International Journal of Decision Support System Technology
Volume 13 • Issue 3 • July-September 2021

A Text Mining Approach Agent-Based DSS for IT Infrastructure Maintenance

Sidhamed Elandaloussi, LIO, Ahmed Ben Bella Oran 1 University, Algeria

https://orcid.org/0000-0002-4342-3332

Pascale Zarate, IRIT, Toulouse University, France

https://orcid.org/0000-0002-5188-1616

Noria Taghezout, LIO, Ahmed Ben Bella Oran 1 University, Algeria

ABSTRACT

Information technology (IT) infrastructure refers to the combined set of network, software, hardware, applications, and all the information technology-related equipment for an enterprise IT environment. In addition, it provides the entire skeleton for an organization to continue delivering several services to its internal members (employees) and external ones (customers/partners). The interruption of services leads to a significant deterioration of the infrastructure, and at the same time, it slows down their functioning. Additionally, it can result in important loss of user trust. Therefore, we need to proactively help the technician teams to assess the quality and availability of their IT infrastructure. This study may help to build an approach for an IT infrastructure called MAITD-2 in order to classify, analyze, and take problems to closure in a short time face to a multi-criteria decision-making problem.

KEYWORDS

Aggregation Function, Corrective Maintenance, Decision Support, IT Infrastructure, Multi-Agent System, Multi-Criteria, Similarity Measure, Terms Extractions, Text Mining

1. INTRODUCTION

IT infrastructure maintenance includes all the tasks and necessary actions for their proper functioning. There are two main categories of IT maintenance: hardware and software one which are divided into several areas: application, virtualization, storage, server, network, security, systems…. Additionally, the problem can be started and treated from the simplest to the most complex. Moreover, it will take time to find the best solution depending on the user's competence. These services can be provided on site or remotely in which they are conducted as corrective maintenance (Kent et al., 2017) or through preventive methods with fixed time (Mehmeti et al., 2018).Indeed, according to this context within IT infrastructure maintenance, several proprietary software has need appeared like: Nagios (Nagios IT management and monitoring product, 2007), Manage Engine (Manage Engine IT management and monitoring product, 2007) and Zenoss (Zenoss IT Infrastructure management and monitoring tool

DOI: 10.4018/IJDSST.2021070105

for hybrid IT environment, 2018). However, all these IT infrastructure solutions are preventive and limited when the problem is already occurred. Recently, the purpose of the maintenance task is to calculate the maintenance needs before the equipment fails, i.e. continuous monitoring (Callewaerta et al., 2017) with the aim to improve controls, processes and to prevent or detect fraud in an IT platform. Additionally, decision support systems have been developed to assist decision-makers in solving problems preventively and correctively thanks to supervised(Zenoss IT Infrastructure management and monitoring tool for hybrid IT environment, 2018) or unsupervised methods(Campos et al., 2007; Liao et al., 2013). Our study is part of corrective decision support systems (DSS) with semi-supervised approach to reduce the time of problem resolution. Unfortunately, the quality of the selected solution often depends on problem description quality and user knowledge level. In some situations, the problem description could be interpreted in different ways. In another situation, the pertinent terms do not appear in the situated problem. Nevertheless, these facts lead us unsatisfactory solutions. Therefore, we developed a text mining-based approach to explore, analyze all unstructured inputted data. Our aim is to provide automated support to the user and deliver new information to the system.

The rest of this paper is organized as follows: We first briefly review the different strategies for maintenance with some related works. Section three is devoted to the description of the proposed approaches by introducing our general architecture. In section four, we present the technics of terms extraction. Section five resumes the different strategies for text similarity measure with some corresponding related works. Section six describes our main idea about implemented extraction methods. Section seven experiments, evaluate, and discuss the obtained results during troubleshooting task. Finally, section eight summarizes the paper with some final remarks by pointing out possible directions for future works.

2. RELATED WORK

The multi-agents systems have offered remarkable results in different disciplines and application including maintenance fields. This one becomes an important research issue. Therefore, the work described in (Haack et al., 2011) uses a hierarchical Multi-Agent System (MAS) for monitoring and reporting policy violations within the security environment. (Kendrickm et al., 2018) proposed a decentralized multi-agent security system (DMASS) as a scalable solution for the collection and analysis of cyber-security and network forensic data. Authors in (Jahanbin et al., 2013) introduced an agent framework for forensic information gathering by using three types of agents for data collection, analysis, and alert generation. In addition, the works in (Bukhsh, 2019) propose to leverage the tree-based classification techniques of machine learning in order to predict maintenance need. (Daniel et al., 2018) Present a novel Multi-Agent System based Cloud Monitoring (MAS-CM) model that supports the performance and security of tasks gathering, scheduling and execution processes in large-scale service-oriented environments. The paper in (Antamoshkin et al., 2015) outlines the general concept of multi-agent approach to develop the automation system for monitoring, forecasting, and managing emergency situations and its models and algorithms by studied different information systems included into distributed computer networks for decision making in extreme situations. All these researches are preventive and don't take into account corrective maintenance that may be appeared in some situation.

On the other hand, several research efforts have mainly focused on corrective maintenance like (Rudrapal et al., 2013), which proposed algorithm considering multiple attributes of user keystroke dynamics. They also proposed a traditional authentication in an organization for distinguishing one user than another. The works in (Liao et al., 2013), proposed the taxonomy to outline modern intrusion detection systems IDSs. Furthermore, the work in (Campos et al., 2007) reports the development of an e-monitoring and maintenance system based on web technology and mobile device. The authors in (Campos & Prakash, 2006) presented a Web and Agent Technologies in monitoring and maintenance condition of mechanical and electrical systems. We observe that many efforts were investigated in this sense but unfortunately, only focused on security field.

Regarding the corrective IT maintenance. Some works have appeared; the authors in (Elandaloussi et al., 2017) proposed a decision support framework in an IT environment which was essentially based on web services and mobile agents. These ones must still be completely solved. The work in (Abid et al., 2015) presented a new technologies to improve the maintenance process, and establish remote maintenance, using of mobile agent technology to reduce the maintenance costs and solve the problem of the unavailability of an expert in all phases of condition-based maintenance (CBM) strategy. Hence, (Elandaloussi et al., 2019) proposed a corrective decision method based on agents for IT infrastructure maintenance.

Our approach is an evolution version of the latest work presented in (Elandaloussi et al., 2019) by including text-mining in agents based DSS for IT infrastructure maintenance. The text mining methods are able to eliminate the occurrence of terms obviously incorrect or not relevant to the defined problem. Moreover, they keep just pertinent terms to decrease the response time during troubleshooting task. As a result, we allow us to treat the generated data (relevant terms) as a multi criteria problem, which is best suited to this kind of situation. Moreover, the agents coordinate and cooperate their action in the goal to have a corrective decision as it is shown on the next section.

3. PROPOSED APPROACH

3.1 General Architecture

Our **MAITD-2** architecture is a multi-layered architecture which is classified into three layers: Presentation, Interpretation, and Data Layer. Each layer incorporates several agents with some components for typical functionalities. The first layer that is named the presentation layer is constituted of two types of agent: the participant and expert agent. These agents act as an interface with the system to deal with all input or output data. Additionally, the rest of the agents are included in the second layer in the middle of our architecture. In our approach, this later is considered as a kernel with its functions and methods that are modeled in java environment. Basically, data layer is a physical layer that brings together all data source which are requested. In the following, we separately described different layers which are defined before by focusing on the main tasks of each agent that they contain:

3.1.1 Presentation Layer

The presentation layer is responsible for the treatment and delivery of inputted information to the lowest layers. Furthermore, it is the first applicative layer in our system that is considered as the user responsible interface for displaying generated information to the user. The main agents represented in this layer are defined below:

Participant Agent: As it is shown in Figure3, especially in Terms Extraction step, this agent breaks down the problems in pertinent terms in order to treats the input information of the user. In addition, it is responsible for forwarding all system notifications to the corresponding participant. The most important module appeared in this agent named decomposition module which leads us to select from each problem the relevant terms necessary to accelerate the information retrieval by applying a set of functions and algorithms which are presented in section4.

Expert Agent: It manages personnel information of assigned expert like his agenda, preference, and profiling. So, it is responsible for accepting and refusing any meeting invitation. Additionally, it intervenes during the construction and enrichment of the domain ontology (Bendaoud et al., 2007) through validating the terms classification in their corresponding taxonomy context to preserve the quality of information. Among the included modules in this agent, a profile module that resume overall expert profile and preference. Furthermore, the scheduling module that takes place in the organization of collaborative, are built according to the expert's availabilities information's.

3.1.2 Interpretation Layer

As mentioned previously, all the rest of agents which are focused to fulfill the overall goals in MAITD-2 architecture are integrated into this layer. They are described as the following:

Analyzer Agent: The similar issues named candidate problems are filtered and selected by this agent type from several data source to construct our corpus. The functionality of some modules is presented below:

Similarity Module (SM): Allows generating our performance matrix via applying a string similarity and knowledge similarity measures, which are defined in section 5 of this paper.

Aggregation Module (AM): Is used to have an efficient ranking of solutions which implements a specific function named WASPAS (Weighted Aggregated Sum Product Assessment) algorithm introduced in section 6.

Solution Agent: It resides with the expert agent in the fact of searching and selecting all interesting solutions via consulting the Universal Description Discovery and Integration (UDDI) registry for new published solutions and knowledge bases for solving problems.

Meeting Agent: The main tasks of this agent is to prepare a collaborative session in the case where any solutions have been found from different data sources also where the problem is considered as newly encountered or incorrectly expressed as mentioned in the figure10.

3.1.3 Data Layer

This layer contains all data that we wanted to process with our system according to different scenarios such as the domain ontology, the UDDI registry, Tree Tagger Dictionary, and Knowledge Base which are described as below:

Tree Tagger Dictionary: It is a labeler for annotating text who provides the grammatical category and lemma information to group different words from the same family. In our approach, it is used during data pre-processing task.

Global Knowledge Bases: It is a centralized information repository, which includes the various interventions and scenarios that are necessary for designing our corpus. Also, it is used as an initial source of solutions and is considered as the shared database with all agents.

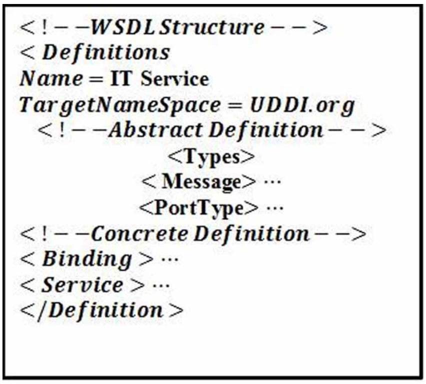
UDDI: The Universal Description, Discovery, and Integration is a directory service that manages information about service providers. It is a central location where customers or agents can dynamically discover the published stored solutions by different organizations. Furthermore, each solution on UDDI is defined by its description WSDL (Web Service Description Language), this one is based on XML notation to describe corresponding web services so it contains all the necessary information that any client application would require to use the relevant web service. As is cited in (Web Services Structure, n.d.), the general structure of a WSDL is described as shown in figure 1.

Let us define in Table 1 a brief description of the tags appeared in this structure that it corresponds to the basic web service.

To illustrate this description, a simple example along the defined web service is giving an output using SOAP binding as shown in Figure 2.

Domain Ontology: It represents and resumes all relevant concepts in our IT domain. As a consequence, it is developed to generate a similar concept to our pertinent terms that appeared in the posed problem at hand and to calculate the assigned weights for each criterion in our associated decisional performance table. Also, this ontology allows us to generate a multi-key term (MKT)

Figure 1. General WSDL Structure



```
<!--WSDLStructure-->
<Definitions
Name = IT Service
TargetNameSpace = UDDI.org
  <!--Abstract Definition-->
              <Types>
            <Message> ···
           <PortType> ···
<!--Concrete Definition-->
<Binding> ···
<Service> ···
</Definition>
```

Table 1. Web services tags description

Tags	Descriptions
<Types>	Provides information about appeared data in the WSDL file(XML elements)
<Message>	Describes the specific protocol and the appropriate data format of solutions. Is used to define a message exchanged between web services.
<PortType>	A set of operations supported by the provider of solutions. Is used to combine multiple messages into a single operation.
<Binding>	Describes the specific protocol and the appropriate data format of solutions. It defines exactly how each operation will take place over network.
<Service>	Defines the IP address and corresponding port of provider. It says where the services can be accessed from.

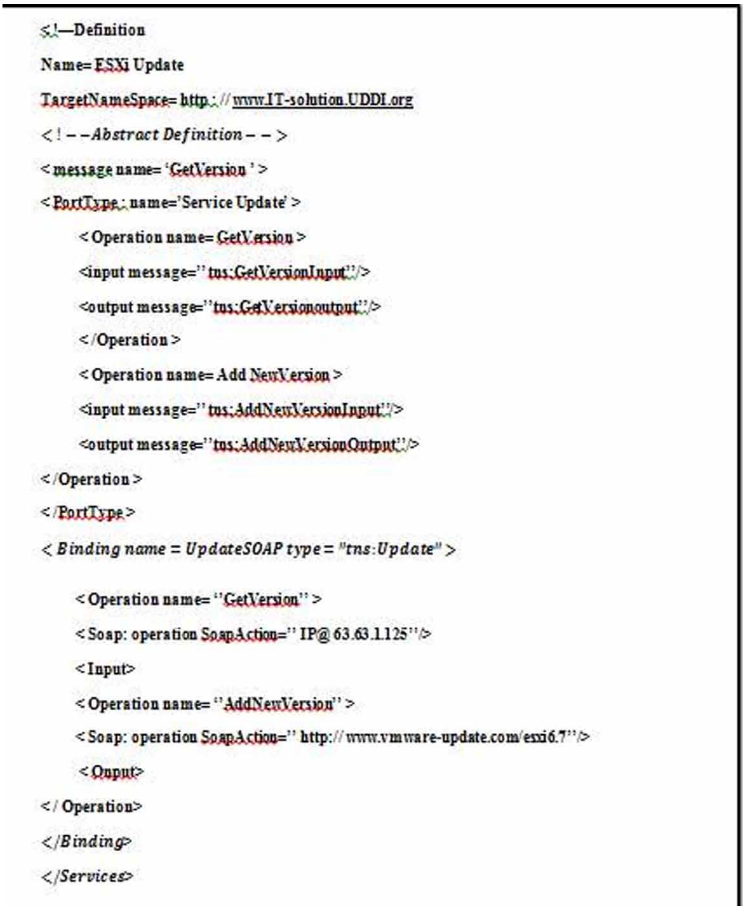
and a specified key term (SKT) that will be defined later in section4. Figure 3 is an upgrade of described architecture in (Elandaloussi et al., 2019) with some numbered flows to illustrate actions sequencing and relationships between all components.

This architecture allows an easier maintenance in which the roles, the scheduling intervention, and the relationship between agents are specified before. Indeed, it allows monitoring of the posed problem life cycle. However, in case that one problem is contributed in all described layers (layer1, layer2, and layer3), in which it is considered as a solved problem. Moreover, more reliability and independence are given to the agents that permit us to be able to update the components of one layer without impacting the other layers. Indeed, we finalize the definition of our architecture by citing the functionality of some modules that fulfill the global structure of all agents as common components like private knowledge and coordination modules.

Private knowledge Base (PKB): It contains all knowledge concerning the particular agent for example algorithm, function parameters, and indexes on the shared domain ontology. So, it's not required that all agents share the same information.

Coordinate Module (CM): This module ensures the coordination between our agents in a decentralized way. As a result, it manages and controls general decision-making.

Figure 2. VMWARE Esxi update Web Service Example



```
<!—Definition
Name= ESXi Update
TargetNameSpace= http.://www.IT-solution.UDDI.org
<! − −Abstract Definition − − >
< message name= 'GetVersion ' >
< PortType : name='Service Update' >
        < Operation name= GetVersion >
        <input message='' tns:GetVersionInput''/>
        <output message=''tns:GetVersionoutput''/>
        < /Operation >
        < Operation name= Add NewVersion >
        <input message='' tns:AddNewVersionInput''/>
        <output message=''tns:AddNewVersionOutput''/>
< /Operation >
< /PortType >
< Binding name = UpdateSOAP type = "tns:Update" >

       < Operation name= ''GetVersion'' >
       < Soap: operation SoapAction='' IP@ 63.63.1.125''/>
       < Input>
       < Operation name= ''AddNewVersion'' >
       < Soap: operation SoapAction='' http:// www.vmware-update.com/esxi6.7''/>
        < Ouput>
</ Operation>
</Binding>
</Services>
```

For more information about the numbered flows appeared in this architecture, brief descriptions are given in Table 2.

3.2 Modules based Architecture

The modules based architecture defined in Figure 4 is described as various information-processing assets with a list of functions needed to meet a pre-defined goal. As viewed in Figure 2, considers that the troubleshooting problem includes the following steps which will be introduced in details from Section4 to Section6.

3.2.1 Step1: Data Pre-processing

It is initially necessary to efficiently deal with the problem before implementing any method. Therefore, for each problem, we start by selecting a corpus from a global knowledge base and/or the web as an external resource by applying a clustering method using the K-means algorithm. This corpus is essential for the smooth running of step1 and step2 that will be defined here. In addition, a lemmatization phase is launched by using a tree tagger tools to obtain a canonical form of words on the same family, for example, the noun, the plural, the infinitive of the verb…. Finally, we generated a list of stop word by eliminating "the empty words", "punctuation" and "carriage returns".

Figure 3. Multi Layers Agents based Architecture-2

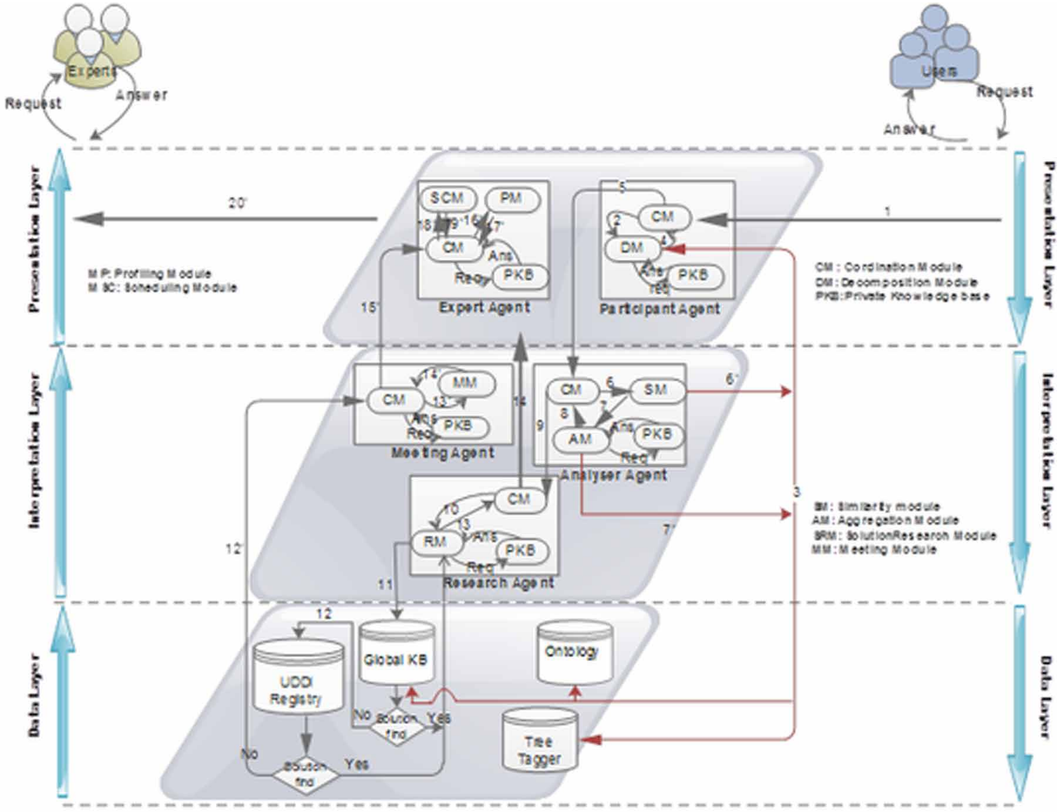
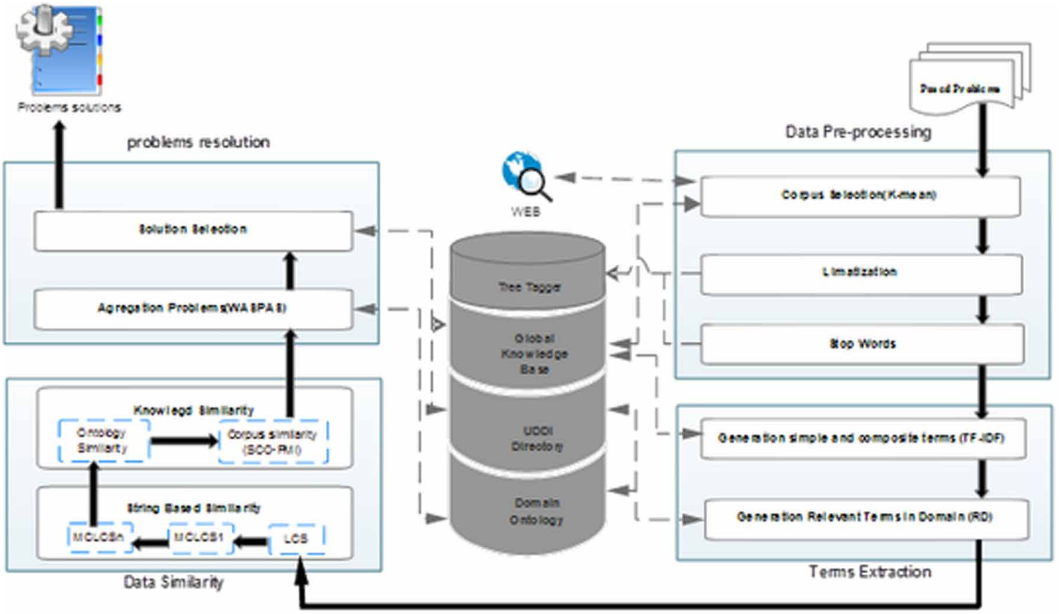


Table 2. Flow Number description

Flow Number	Description	Flow Number	Description
1	Receive Problem	12	Interrogate UDDI if no solution is found in Global knowledge
2	Decomposition of problem	13	Report before coordination
3	Interrogate External Data Source	14	Transmit Report to participant
4	Decomposition Report	15	Inform Expert Agent if no solution is found
5	Coordinate with Analyzer Agent	16	Checking Expert Profile
6	Measure of Similarity	17	Checking Expert Availability
7	Aggregate Solutions	18	Generate Initial expert list
8	Aggregation Report	19	Generate Final Expert List
9	Coordinate with Research Agent	20	Prepare Meeting
10	Research Corpus or Solution	20’	Inform Expert and Participant
11	Interrogate Global Knowledge base		

Figure 4. Module based Architecture for MAITD-2



3.2.2 Step2: Terms Extraction

The automatic key terms extraction task aims to extract the most representative words in the inputted problems. On the current step, we evaluate the important words that appear together in a corpus focusing on TF-IPF algorithm which will be presented in section4.1. Furthermore, the relevance in the domain algorithm that will be defined in section4.2, is implemented to extract the pertinent words by interrogating the domain ontology and a corpus generated before.

3.2.3 Step3: Data Similarity

The similarity measure quantifies the relationship between two objects. Therefore, this step is divided into two phases. The first phase is to eliminate the appeared errors after a bad identification of the problem by combining three types of algorithm, which is well explained in section 5.2. The second phase is according to two concepts. Firstly, the similarity based ontology to obtain semantic relations. Secondly, the similarity based corpus on information gained from large corpora to determine the similarity between words that do not co-occur frequently by implementing an SCO-CMI algorithm that will be defined in section5.3.

3.2.4 Step4: Problems Resolution

To obtain the sets of relevant solutions related to the studied problem, we start with the aggregation of a generated solution by implementing a WAS-PAS algorithm as defined in section 6 before selecting the solution from the UDDI.

4. TERMS EXTRACTION

Terms extraction is the task of identifying single or multi-word expressions and it is divided into two categories: supervised or unsupervised methods. It was useful in many research area likely information retrieval (Medelyan & Witten, 2008), document summarization (Litvak & Last, 2008) and document clustering (Hammouda et al., 2005), Key phrases extraction (Liu et al., 2010), automatic Naturel languages processing (ANLP) (Romero et al., 2012), text mining (Allahyari et al., 2017). The extraction tasks are focused on statistical, linguistic, clustering and based graph technical; all are trained around supervised or unsupervised approaches as mentioned before. In the statistical terms extraction, many variations have been proposed in the literature including term frequency-inverse document frequency (TF.IDF) as in (Hisamitsu et al., 2000) who measures certain document term's importance in relation to other documents in the same collection. This method seems to be not appropriate to the terms appearing rarely but which are important on a general point of view. However, (Paukkeri et al., 2008; Paukkeri & Honkela, 2010) is a statistical approach that outperforms both supervised and unsupervised baseline methods at the same time. As linguistic technical, the authors in (Liu et al., 2011) use word alignment models (WAM) in statistical machine translation (SMT) and propose a unified framework for key phrase extraction. In addition, to illustrate clustering methods, the authors of (Liu et al., 2009) propose an unsupervised method for key phrase extraction. Firstly, the method finds relevant terms by leveraging clustering techniques, which guarantees the document to be semantically covered by these exemplary terms. Then the key phrases are extracted from the document using the exemplary terms.

On the other hand, as defined in (Romero et al., 2012) they are two classes of terms: multi-domain key terms (MKT) and specific domain key terms (SKT). The authors of this one combine between two approaches in simple reasons to deal with a limit of frequency-based approach. A thesaurus (Wikipedia) based approach is employed to detect MKT and a controlled dictionary due to define its SKT. Therefore, to improve the terms extraction quality it is very important to take into account the single concepts and the composite ones.

Indeed, among all the different issues treated in our work. By presenting the current section we hope to respond to this question "how to effectively filter and deal with the incoming problems in order to extract the relevant terms that should cover the whole problem description". In this fact, we have proposed a hybrid method to obtain good results by regrouping two statistical methods and one clustering-based method. Firstly, we combine statically and semantically approach by taking into account the pertinence of terms in the domain ontology, or in the initial corpus that contains similar problems. This corpus was automatically built by using a k-means algorithm as a clustering method, but the way to do it is not specified in this work, in which the generated data set is basically focused on web and also from history interventions and scenario. Secondly, a word's frequency (TF-IPF) is used from the corpus in order to generate simple and composite terms as a second statistical method. (Mothe & Ramiandrisoa, 2016) proposed a survey analyzing these methods which are used in our approach. According to what is mentioned before, the following essential keys are derived:

4.1 Pertinence in Domain

To compute the relevance of terms in a domain, we first collected corpora in several domains via different data sources according to the IT infrastructures issues focused on K-mean clustering method. However, the techniques described in this section combine between statically approaches and semantically ones as mentioned before. Regarding (Velardi et al., 2001) a specific score, called Domain Relevance (DR), has been defined. More precisely, given a set of n domains ($D_1 \ldots D_n$). The domain relevance of a term T is computed to obtain pertinent terms (PT) using the following formula:

$$PD\left(Ti, Di\right) = \frac{P\left(Ti, Di\right)}{\sum_{i=1}^{n} P\left(Ti, Di\right)}$$

where $P(T_i,D_i\ )$ whose signified the conditional probabilities of the term T_i on the domain D_i is estimated as:

$$P\left(Ti,Di\right)=\frac{freq\left(Ti\ in\ Di\right)}{\sum_{i=1}^{n}freq\left(Ti,Di\right)}$$

In practice, according to the latest work (Berkhin, 2006) they found that the threshold PDT ($T_{i,}$ D_i) =0.35 for the Term's Domain Relevance is a generally "good" value. The following screenshot is an overview of the different measure generated by our system around one defined problem.

As shown in Figure5, the set of terms {Start, VMware, Virtual, Machine} are relevant in virtualization domain which all of them have a value greater than threshold α=0.35. So, each problem type named CP_i (candidate problem) is composed of multiple important words PT_i (pertinent terms) in several contexts. However, this corpora contains 15 different issues as candidate problems coming from 4 data sources.

4.2 Terms Frequency TF-IPF

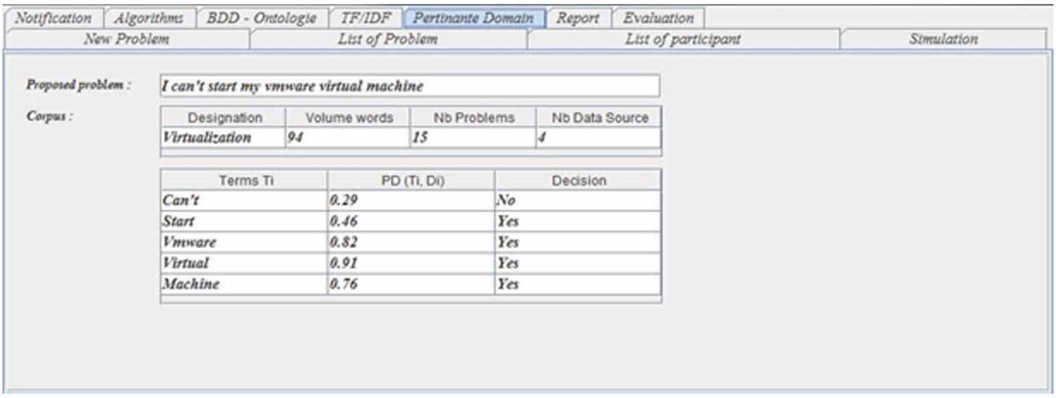
The **TF×IDF** score named **TF.IPF** (Term Frequency–Inverse Problems Frequency) in our survey is implemented to select the terms type either composed or simple in goal to compare the frequency of composed words, which are eventually used in a particular problem with the frequency of that block in a corpus. This corpus is represented by different similar problems to the defined problem generated automatically via the web or historically interventions using a k-mean method. The TF.IPF score for composed terms in the problems is given by:

$$TF-IPF\left(T,T'\right)=\frac{nb\left(T,T'\right)}{nb\left(T\right)*nb\left(T'\right)}$$

where nb(T,T') is the number of times whose T and T' are consecutively appeared on the corpus, nb(T) and nb(T') count the number of time that T and T' appear on the same corpus.

In this way, if the terms T and T' appear consecutively together very often we are going to consider that T and T' construct the same concept named T T'. The minimal threshold is estimated at Ω=0.7 in order to have satisfactory results. This value is deduced after 50 requests with12 different Corpus. Figure 6 presents the results which are obtained after executing our TF.IPF method with vector representations of one posed problem and statistical information about our corpus.

Figure 5. Pertinent in domain



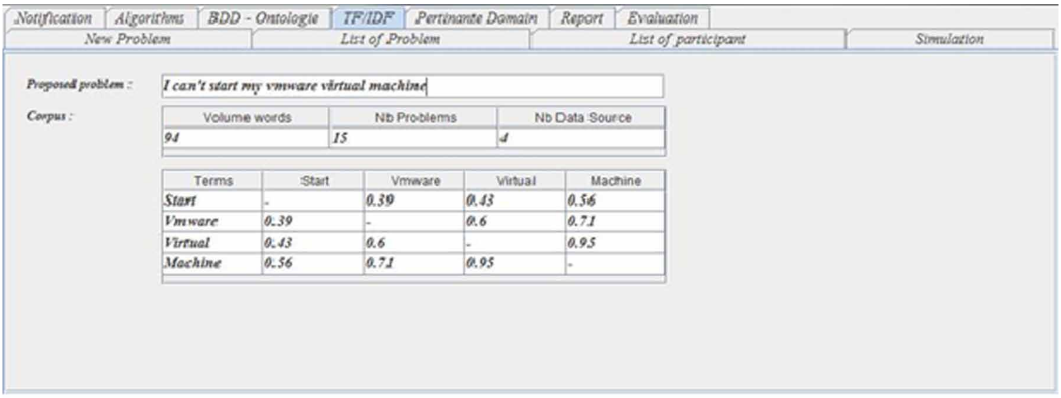
Proposed problem : I can't start my vmware virtual machine

Corpus :

Designation	Volume words	Nb Problems	Nb Data Source
Virtualization	94	15	4

Terms Ti	PD (Ti, Di)	Decision
Can't	0.29	No
Start	0.46	Yes
Vmware	0.82	Yes
Virtual	0.91	Yes
Machine	0.76	Yes

Figure 6. TF-IDF Algorithm



Proposed problem : I can't start my vmware virtual machine

Corpus :

Volume words	Nb Problems	Nb Data Source
94	15	4

Terms	Start	Vmware	Virtual	Machine
Start	-	0.39	0.43	0.56
Vmware	0.39	-	0.6	0.71
Virtual	0.43	0.6	-	0.95
Machine	0.56	0.71	0.95	-

After execution of the algorithm, the set of the composite terms appeared in the current posed problem is {Virtual machine} because the TP-IPF value between the both of words ''Virtual'' and ''Machine'' has a value higher than threshold Ω cited before. Moreover, the simple terms dataset is {start, VMware}.

5. TEXT SIMILARITY MEASURE

5.1 General Context

The similarity measure between textual documents is one of the important issues in several disciplines such as text classification, text data analysis (Berkhin, 2006), knowledge extraction from textual data (Text Mining) or retrieval information (Medelyan & Witten, 2008), document clustering (Berkhin, 2006). Also finding similarities between two texts consists of comparing the words of them. Words can be similar in two ways: lexically and semantically. On one hand, words are similar lexically if they have a similar character sequence. On the other hand, words are similar semantically if they have been used in the same context (Islam & Inkpen, 2008). In the following, we will give a short description of these measures with some corresponding screenshots to our developed system after adopted them on the studied context.

5.2 Lexical Similarity Measure (String Bases Similarity Measure)

This method does not require understanding the vocabulary or grammar of the text's language. A variety of algorithms have emerged in goal to meet these requirements: Longest Common Substring (LCS), Damerau-Levenshtein, N-gram NGD, Cosine similarity, Euclidean distance, Jacquard similarity, PMI-IR, SCO-PMI, HAL, LSA, GLSA, CL-ESA …. In order to estimate and eliminate a bad description of the problem, our survey used string-based approach as lexical similarity by deploying three different modified versions of the LCS algorithm (longest common subsequence),. In Fact, the string bases similarity between a posed problems P_i and candidate problems PC_j is estimated by calculating the corresponding data matrix SSMTT'$_{ij}$ means String Similarity Matrix Terms to Terms that have c columns and r rows according to the T_i pertinent terms of the posed problem and T'$_j$ for candidate problem respectively. In addition, ssmtt'$_{ij}$ represents a correspondent string similarity measure values. This later is through using the different modified versions of the Longest Common Subsequence LCS algorithm as defined in (Velardi et al., 2001). So the string similarity between two terms is calculated with the function defined as:

$$\boldsymbol{Smtt}_{ii'} = \frac{length\ \ LCS\left(Ti,Ti'\right)^2}{3length\left(Ti\right)*length\left(Ti'\right)} + \frac{length\,MLCS1\left(Ti,Ti'\right)^2}{3length\left(Ti\right)*length\left(Ti'\right)} + \frac{length\,MLCSn\left(Ti,Ti'\right)^2}{3length\left(Ti\right)*length\left(Ti'\right)}$$

where LCS means longest common subsequence needs not to be consecutive and MCLCS_1 is a maximal consecutive longest common subsequence starting at first character, finally MCLCS_n signified a maximal consecutive longest common subsequence starting at any character n.

Figure 7 is the result of the implementation and adaptation of the LCS algorithm in our system.

As shown in Figure 7 our $\text{MSSTT'}_{ii'}$ matrix contains 4 colums { start, vmware, virtual, machine} and 4 rows: {Vmkernel,service,stop,response}. Hence, the similarity values between matrix's terms are figured before like: $\text{mstt'}_{\{\text{vmkernel,start}\}}$= 0.0082 $\text{mstt'}_{\{\text{vmkernel,vmware}\}}$= 0.165, $\text{mstt'}_{\{\text{service,machine}\}}$=0.0875.

5.3 Semantic Similarity Measure

To ensure semantically similarity, various algorithms have appeared. We can define two main axes for these algorithms:

5.3.1 Corpus-Based Approach

Principally, we now take the semantic similarity, in which we computed the SOC-PMI (Second Order Co-occurrence Pointwise Mutual Information) (Aminul & Inkpen, 2006) similarity values by using our corpus as a source of frequencies and contexts. Our algorithm is used to construct the n*m semantic similarity matrix between T_i and $T'_{i'}$, terms that correspond to PP_i (Posed problem) and PC_j (Candidate Problem) respectively, namely $\text{MSMTT}_{ii'}$, as shown in Figure 8.

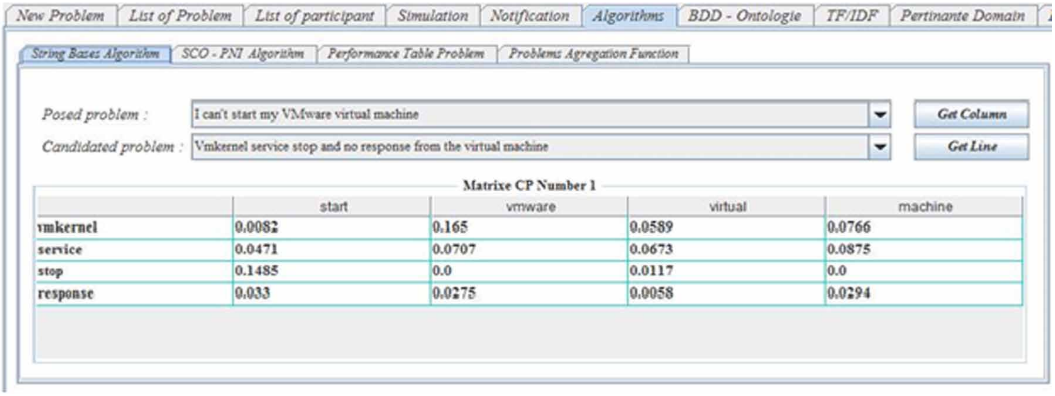
As shown in Figure 8, both terms vmkernel and virtual are semantically similar because they co-occur often with the common words on the same corpus.

5.3.2 Knowledge-Based Approach

Knowledge-based word similarity approaches are based on a semantic network of words, certainly, the structure of the database is taxonomy in which each node is a concept such as WordNet (Fellbaum, 1998). Many algorithms are developed according to this context as defined in **RES**, **LIN**, **JCN**, **LCH**, **PATH**, **WUP**.

Conversely, hybrid methods use the both corpus-based measures and knowledge-based measures to determine the text similarity. Several researches are realized in this sense as the work described in (Mclean et al., 2006; Mihalcea et al., 2006). Moreover, two semantically approaches one based

Figure 7. String Bases Similarity Measure



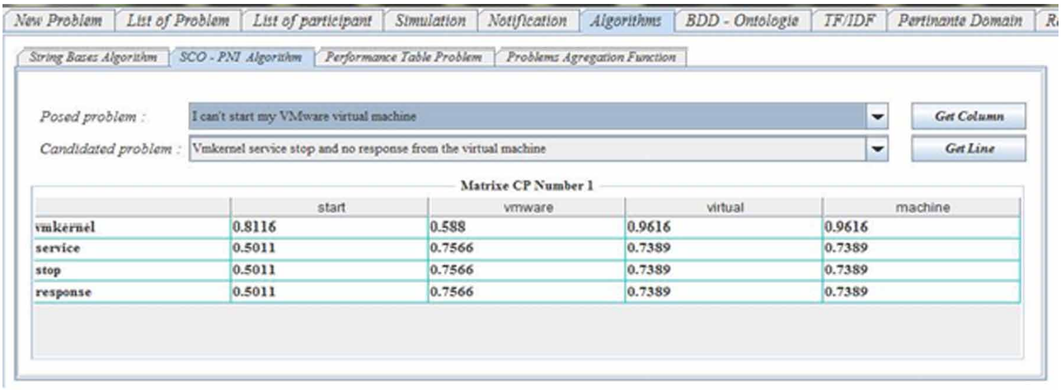
Posed problem : I can't start my VMware virtual machine

Candidated problem : Vmkernel service stop and no response from the virtual machine

Matrixe CP Number 1

	start	vmware	virtual	machine
vmkernel	0.0082	0.165	0.0589	0.0766
service	0.0471	0.0707	0.0673	0.0875
stop	0.1485	0.0	0.0117	0.0
response	0.033	0.0275	0.0058	0.0294

Figure 8. Semantic Similarity Measure terms to terms (MSMTT$_{ij}$ matrix) using SCO-PMI Algorithm



Posed problem : I can't start my VMware virtual machine

Candidated problem : Vmkernel service stop and no response from the virtual machine

Matrixe CP Number 1

	start	vmware	virtual	machine
vmkernel	0.8116	0.588	0.9616	0.9616
service	0.5011	0.7566	0.7389	0.7389
stop	0.5011	0.7566	0.7389	0.7389
response	0.5011	0.7566	0.7389	0.7389

corpus and the second based knowledge using SOC-PMI algorithm are used to calculate the similarity between two words that do not co-occur frequently because they co-occur with the same neighboring words, also by integrating our domain ontology as semantically database. Indeed, merging between ontology and corpus is one of the most prominent aspects in our approach.

6. AGGREGATION FUNCTION

Aggregation function product a specific calculation on a set of values, and returns a single value. In fact, at the moment of constructing the set of **MSMTT$_{ii'}$** matrix corresponding to the different candidates problems as shown in Figure6, we regroup all the **MSMTT$_{ii'}$** matrix together in the same **MSMTP$_{ij}$** (semantic Similarity Matrix Term to Problem) matrix to obtain our performance table with n column (pertinent terms of P$_i$) and r rows (similar problems to P$_j$). As displayed in Figure 9 by applying the following algorithm:

```
$Input : Ti, T'i', MSMTTii' *$
$Output : MSMTPij$   "$semantic\,similarity\,matrix\,terms\,Ti\,to\,problem\,Pj$"
$For\,i \leftarrow 1\,to\,c\,do$   "$c\,is\,MSMTTii'column\,number$"
        $For\,i' \leftarrow 1\,to\,r\,do$   "$r\,is\,MSMTTii'row\,number$"
$For\,j \leftarrow 1\,to\,k\,do$   "$k\,is\,candidate\,problem\,number$"
        $msmtpij \leftarrow \max\left(msmttij\right)$
          $End$
    $End$
$End$
```

In order to synthesize the semantic performance into a global score as **sim(P$_i$, PC$_j$)** in the sense of extracting similar most problems. Several aggregation functions have appeared for example **MOORA**, **COPRAS, TOPSIS, WASPAS, EDAS….**, as it is defined in the work (Fomba, 2018). In this survey, we use **WASPAS** (**Weighted Aggregated Sum Product Assessment)** method which is one of the most recently developed multi criteria decision making methods that is designed to address quantitative problems. Additionally, it is simple to implement and gives in general satisfactory results. Ultimately, the problem is considered as incorrectly expressed or encountered, if this function gives a small aggregation value, so the meeting agents intervene to prepare collaboration sessions. However, this method combines between the weighted sum and product as is mentioned in (**). Moreover, Figure 9 is a screenshot of our aggregation:

$$Aj = 0.5\sum_{i=1}^{c}\left(msmtp\,ij\right) * wi + 0.5\prod_{i=1}^{c}\left(msmtp\,ij\right)^{wi} \ldots\left(**\right)$$

W_i corresponds to the weight of criteria i and A_j as an aggregation of alternative j (candidate problems).

As shown in Figure 9, our WAS-PAS algorithm gives an important rank. For example, problem3 is more similar to the treated problem. Additionally, all aggregations A_j which are higher than threshold Ω≈0,55 generate a good and accepted result. However, the fundamental purpose of our work is how exactly the weight of criteria in the column is generated from our domain ontology.

7. EXPERIMENTATION AND VALIDATION

This section presents some interfaces of the **MAITD-2** defined on section3, where the functions and multi-agent deployment are respectively implemented by using Java Net Beans environment and Jade Platform for agent's creation. Moreover, WAMP server for database deployment, Tree Tagger as labeler of lemmatization and Protégé tools for domain ontology development. In our approach, the corpus is created from real IT infrastructure database problems generated by specialized experts.

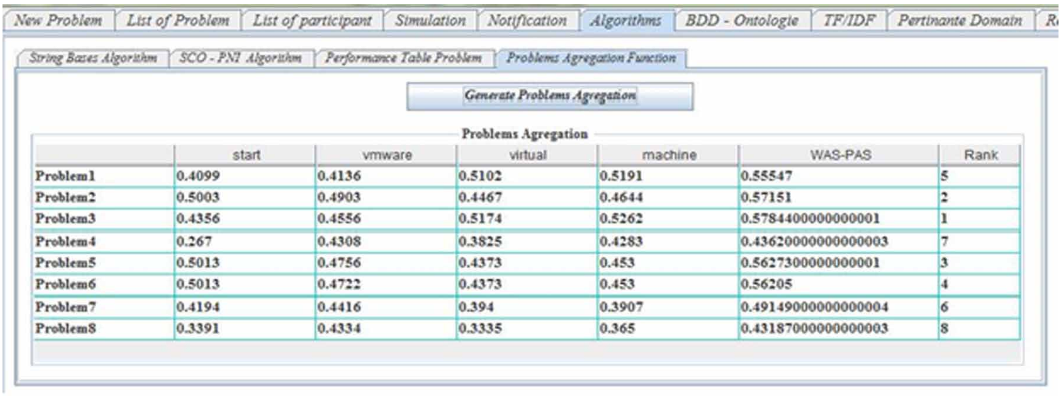
7.1 Description Problems (Layer1)

After sign-in in his private session, the user identifies his problem and the corresponding domain as shown in Figure 10.

7.2 Data Pre-Processing (Layer1)

Figure 11 is a result of step1 which is a part of the first layer (presentation layer) that is executed by the participant agent. For the posed problem "no start of virtual machine on shell mode", the system generates 9 candidate problems numbered in the Figure 10 from1 to 9 by applying a k-means algorithm. Also, we present the initial pertinent terms list after lemmatization and words stop tasks. In the current step, our system selects and display the pertinent terms for each candidate problem by using all algorithms presented before in section4.

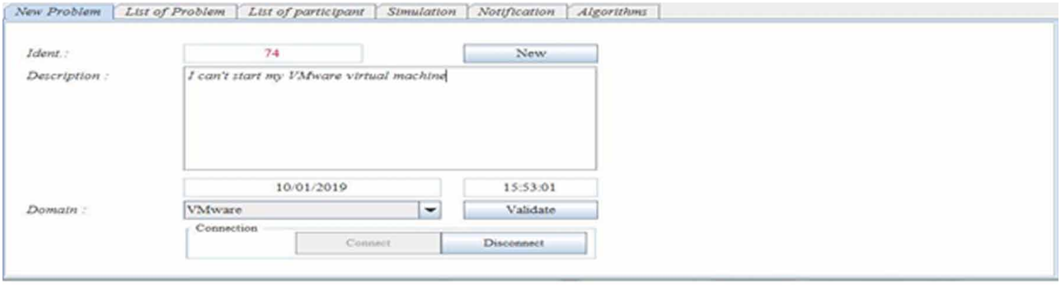
Figure 9. Semantic similarity matrix terms to problems (MSMTP$_{ij}$ matrix) with aggregation



Problems Agregation

	start	vmware	virtual	machine	WAS-PAS	Rank
Problem1	0.4099	0.4136	0.5102	0.5191	0.55547	5
Problem2	0.5003	0.4903	0.4467	0.4644	0.57151	2
Problem3	0.4356	0.4556	0.5174	0.5262	0.5784400000000001	1
Problem4	0.267	0.4308	0.3825	0.4283	0.43620000000000003	7
Problem5	0.5013	0.4756	0.4373	0.453	0.5627300000000001	3
Problem6	0.5013	0.4722	0.4373	0.453	0.56205	4
Problem7	0.4194	0.4416	0.394	0.3907	0.49149000000000004	6
Problem8	0.3391	0.4334	0.3335	0.365	0.43187000000000003	8

Figure 10. Problem Description screenshot



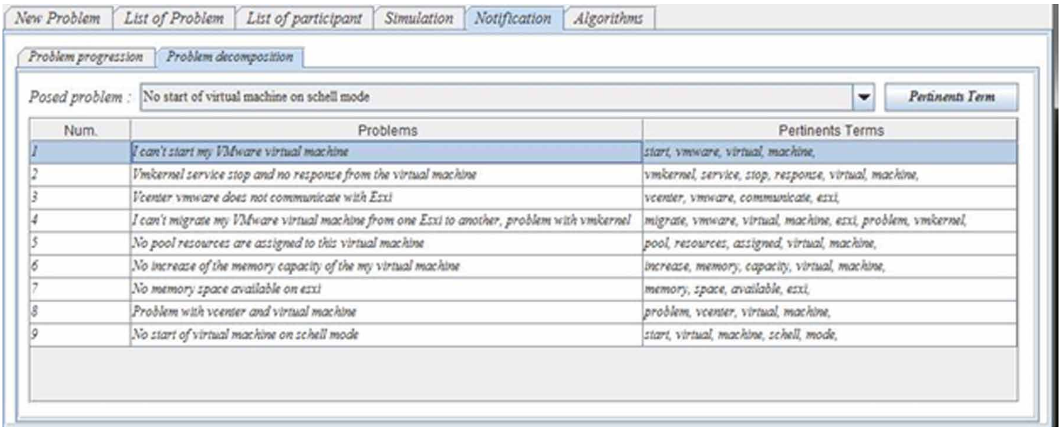
Ident. : 74

Description : I can't start my VMware virtual machine

10/01/2019 15:53:01

Domain : VMware

Figure 11. Data Pre-processing Screenshot



Posed problem : No start of virtual machine on schell mode

Num.	Problems	Pertinents Terms
1	I can't start my VMware virtual machine	start, vmware, virtual, machine,
2	Vmkernel service stop and no response from the virtual machine	vmkernel, service, stop, response, virtual, machine,
3	Vcenter vmware does not communicate with Esxi	vcenter, vmware, communicate, esxi,
4	I can't migrate my VMware virtual machine from one Esxi to another, problem with vmkernel	migrate, vmware, virtual, machine, esxi, problem, vmkernel,
5	No pool resources are assigned to this virtual machine	pool, resources, assigned, virtual, machine,
6	No increase of the memory capacity of the my virtual machine	increase, memory, capacity, virtual, machine,
7	No memory space available on esxi	memory, space, available, esxi,
8	Problem with vcenter and virtual machine	problem, vcenter, virtual, machine,
9	No start of virtual machine on schell mode	start, virtual, machine, schell, mode,

7.3 Preparing Collaborative Session (Layer2)

The collaboration sessions have been prepared by the meeting agents. Indeed, this session will be scheduled in the case when no solution is found. This meeting agent collaborates with all participant agents and expert ones to elaborate and control this session.

As shown in Figure 12, the meeting agent gives a global overview of different problems with their evolution information meeting like date/time, and information about the invited participants and forum's number which are included.

7.4 Monitoring Problem Possessing (Multi Layers)

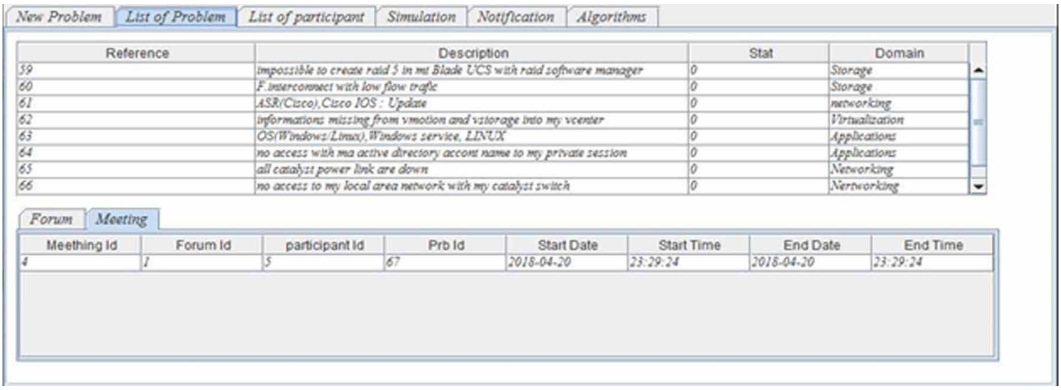
The monitoring task is established by the participant agent. This one supervises in the same interface all problems progress as shown in Figure 13. Especially, it resumes the sequencing of overall tasks.

7.5 Validation

The evaluation of solutions follows several methods. Here, we highlight one of them which is mainly used in various applications, it is about F-measure algorithm. This one is considered as a harmonic mean of recall and precision with the maximum possible is 1. The formula for F-measure is given as:

$$F - measure = \frac{2\,precision * recall}{precision + recall}$$

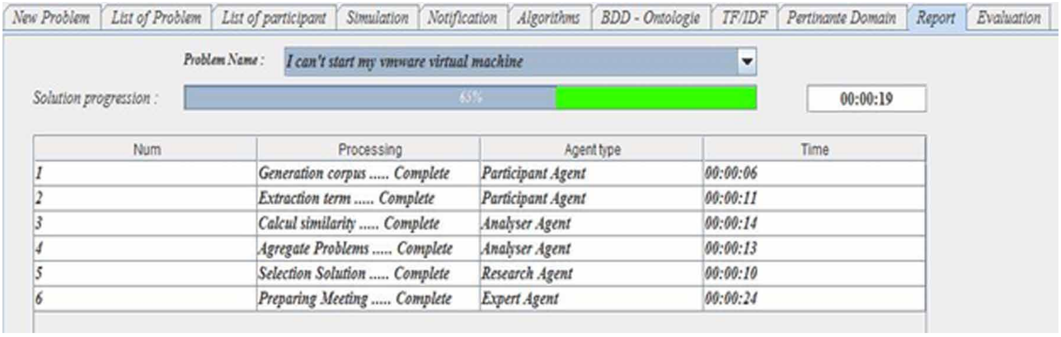
Figure 12. Preparing Collaborative Session Screenshot



Reference	Description	Stat	Domain
59	impossible to create raid 5 in mt Blade UCS with raid software manager	0	Storage
60	F.interconnect with low flow trafic	0	Storage
61	ASR(Cisco),Cisco IOS : Update	0	networking
62	informations missing from vmotion and vstorage into my vcenter	0	Virtualization
63	OS(Windows/Linux),Windows service, LINUX	0	Applications
64	no access with ma active directory accont name to my private session	0	Applications
65	all catalyst power link are down	0	Networking
66	no access to my local area network with my catalyst switch	0	Nertworking

Meething Id	Forum Id	participant Id	Prb Id	Start Date	Start Time	End Date	End Time
4	1	5	67	2018-04-20	23:29:24	2018-04-20	23:29:24

Figure 13. Participant Monitoring Screenshot



Problem Name : I can't start my vmware virtual machine

Solution progression : 65% 00:00:19

Num	Processing	Agent type	Time
1	Generation corpus ..... Complete	Participant Agent	00:00:06
2	Extraction term ..... Complete	Participant Agent	00:00:11
3	Calcul similarity ..... Complete	Analyser Agent	00:00:14
4	Agregate Problems ..... Complete	Analyser Agent	00:00:13
5	Selection Solution ..... Complete	Research Agent	00:00:10
6	Preparing Meeting ..... Complete	Expert Agent	00:00:24

For precision, we are able to usually define accurately the ratio of the correct solutions divided by the number of the returned results as below:

$$precision = \frac{SPPN}{SPPN + SNPPN}$$

SPPN: represent Selected Pertinent Problem Number.

SNPPN: represent Selected No Pertinent Problem Number.

However, for recalling, we will be ready to calculate the percentage of relevant solutions that are returned by the system like the following:

$$recall = \frac{SPPN}{SPPN + NSPPN}$$

NSPPN: No Selected Pertinent Problem Number

To prove the validity of our system and to ensure the quality of solutions, we have designed a set of experiments by using the following methodology:

- Description of 50 different IT problems in several domains.
- Evaluation of the generated results by our experts in the field of IT diagnostics in order to classify the proposed solutions relevantly and irrelevantly before calculating the precision, recall, and F-measure. The result of this step is presented in Figure 14.
- The dataset used in the experiments is a collection of 50 posed problems and 180 corresponding candidate problems to construct several corpora. This corpus was built using WWW and Global Database. The Current corpus has only about 120 words spread over 9 different problems.

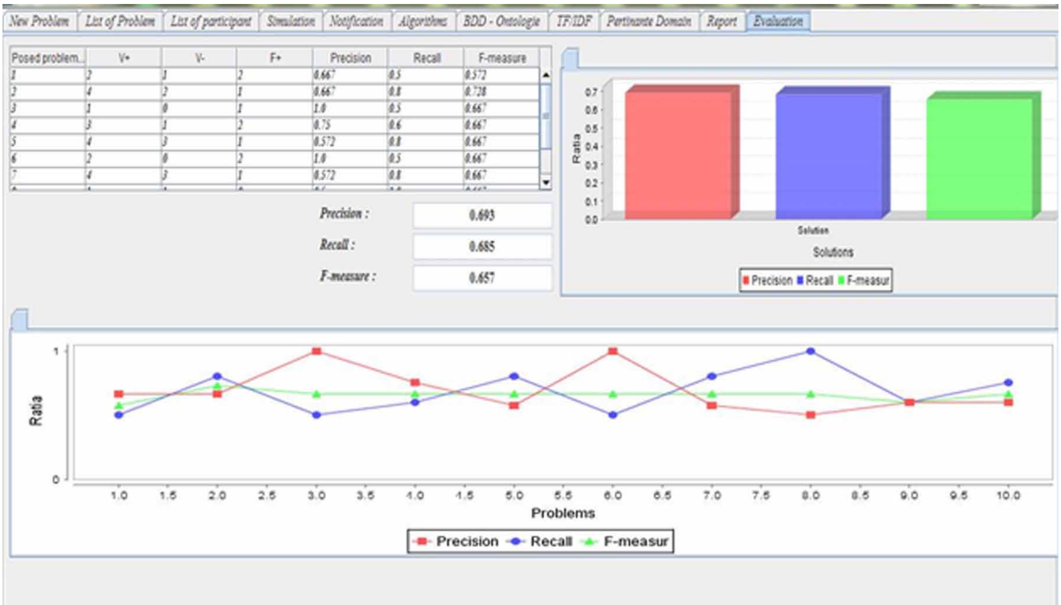
In Figure 14, the precision-recall and F-measure of various posed problems are given.

In Figure14, we observe that the average of measure for our system regarding the studied dataset is respectively estimated at 0.693, 0.685, 0.657 for precision, recall and f-measure. Furthermore, the experimentation has shown that the minimum acceptable values for each function are estimated at 0.58. As a consequence, these values seem to be very encouraging. Thus, we can say that the proposed approach proved to be a good solution. Furthermore, the depicted graphs in Figure 14 show the result of F-measure, Recall and Precision for establishing the methodology proposed.

8. CONCLUSION

In this paper, a Multi-layered architecture for IT Diagnostic named MAITD-2 is proposed. However, we developed an unsupervised clustering-based method which is a Multi-Criterion Decision Making (MCDM) framework. The ultimate goal of this work is to enforce the corrective operational maintenance among a set of agents, taking into account the complexity and the dynamic representation of problems. Firstly, the clustering method groups all candidate problems into the same clusters as

Figure 14. Precision, Recall and F-measure for MAITD-2



Posed problem...	V+	V-	F+	Precision	Recall	F-measure
1	2	1	2	0.667	0.5	0.572
2	4	2	1	0.667	0.8	0.728
3	1	0	1	1.0	0.5	0.667
4	3	1	2	0.75	0.6	0.667
5	4	3	1	0.572	0.8	0.667
6	2	0	2	1.0	0.5	0.667
7	4	3	1	0.572	0.8	0.667

Precision : 0.693
Recall : 0.685
F-measure : 0.657

similar to the initial posed problem. Secondary, the system extracts automatically in an unsupervised way the terms that several experts identified them as pertinent in order to have coverage of the entire addressed problem. In addition, we have shown how the similarity methods and solution aggregation are implemented in our work before selected a specified solution. For this reason, we use multi-agent systems to search for the best one. Moreover, the current approach provides the monitoring services and a corrective diagnostic, which is able to react independently to the requested IT problem. Also, it could be adapted to other domains by introducing corresponding ontologies. As future work, we plan to complete the proposed approach to achieve preventive maintenance instead of a corrective one. On the other hand, we will show how the criteria weight is calculated from the developed domain ontology and the different mechanism implemented to integrate a solution from the UDDI.

REFERENCES

Abid, K., Mouss, H., Kazar, O., & Kahloul, L. (2015). A Novel Approach for Mobile Maintenance Using Mobile Agents Technology and Mobile Devices. *Journal of Advanced Manufacturing SystemsVol.*, *14*(02), 55–74. doi:10.1142/S0219686715500055

Allahyari, Pouriyeh, & Assef. (2017). *Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques*. Academic Press.

Aminul & Inkpen. (2006). *Second Order Co-occurrence PMI for Determining the Semantic Similarity of Words*. School of Information Technology and Engineering, LREC.

Antamoshkin, O., Antamoshkina, O., & Smirnov, N. (2015). Multi-agent automation system for monitoring, forecasting and managing emergency situations. *XIX International Scientific Conference Reshetnev Readings*.

Bendaoud, E. R., Hacene, A. M. R., Toussaint, Y., Delecroix, B., & Napoli, A. (2007). Construction d'une ontologie à partir d'un corpus de textes avec l'ACF. 18eme journées francophones d'ingénieries de connaissances, Grenoble, France.

Berkhin, P. (2006). *Survey of Clustering Data Mining Techniques*. Springer-Verlag. doi:10.1007/3-540-28349-8_2

Bukhsh. (2019, April 1). Predictive maintenance using tree-based classification techniques: A case of railway switches. *Transportation Research Part C, Emerging Technologies*, *101*, 34–54.

Callewaerta, P., Verhagena, W. J. C., & Curran, R. (2017). Integrating maintenance work progress monitoring into aircraft maintenance planning decision support. *6th CEAS Air and Space Conference*.

Campos, J., Jantunen, E., & Prakash, O. (2007). Development of a Maintenance System Based on Web and Mobile Technologies. *Journal of International Technology and Information Management*, *16*(4).

Campos, J., & Prakash, O. (2006). information and communication technologies in condition monitoring and maintenance. *12th IFAC symposium on information control problems in manufacturing, 39*, 3-8.

Daniel, G., Agnieszka, J., Joanna, K., & Sabri, P. (2018). Using a Multi-Agent System and Artificial Intelligence for Monitoring and Improving the Cloud Performance and Security. *Future Generations Computer Systems, 86*, 1106-1117.

Elandaloussi, S., Taghezout, N., & Zaraté, P. (2017). A Collaborative Solution for IT Infrastructure Maintenance Based on Web Services and Mobiles Agents. Academic Press.

Elandaloussi, S., Zarate, P., & Taghezout, N. (2019). A Multi Agent Architecture for IT nfrastructure Diagnostic. *International Conference on Decision Support System Technology ICDSST2019*, Madere, Portugal.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge. doi:10.7551/mitpress/7287.001.0001

Fomba, S. (2018). *Un système de recommandation pour le choix de l'opérateur d'agrégation* (PhD). Toulouse University, Institut de recherche en informatique de Toulouse.

Haack, J. N., Fink, G. A., Maiden, W. M., McKinnon, D., & Templeton, S. J. (2011). Ant-based cyber security. *Proceedings - 2011 8th International Conference on Information Technology, ITNG*, 918–926. doi:10.1109/ITNG.2011.159

Hammouda, K., Matute, D., & Kamel, S. (2005). CorePhrase: keyphrase extraction for document clustering. *Proceedings of MLDM*.

Hisamitsu, T., Niwa, Y., Nishioka, S., Sakurai, H., Imaichi, O., Iwayama, M., & Takano, A. (2000). Extracting terms by a combination of term frequency and a measure of term representativeness. *Terminology*, *6*(2), 211–232. doi:10.1075/term.6.2.06his

Islam & Inkpen. (2008). Semantic text similarity using corpus based word similarity and string similarity. *Transaction on Knowledge Discovery From Data*.

Jahanbin, A., Ghafarian, A., Seno, S. A. H., & Nikookar, S. (2013). A computer forensics approach based on autonomous intelligent multi-agent system. *International Journal of Database Theory and Application*. doi:10.14257/ijdta.2013.6.5.01

Kendrickm, P., Criado, N., Hussain, A., & Randles, M. (2018). A self-organising multi-agent system for decentralized forensic investigations. *Expert Systems with Applications*, *102*, 12–26. doi:10.1016/j.eswa.2018.02.023

Kent, M. D., Costello, O., Phelan, S., & Petrov, K. (2017). *Cost Oriented Maintenance management Systems for Manufacturing Processes*. Waterford Institute of Technology.

Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), 16–24. doi:10.1016/j.jnca.2012.09.004

Litvak, M., & Last, M. (2008). Graph-Based Keyword Extraction for Single-Document Summarization. *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summuarization*.

Liu, Li, Zheng, & Sun. (2009). *Clustering to Find Exemplar Terms for Keyphrase Extraction.* Academic Press.

Liu, Z., Chen, X., Zheng, Y., & Sun, M. (2011). Automatic keyphrase extraction by bridging vocabulary gap. *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, 135–144.

Liu, Z., Huang, W., Zheng, Y., & Sun. (2010). Automatic Keyphrase Extraction via Topic Decomposition. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Manage Engine IT management and monitoring product. (2007). https://www.manageengine.com/

Mclean, Bandar, Oshea, & Crockett. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Trans Knowl. Data Eng*.

Medelyan & Witten. (2008). *Domain-Independent Automatic Keyphrase Indexing with Small Training Sets*. Academic Press.

Mehmeti, X., Mehmeti, B., & Sejdiu, R. (2018). *The equipment maintenance management in manufacturing enterprises*. www.sciencedirect.com

Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. The American Association for Artificial Intelligence.

Mothe & Ramiandrisoa. (2016). *Extraction automatique de termes-clés : Comparaison de méthodes non supervisées.* Academic Press.

Nagios IT management and monitoring product. (2007). https://www.nagios.com/

Paukkeri, Nieminen, Poll, & Honkela. (2008). *A Language-Independent Approach to Keyphrase Extraction and Evaluation*. Academic Press.

Paukkeri, M., & Honkela, T. (2010). Likey: Unsupervised Language-Independent Keyphrase Extraction. *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Romero, M., Moreo, A., Castro, J., & Zurita, J. (2012). Using Wikipedia concepts and frequency in language to extract key terms from support documents. *Expert Systems with Applications*, *39*(18), 13480–13491. doi:10.1016/j.eswa.2012.07.011

Rudrapal, D., Das, S., Debbarm, N., & Ebbarma, S. (2013). Internal Attacker Detection by Analyzing User Keystroke Credential. *Lecture Notes on Software Engineering*, *1*(1), 49–52. doi:10.7763/LNSE.2013.V1.11

Velardi, Missikoff, & Basili. (2001). *Identification of relevant terms to support the construction of Domain Ontologies*. Academic Press.

Web Services Structure. (n.d.). https://www.guru99.com/wsdl-web-services-description-language.html

Zenoss IT Infrastructure management and monitoring tool for hybrid IT environment. (2018). https://www.zenoss.com/

N. Taghezout is a full professor at university of Oran 1 Ahmed BenBella, Algeria. She holds her doctorate thesis in MITT at Paul Sabatier University in France in 2011. She also received another doctorate thesis in Distributed Artificial Intelligence from university of Oran 1 Ahmed BenBella in 2008. She holds a Master degree in Simulation and Computer aided-design. She conducts her research at the LIO laboratory as a chief of the research group in Modeling of enterprise process by using agents and WEB technologies. Since she studied in UPS Toulouse, she became a member of the EWG-DSS (Euro Working Group on Decision Support Systems). She is currently lecturing Collaborative decision making, Enterprise management and Interface human machine design. Her seminars, publications and regular involvement in Conferences, journals and industry projects highlight her main research interests in Artificial Intelligence.