task by transforming the given text data in to a structured database and extracts information from it using the given input string.

Learning task associated with the Preference based text data cleaning makes use of the machine learning approach 'decision tree' to enable the learning behavior of an agent. Decision tree approach yields either of the binary decision 'yes/no'. For the preference based text data cleaning, the learning agent (KMA) makes use of the decision tree (Fig.5) to perform the learning action. Learning agent is trained using the training examples present in Table 1 to perform preference based text data cleaning [28]. 'File-Avail', 'Pref', 'Database', and 'Clean' are the node attributes associated with the preference based text data cleaning. Data cleaning algorithm present in this section enables the agent to learn from the environment and reacts to the situation by executing appropriate cleaning methods. The algorithm is tested using the training example.

5.3 Decision Tree construction process for Preference based Text Data Cleaning

Decision tree construction process makes use of the ID3 (Induction Decision Tree Version 3) machine learning inductive algorithm. This algorithm requires computing the measures *entropy* and *information gain.* The training data used in the decision tree construction process is built by using the set of node attributes along with its possible values [26]. ID3 algorithm uses this training data as input and constructs the decision tree by placing the attribute node in its place by using the measures entropy and information gain. The entropy and information gain are computed using the following formulae.

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i$$

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Evaluation of the intermediate nodes is done by an Evaluation function that applies the Deterministic Competitive Learning algorithm (DCL) concept. The algorithm found to be more suitable for Boolean valued function and is used in the evaluation function. The DCL algorithm changes its state and moves to one more level down in the decision tree if the Evaluation Function returns true value otherwise it remains in the same state. In other words, the algorithm learns only if the status of the output has been changed compared to the previous iteration. DCL updates the weights using the following procedure [4].

$$W_j[n+1] = W_j[n] + S_j(y_j)(x - W_j[n])$$

Where W is the weight vector, x is the given input and

$$S_j(y_j) = \begin{cases} 1, & \text{if } j \text{ is the winner} \\ 0, & \text{otherwise.} \end{cases}$$

The Decision Tree construction process for preference based Text Data Cleaning is described below in detail.

The Decision Tree construction process for the preference based data cleaning uses the Table 1 as training example to train the Knowledge Management Agent and makes use of the intelligence component of IMASDC architecture to determine the attribute that can be placed at the root of the decision tree. Attributes involved in the decision tree construction process is in the Table 1. Placing a node at the root of the decision tree is done by computing the measures Entropy and Information gain using the attributes identified in Table1. Ten data items are taken as a sample to train the agents of which four samples returns *yes* as its learning state and other six returns with *no*.

Table 1. Sample training data for Preference Based Text Data Cleaning

Sl No	FILE-AVAIL	PREF	DB-AVAIL	CLEAN	Learn
S1	YES	MAT	AVAIL	REQ	Y
S2	YES	MAT	AVAIL	NREQ	N
S3	YES	MAT	NAV	REQ	Y
S4	YES	MAT	NAV	NREQ	Y
S5	YES	UM	AVAIL	REQ	N
S6	YES	UM	AVAIL	NREQ	N
S7	NO	MAT	AVAIL	REQ	N
S8	NO	UM	NAV	NREQ	N
S9	NO	MAT	NAV	REQ	N
S10	NO	UM	AVAIL	NREQ	N

Knowledge Management Agent makes use of the training data (Table 1) to learn the preference based text data cleaning process. FILE-AVAIL, PREF, DATABASE and CLEAN are the node attributes identified to train the Knowledge Management Agent. KMA initiates the learning process for the Preference based data cleaning from the root node (FILE-AVAIL) of the decision tree. KMA reads the value of the training example and determines the action. If the value of the root node attribute FILE-AVAIL is 'avail' then KMA initiates the cleaning action by moving one level down in the decision tree and reaches the node PREF. The KMA determines not to perform any action / operation for the value 'not avail' of the root node FILE-AVAIL.

After reaching the node PREF, KMA reads the value of the node PREF from the training example. If the value of the node PREF 'matches' then KMA moves from the current node and reaches the node DATABASE at the next level. No action is taken by the KMA, if the value of the node PREF is 'unmatched'. The node DATABASE has two values 'available' and 'not available'. If the value read by the KMA is 'available', KMA makes use of the database file to perform the cleaning task otherwise it creates the database file. At this stage KMA reaches the node CLEAN in the decision tree by reading the value 'matches' of the node PREF. If the value of the node attribute CLEAN is 'required', KMA performs the data cleaning using the user specified preference. Otherwise (not required) KMA makes use of the existing cleaned details and sends the same as output.

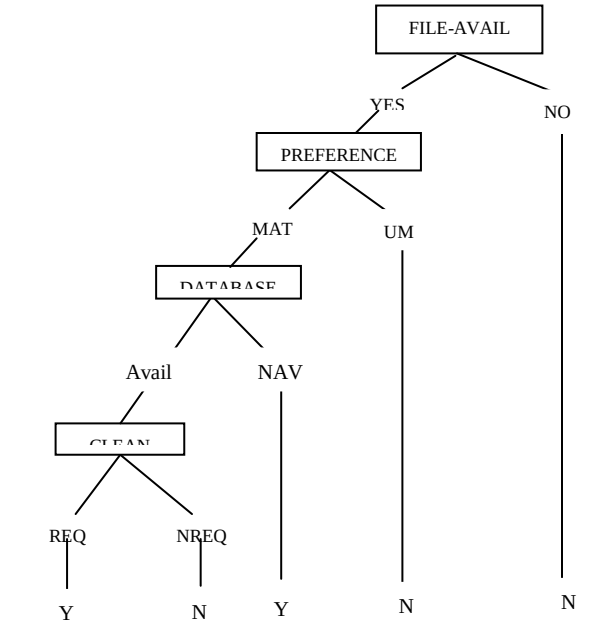


Fig 5 Decision Tree Learning Approach for Preference Based Data Cleaning

6. INTELLIGENT MULTI-AGENT SYSTEM ARCHITECTURE FOR DATA CLEANING

Intelligent Multi-Agent System Data Cleaning (IMASDC) Architecture combines the features of Data Cleaning Multi-Agent Architecture, MAS and MAS-L frameworks to carry out data cleaning process (Fig 6). The MAS and MAS-L frameworks are designed using the object oriented concept and IMASDC Architecture makes use of them to improve the performance of the cleaning task. Agents present in the IMASDC Architecture inherit all the classes and interfaces from the Agent Class of the MAS framework, and the learning aspect is initiated by using the intelligent class of the MAS-L Framework. These agents use the *Interaction Protocol* to interact with the other agents and make use of the *Agent Communication Layer* to implement the communication protocol to interact with the data cleaning system. All communications are done through the *Interaction Protocol* that uses *AgentBlackBoardinformation* to exchange the

message. Black board is used to establish communication between the agents available in the data cleaning architecture.

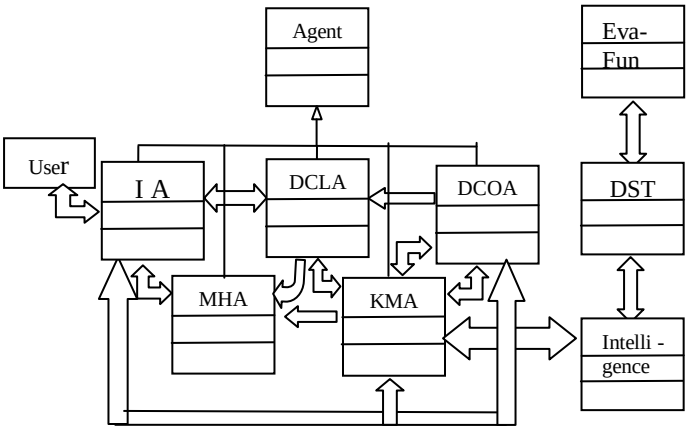


Fig 6. IMASDC Architecture (with Multi-Agent Data Cleaning Architecture, MAS and MAS-L Frameworks)

Agent's communication is visualized through Black Board and is used as a common platform for all the agents to communicate [29]. Interface Agent uses the *Agent Interface* to initiate other agents. MHA uses *Processmessagethread* and *Agent Message* of the MAS framework to prepare and handle the message. KMA initiates the intelligence class of MAS-L framework to enable the learning activity. The intelligence Class in turn calls the Decision Search Tree class which constructs the nodes for the Decision Tree using the ID3 machine learning algorithm.

6.1 Preference based Text Data cleaning using IMASDC Architecture

User initiates the data cleaning task by providing the source file name and the details necessary to perform the cleaning task. Multi-agent present in the data cleaning architecture receives the user provided information and determines the action with the help of the KMA. Preference based Text data cleaning process uses IMASDC Architecture to transform the given data into a structured database and to extracts the required information. KMA present in the architecture uses specific key value to initiate the intelligence component to construct appropriate decision trees. The preference based text data cleaning approach considers the attributes FILE AVAIL, PREFERENCE, DATABASE and CLEAN for the decision tree construction. Table 1 is used as a training sample to train the KMA and to plan for further action plan. KMA initiates the search process from the root of the decision tree and makes use of the evaluation function to determine the action. In the decision tree search process, agent learns to move from the current node to one more level down in the decision tree or remains in the same node based on the value returned by the evaluation function. Preference based Text data cleaning using the IMASDC Architecture is described in detail as follows.

Text Data cleaning process receives the input from the user and initiates data cleaning action through Interaction Agent (IA). IA sends the details received from the user to the Data Collection Agent (DCOA) and DCOA sends the same to the Knowledge Management Agent (KMA) for further processing. KMA verifies its knowledgebase and checks the existence of cleaned details of the source file. If search is successful then KMA sends the cleaned details to the Data Cleaning Agent (DCLA) to perform the cleaning task. If the search is unsuccessful KMA makes use of the keyword to select the data cleaning method to perform the action. If the keyword provided by the user is Preference, KMA selects the method Preference-Clean() to clean the text data. Once the selection of the cleaning method is over KMA provides the source file and the method needed for cleaning to the DCLA to initiates the cleaning action. DCLA starts cleaning the text data by applying user specified preference to extract the data from to the source file. After cleaning the source data DCLA sends the cleaned data to the KMA and IA. Cleaned details are updated in the KMA's knowledgebase as a part of its learning activity. IA sends the resultant output file received from DCLA to the user. MHA is capable of preparing interaction dialogues or error messages or interaction message to other agents. KMA and DCLA initiate the MHA to prepare message when they encounter an erroneous situation or need more information from user or to provide some useful information to the user or agents.

6.2. Implementation

Our experiment considers the text data for analysis and focuses on Preference Extraction based Text cleaning. User specified preference is given to the system to extract the data that match the given keyword string. Subsequent paragraph describes the implementation aspect of text data cleaning with IMASDC Architecture.

IMASDC prompts the user to enter his name and password to enter into the system. The system classifies the user as an existing user or a new user. If the user is new, the system requests the user to choose/enter the name of the file to be cleaned along with the key attribute. If an existing user logs into the system, the KMA prepares the user's default file and the key attribute from its knowledgebase by analyzing the user's previous entries and returns the most preferred key. This key is used to set the user's current preference to clean the data. Now the Message Handling Agent interacts with the user by sending message through Interface Agent as "Want to continue the cleaning task with this preference?". If the user's response is "yes", Data Cleaning Agent (DCLA) continues with the existing preference otherwise it permits the user to alter his preference.

Data given by the user or the default preference is sent to the Interface Agent. Interface agent forwards the details to the DCOA and it in turn sends the same detail to the KMA. KMA sends the source file to be cleaned along with the keyword to the Intelligence Component for further processing. Intelligence component (IC) checks the keyword and chooses appropriate decision tree (preference based decision tree). After selecting the decision tree IC sends the information to the concerned decision tree and initiates the search process from its root node. Decision Search Tree component forwards the source file to the evaluation function to determine the cleaning action. Evaluation function makes use of the keyword to check the availability of the file with *.PRE.DB extension for the keyword 'preference'.

*.PRE.DB stands for Preference based Data Cleaning database file. If the file with this extension is present, the evaluation function returns 0 (without changing state) and the decision tree chooses the 'other' option and determines not to perform any cleaning operation. Decision tree forwards the cleaned file name (with .PRE.DB extension) to the intelligence component. Intelligence component forwards the file to the KMA. KMA sends the cleaned file to the DCLA for further processing. KMA initiates the MHA to prepare a message about the file existence and MHA in turn sends the 'file existence' message to the user via IA.

Once the KMA identifies the cleaning method, it starts learning from the execution environment. To clean the text data the system receives the file name and the keyword as input from the user via Interaction Agent which sends the details to the DCOA and it in turn sends the same to the KMA. If the keyword is "preference", then KMA initiates the Intelligence component to use PREF-BASED decision tree to enable the agent learning behavior. The decision tree search process starts from the root node AVAIL to check the availability of the given source file. Intelligence component sends this detail to the KMA to check the availability of the file. If the file is present KMA sends the reply as 'yes' to the intelligence component and the decision tree uses the value 'yes' to choose the branch of the tree. The value matches with the left child of the root node. Now evaluation function determines the action plan by changing its state and moves to one more level down in the decision tree to reach the node PREF. The Intelligence component receives the user preference needed for data cleaning from the KMA and compares it with the preferences defined in the system. If the preference matches then the evaluation function returns 1 and moves the decision tree from the current node to one more level down by changing its state and reaches the node DB-AVAIL.

Intelligence Component sends the current state to the KMA and checks the availability of the Database file (file with *.DB Extension). If the database file associated with the source file is 'found' then the KMA chooses the file with *.DB extension to perform the cleaning task and sends it to the Intelligence Component. Now the evaluation function returns 1 and changes its state that enables the decision tree to move one more level down to reach the node CLEAN. Now the intelligence component sends the current node detail to the KMA.

KMA verifies its knowledgebase and determines whether cleaning is required or not. If cleaning is 'required' (new preference but existing source file) then the Intelligence component initiates the decision tree to continue the search process and it in turn initiates the evaluation function to determine the action. The evaluation function returns 1 and moves one more level down in the decision tree. The search process of the decision tree encounters a leaf node. Now Intelligence component sends the current state to the KMA to perform the action. KMA sends the cleaning details to the DCLA to perform cleaning task. DCLA cleans text data and returns the cleaned file to the KMA which stores the file with *.PREF.DB in its knowledgebase. KMA initiates the MHA to prepare a message for the completion of data cleaning process. DCLA sends the cleaned details to Interaction agent. Interaction agent sends the cleaned details to the user. MHA prepares the file cleaned message and sends it to the Interaction Agent and IA in turn forwards the message to the user.

If the cleaning is 'not required' (cleaned file details are available) then the intelligence component initiates the evaluation function to determine the action. The evaluation function returns 0 and remains in the same state. The decision tree search process reaches the leaf node. The Intelligence Component sends the current state to the KMA to perform action. KMA extracts the cleaned file from its knowledgebase and sends it to the DCLA to perform the cleaning. KMA also prompts the MHA about the file existence. DCLA sends the cleaned details to the Interaction Agent and it in turn sends the resultant cleaned file to the user. MHA sends a message to the Interaction Agent about file existence and IA in turn sends the same to the user.

If the database file associated with the source file (with *.DB extension) is 'not found' then the KMA sends the source file to the DCLA to clean the data by transforming the text data into a structure database. The transformation process is done for all the characters present in the text file. DCLA uses this occasion by simultaneously cleaning the text data using the preference and transforming the text data into a structured form. After cleaning / transformation process is over DCLA sends the cleaned data along with the transformed file to KMA and it stores the cleaned file with *.PREF.DB extension in its knowledgebase. KMA initiates the MHA to prepare message into the completion of data cleaning process. MHA prepares a message and sends it to the Interaction Agent and it in turn sends to the user.

If the user specified preference 'does not match' with the system provided preference then the Intelligence component sends the error signal to the KMA. After receiving the error signal from intelligence component KMA initiates the MHA to prepare an error message as 'Preference does not match with the system'. MHA prepares and sends the message to the interaction agent and it in turn sends it to the user.

If the user specified source file is not found due to wrong representation of path or the filename or the file extension the Intelligence component sends the error signal to the KMA.

After receiving the error signal from intelligence component KMA initiates the MHA to prepare an error message as 'file not found'. MHA prepares and sends the message to the interaction agent and it in turn sends it to the user.

6.3 Sample Text data and Cleaned Text data Sample Source data

The need for data cleaning is centered around improving the quality of data to make them "fit for use" by users through reducing errors in the data and improving their documentation and presentation (see associated document on Principles of Data Quality – Chapman 2005a). Errors in data are common and are to be expected. Redman suggested that unless extraordinary efforts have been taken, that a field error rate of 1-5% should be expected.

Transformed data (Database)

Type	Content
A	The Need for data cleaning is centered around improving the quality of
A	data to make them
S	“
A	fit for use
S	“
A	By users through reducing errors in the data and improving their
A	Documentation and presentation
S	(
A	See associated document on Principles of Data Quality
S	-
A	Chapman
N	2005
A	a
S	)
S	.
A	Errors in data are common and are to be expected. Redman suggested
A	that unless extraordinary efforts have been taken, that a field error rate
A	of
N	1
S	-
N	5
S	%
A	should be expected

Fig 7. Transformed data

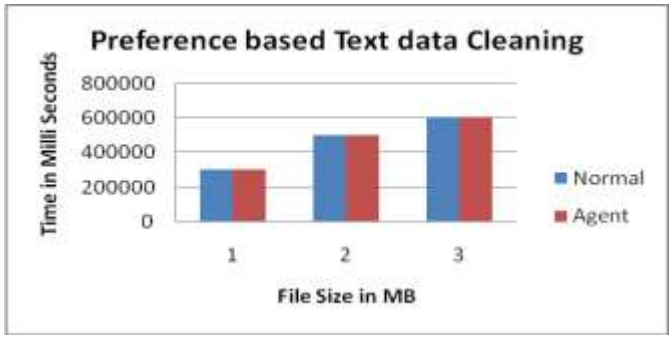
Cleaned output for the user specified preference

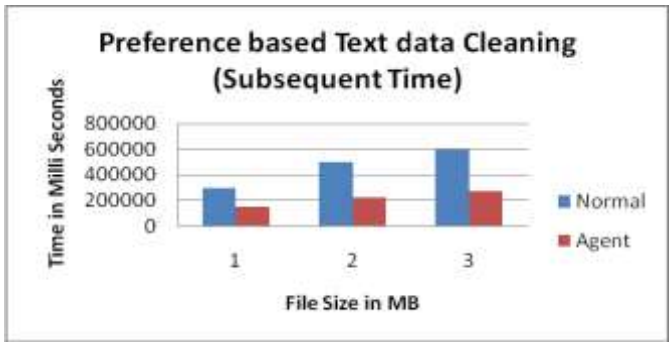
(User preference Alphabet –A)

The need for data cleaning is centered around improving the quality of data to make them fit for use by users through reducing errors in the data and improving their documentation and presentation see associated document on Principles of Data Quality Chapman a. Errors in data are common and are to be expected. Redman suggested that unless extraordinary efforts have been taken, that a field error rate of should be expected.

6.4 Analysis

Data cleaning associated with the text data is performed with Preference based cleaning method. Analysis of the text data cleaning of the method is done with agent and non agent based cleaning. Text data cleaning based on Preference based data cleaning process transforms the given text data into a text database and extracts the information from it. Agent based system makes use of such transformed database file and reduces the time required to perform data cleaning.





The above charts are obtained by analyzing the text data cleaning in terms of normal cleaning and agent based cleaning. Agent based text data cleaning is found to be much better than the normal cleaning procedure. It is observed that for the first time the agent based text cleaning and normal cleaning procedure requires same amount of time. In the subsequent process the agent based system uses its previous experience (stored in its knowledgebase) for cleaning with reduced time required to perform the data cleaning. Use of Multi-Agents in the data cleaning architecture reduces the computation time and the performance of the data cleaning system is improved by the learning behavior of the agent. It is observed that IMASDC Architecture based text data cleaning gives better performance over non agent based (normal) text data cleaning.

7. CONCLUSION

This paper proposes Intelligent Multi-Agent Data Cleaning (IMASDC) Architecture that uses the Object oriented frameworks MAS and MAS-L. Functionalities and the behaviour of the agents present in this architecture are described in detail. Among the agents present in the architecture KMA is

designated to be the learning agent. To implement the learning characteristics of KMA a learning algorithm based on the machine learning tool Decision Trees was used. Text data cleaned using IMASDC Architecture brings the data cleaning system to have intelligence behaviour and improves the functionality of the system with higher performance than ordinary systems. Future work on this paper may use the IMASDC architecture to clean the text database, clean biological database and Email data.

REFERENCE

[1] Ayse Yasemin SEYDIM, "Intelligent Agents : A Data Mining Perspective", CiteSeer-IST Scientific Literature Digital Library, 1999.

[2] Alex Bordetsky, "Agent-Based Support for Colloborate Data Mining in System Management", Proceedings of the 34th Hawaii international conference on System Science, 2001,vol ©IEEE, ISBN 0-7695-0981-9/01

[3] Dae Su Kim, Chang Suk Kim, Kee Wook Rim, "Modeling and Design of Intelligent Agent System", International Journal of Control Automation and Systems, 2003, Vol.1 No.2.

[4] Dong-Chul Park,"Centroid Neural Network for Unsupervised Competitive Learning", IEEE Transactions on Neural Networks, 2000, Vol 11 © IEEE, ISBN S1045-9227(00)02998-2.

[5] Dinah Payne, Cherie Courseault Trumbach, "Identifying synonymous concepts in preparation for technology mining", Journal of Information Science, 2007, DOI: 10.1177/0165551506076401

[6] Fayad. M, D.Schmidt, "Building Application Frameworks: Object-Oriented Foundations of Design",John Wiley & Sons,1999.

[7] Feldman R , Fresji M, Hirsh H, Aumann Y, Liphstat O, Schlter Y , Rajman M, "Knowledge Management : A Text Mining Approach" ,Proceedings of the 2nd International conference on Practical Aspects of Knowledge Management(PAKM98),1998.

[8] Ferber, J, "Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence", Addison-Wesley Pub Co, 1999.

[9] Garcia, A., Silva, V., Lucena, C., Milidiú, R. "An Aspect-Based Approach for Developing Multi-Agent Object-Oriented Systems", Simpósio Brasileiro de Engenharia de Software Rio de Janeiro, 2001.

[10] Garcia, A., Lucena, C. J., Cowan, D.D., "Engineering Multi-Agent Object-Oriented Software with Aspect-Oriented Programming", Elsevier, 2001.

[11] Garcia, A., Lucena, C. J.,"An Aspect-Based Object-Oriented Model for Multi-Agent Systems", 2nd Advanced Separation of Concerns Workshop at ICSE-2001, 2001.

[12] Garro, A., Palopoli, L.,"An XML Multi-Agent System for e-Learning and Skill Management", Third International Symposium on Multi-Agent Systems-Large Complex Systems and E-Businesses (MALCEB-2002), 2002.

[13] Gerhard Weiss, "Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence", The MIT Press, 1999.

[14] Helena Galhardas, Daniela Florescu, Dennis Shasha, Eric Simon, Christian-Augustin Saita, "Declarative Data Cleaning: Lanaguage, Model and Algorithms", Proceedings of the 27th VLDB conference, 2005.

[15] Ioerger, T. R. He, L. Lord, D. Tsang, P , "Modeling Capabilities and Workload in Intelligent Agents for Simulating Teamwork", Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society (CogSci-02),2002, PP482-487.

[16] Jiawei Han , Micheline Kamber, "Data mining: Concepts and Techiniques" , Morgan Kaufmann Publishers- Elsevier, 2001.

[17] Jie Tang, Hang Li, Yunbo Cao, Zhaohui Tang , "Email Data Cleaning", Proceedings of the KDD'05 , 2005, vol © ACM, ISBN 1-59593-135-X/05/0008.

[18] Kollayut Kaewbuadee, Yaowadee Temtanapat, Ratchata Peachavanish, "Data Cleaning using FD from Data Mining process", Proceedings of conference, 2003.

[19] Margaret H Dunham, Sridhar S "Data Mining – Introductory and Advanced Topics" Pearson Education, Inc., Copyright ©2003, ISBN 81-7758-785-4

[20] Massimo Cossentino, Antonio Chella and Umberto Lo Faso, "Designing agent based systems with UML" ,international conference on agents, 2006.

[21] Raymond J Mooney , Razvan Bunescu, "Mining knowledge from text using information extraction", SIGKDD Explorations, 2003, vol 7 issue 1, pp 3-10.

[22] Russell, S., Norvig, P., "Artificial Intelligence, A Modern Approach", Prentice-Hall, 1995.

[23] Sardinha, J.A.R.P., Ribeiro, P.C., Lucena, C.J.P., Milidiú, R.L., "An Object-Oriented Framework for Building Software Agents", Journal of Object Technology,2003, Vol 2No.1.

[24] Sardinha, J.A.R.P., Milidiú, R.L., Lucena, C.J.P., Paranhos P , "An Object-Oriented Framework for Building Intelligence and learning properties in Software Agents",Journal of object Technology, 2004.

[25] Shahram Rahimi and Norman F. Carver , "A Multi-Agent Architecture for Distributed Domain-Specific Information Integration", Proceedings of the 38th Hawaii International Conference on System Sciences,2005, vol ©IEEE, ISBN 0-7695-2268-8/05.

[26] Soman K P, Shyam Diwakar, Ajay. V, "Insight into Data Mining Theory and Practice", PHI, 2008, ISBN 978-81-203-2897-6.

[27] Stader, J., Macintosh, A., "Capability Modeling and Knowledge Management- In Applications and Innovations in Expert Systems VII", 19th Int Conf on Knowledge-Based

Systems and AAI Springer-Verlag, 1999, pp 33-50 ISBN 1-85233-230-1.

[28] Symeonidis A L, Chatzidimitriou K C, Athanasiadis I N, Mitkas P A , "Data Mining for agent reasoning : A synergy for training agents", Engineering Applications of Artificial Intelligence- Elsevier, 2007.

[29] Tobin J Lehman, Stephen W. McLaughry, Peter Wyckoff, "T Spaces : The next Wave", Proceedings of international conference on Machine Learning, 1999.

[30] Tom M Mitchell, "Machine Learning", McGrawHill,1997, ISBN 0070428077.

[31] Weiss, G., "Multiagent systems: a modern approach to distributed artificial intelligence", The MIT Press, 2000.

[32] William E Winkler, "Data Cleaning Methods", Conference SIGKDD'03 , 2003, vol © ACM, ISBN 1-58113-000.

[33] Winston, PH. "Artificial Intelligence", Addison Wesley, 1992.

[34] Wooldridge, M, Jennings, N. R., Kinny, D."The Gaia Methodology for Agent-Oriented Analysis and Design",Kluwer Academic Publishers, 2000.

[35] Zili Zhang, Chengqi Zhang and Shichao Zhang, "An Agent-based hybrid framework for database mining", Taylor & Francis Group Applied Artificial Intelligence, 2003,pp 17:383-398.

[36] Zhang Jin , "Research on Data Cleaning in Data Acquisition", conference on data mining, 2001.