

# A Computer-Facilitated Method for Matching Incident Cases Using Semantic Similarity Measurement

Yong-Bin Kang, Arkady Zaslavsky and Shonali Krishnaswamy  
Caulfield School of IT, Monash University, Australia  
{yongbin.kang, arkady.zaslavsky, shonali.krishnaswamy}  
@infotech.monash.edu.au

Claudio Bartolini  
HP Labs, Palo Alto, USA  
claudio.bartolini@hp.com

**Abstract**—This paper presents a new computer-facilitated approach for incident management to improve typical incident management. Our approach automates typical manual-based incident resolution process by proposing a new semantic similarity measurement between a given incident call and incident cases stored already in a case base. The proposed semantic similarity measurement distinguishes traditional similarity measures by incorporating additional useful information and exploiting semantic knowledge about features appeared in two incident descriptions to be compared. First, we state how typical incident management is processed and what its problems are. We then propose our automated incident resolution process with its core components. After that we introduce our identified additional useful information for our similarity measurement and describe how our similarity measurement algorithm is carried out. In an experimental evaluation, we show the technical coherence and feasibility of the proposed solution using a real dataset.

## I. INTRODUCTION

A primary goal of IT Service Management (ITSM) is to satisfy customers' business requirement by providing IT best practices, based on accumulated knowledge with abundant experience [19]. A core principle of ITSM is to ensure that such practices must meet customers' requirement and expectation appropriately at all times. One of the most important and fundamental processes of ITSM can be regarded as *incident management*. The aim of the incident management is to quickly solve reported incidents that affect the normal running of customers' IT services [7]. An incident can be defined as "any event which is not part of the standard operation of a service and which causes, or may cause, an interruption to, or a reduction in, the quality of that service[19]." This is a pretty large definition that covers two broad types of work [14]: (1) some error or failure of some component in IT systems and (2) requests for new or additional services.

Typical incident management can be processed as follows [12]: Once an incident call is reported by a customer, a level 1 support specialist (L1) describes its incident description and enters it into a system; Then, L1 attempts to solve this incident by identifying the most similar incident stored in the system using his/her knowledge and experience. Otherwise, the incident is escalated to an appropriate L2; Next, if a L2 receives the escalated incident, he/she tries to solve the

incident if he/she can. If the incident is solved, the solution is reported to the customer, and then the incident is closed. If not, the incident is escalated an appropriate L3; This process is iterated to Lk (normally,  $k \in [3..5]$ [3]) until the incident is solved.

An incident description may consist of the following types: (1) customer information who requested the incident call, (2) incident classification information selected as the appropriate type of the incident call by L1, (3) incident support specialist (e.g., L1) information who has the responsibility for resolving the incident call, and (4) problem description information which describes the symptom of the incident call using free-text.

However, the main weakness of the typical incident management lies in its high dependency on SDA's<sup>1</sup> manual work during the incident resolution is performed. This drawback subsequently may cause these two negative effects:

- 1) **Inconsistent way of incident solving:** It may be difficult to guarantee that same or similar repetitive incidents are solved in a consistent way, due to the gaps between knowledge and experience of SDAs. For instance, given an incident call, a novice L1 may produce too many possible solutions that he/she may choose a wrong final solution from them. On the other hand, a more skillful L1 or L2 may produce a more accurate solution. This problem would be more serious in a situation where the positions or the roles of SDAs are frequently changed in IT companies today.
- 2) **Time-consuming:** One important standard for evaluating efficiency of the incident management may be the *speed* of the process to solve a given incident [7]. According to [8], the speed of the response and resolution of the incident was the most frequently mentioned requirement of customers. In fact, for IT service providers to benefit by reducing a total cost for resolving high volume of incident calls, a large amount of incident calls needs to be solved in a short time period. However, in a

<sup>1</sup>In the rest of this paper, to simplify the presentation, we will use term 'SDA' (Service-Desk Agent) to call any level of support specialist.

real situation, the speed for incident resolution process may be slower than customers expect. This is because the incident resolution depends highly on SDA's manual work with complicated incidents assignment procedures, which are commonly recognized for a relatively long period of time.

The primary goal of this paper is to propose a new computer-facilitated approach for incident management to improve the typical incident management. To accomplish this goal, we present a new incident resolution process which automates a process for resolving a given incident call. As the key technique for this automated process, we propose a new semantic similarity measurement. This similarity measurement is performed by considering these two sources: (1) incorporating additional useful information outside of the problem description of a given incident description, and (2) exploiting semantic knowledge about features contained in the incident description.

This paper is organized as follows. In Section 2, the related work is reviewed. Then, we describe how our proposed computer-facilitated incident management is processed compared to the typical incident management in Section 3. In Section 4, the proposed similarity measurement is described in detail. Section 5 shows our experimental evaluation and finally the future work and conclusion close this paper in Section 6.

## II. RELATED WORK

To overcome the drawback of the typical incident resolution, intelligent systems are needed to provide a decision support in resolving a given incident. A practical approach that most of these systems have adopted is Case-Based Reasoning (CBR) [1], [22]. Here, we first summarize the related work in the domain of CBR, and then provide an analysis on another recent approach aimed to automate ITSM incident management workflow.

### A. Similarity Measurement in CBR

CBR is a problem-solving methodology in which optimal solutions for a given incident can be retrieved based on the similar experience that have been learned in the past [1], [22]. In CBR, a *case* contains a unit of decision bases (i.e., problem and solution description) which is stored in a knowledge base (i.e., *case base*).

The main premise under CBR approach is that the more similar two *problem* descriptions, the more similar their *solutions* are. In particular, the heart of the solution retrieval for a given incident can be regarded as *similarity measurement* to retrieve optimal cases which computes the similarity between identified problem features and the features of stored cases [22]. Therefore, in this paper, our main attention is restricted to similarity measurement used in CBR.

The ultimate goal of similarity measurement is to compute a cognitively plausible similarity score between two objects, imitating human common judgments on how similar these two objects are [15]. Given the set of keywords of a given problem description  $I$ , and the set of keywords of a given

case  $C$ , their similarity  $sim(I, C)$  can be computed by the following various ways: (1) The pairwise similarity approach aggregates the similarities between all pairs of keywords from the two sets,  $I$  and  $C$ ; (2) In intersection-based approach, the similarity between  $I$  and  $C$  is computed by their set intersection (e.g., Jaccard or Dice's Coefficient) [21]. That is, the more keywords two sets have in common, the more similar they are considered; (3) The vector-based approach treats the set of keywords as a vector. The vectors are represented by the set of individual vectors of the keywords of the two sets  $I$  and  $C$ , denoted by  $\vec{I}$  and  $\vec{C}$ . Then, the similarity between  $\vec{I}$  and  $\vec{C}$  can be measured by using the *cosine* of the angle of them [5].

However, those traditional similarity approaches in CBR need to be improved, due to the following two reasons:

1) *Reason*: The traditional similarity approaches are mostly performed in the *limited decision space* for similarity measurement in common. That is, the similarities of these approaches are calculated within a decision space, which only consists of single type of information (i.e.,  $I$  and  $C$ ), ignoring a sufficient set of other information available outside of  $I$  and  $C$ .

For example, in Caseadvisor [23], once a problem is presented to the system the system tries to retrieve similar cases for the problem. First, the problem features (keywords) are extracted and the resolution is carried out based on similarity computation between the problem features and the features of case descriptions. However, this similarity is only performed within the given problem description and case description.

To extend the limited decision space, various approaches have been proposed by attempting to utilize various types of other useful information. For example, knowledge management-centric help desk system [6] attempts to use diverse knowledge sources in the organization, including database, files, and experts to ensure high utilization and maintenance of knowledge stored. But, this approach still seems to rely on SDA's capability to handle those knowledge resources. OPAS [2] uses semantic knowledge provided by ontological knowledge representation about problem features, while the system in [4] presents a matching technique to find the closest case for a given problem description using an alias table that contains a list of all abbreviations and non-key words related to the problem description. Even though, these two works try to extend the limited decision space by utilizing more amounts of various forms of knowledge, the knowledge used in those approaches may be viewed as an extension of the same type information about  $I$  and  $C$ .

2) *Reason*: The traditional similarity approaches depend mainly on the notion of *syntactic differences* between two sets (i.e.,  $I$  and  $C$ ) being compared. Typical examples relied on syntactic similarity are founded on common distance metrics such as the Hamming distance for pure binary or symbolic attributes, the Euclidian distance for pure numeric attributes, and the Heterogeneous Euclidian Overlap Metric for mixed representation [21].

However, these would not often be sufficient when comparing non-numeric elements, since the semantic relationships

between different elements cannot be modelled with these types of similarity measures. For instance, consider the similarity between two elements: *database* and *software*. From the viewpoint of syntactic differences, the similarity between these two elements may be 0, because these two elements have no common word. But, these two elements may be to some degree semantically similar since *database* can be seen as one type of *software*.

### B. Automating ITSM incident management approach

So far, very few studies have been done on designing approaches for automating and improving typical manual-based incident resolution process without using CBR. However, based on our recent survey, we found that the aim of the work proposed in [7] is quite similar to our goal in terms of improving the typical manual-based incident management.

The approach in [7] tries to automate the typical incident management by automatically finding the failing components related to identified keywords written in the problem description of a given incident. Finding failing components is carried out by combining the identified keywords and possible configuration items, which are identified in Configuration Management Database (CMDB)<sup>2</sup>. To identify these configuration items, this approach exploits the implicit and explicit relational information between the identified keywords provided by CMDB. Finally, the inferred failing components can be used for correctly routing a given incident to SDAs and its root cause analysis.

Although, this approach attempts to automate the typical incident management by providing additional information about failing components, however, the incident resolution seems to be still relying on SDAs who deal with given incidents. In contrast to this approach, our work tries to automate the typical incident resolution process by proposing a new similarity measurement, which will be used for matching the most similar stored incident to a given incident.

## III. A COMPUTER-FACILITATED INCIDENT MANAGEMENT PROCESS

### A. Manual-based Typical Incident Resolution Process

Fig. 1 shows how the typical manual-based incident management is processed for resolving a given incident in detail[12], [7]:

(1) **“Report incident call”** - Once a customer encounters an error (or failure) that needs to be fixed, this customer reports an incident call for this error using phone, email, web/internet, etc; (2) **“Open incident ticket & Create Incident description”** - Once an incident is reported, a SDA (normally L1) creates a new incident ticket by entering the incident description into the system. At this time, the SDA classifies the appropriate incident type of this incident, which is used to classify the incident in order to provide initial support. Initial

<sup>2</sup>The database that stores the resulting description of the IT infrastructure is called CMDB. That is, CMDB stores configuration items which represent systems, software, and people in IT infrastructure, as well as the relationships of these items.

support means proper analysis, evaluation, if required, routing, not determining root causes of the incident [14]. (3) **“Identify keywords with SDA intelligence”** - The incident resolution process is performed by the SDA. The SDA first tries to identify relevant keywords from the incident description along with human intelligence to guess core components related to the incident; (4) **“Search using keywords from problem description to find similar past incidents”** - Based on these identified keywords, the SDA attempts search procedures to match a set of similar incidents from a knowledge base. If any matched incident is found, its *solution* may be used as the resolution of the incident. Otherwise, the SDA routes the incident to the appropriate SDA (e.g., L2), referring to the incident assignment procedure document; (5) **“Review and Diagnose incident description” / “Generate resolution”** - Once the incident is escalated to the higher level of SDA, the assigned SDA reviews the forwarded incident description to understand the customer problem and tries to diagnose the incident with more specialized skills to solve the incident. If this SDA cannot solve the incident, it is escalated to the next higher level of SDA (e.g., L3). This step is iterated until the incident is resolved; (6) **“Close the incident ticket”** - If the incident is solved, the solution is notified to the customer and the incident ticket is closed. The SDA who solved the incident updates itself with resolution of the incident. The lower level of SDA (e.g., L1) also updates the incident ticket with the appropriate information before it can be closed.

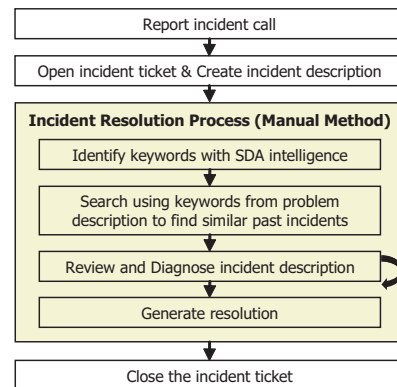


Fig. 1. Manual-based typical incident management process.

### B. Automated Incident Resolution Process

Our proposed computer-facilitated incident management process is shown in Fig. 2. As seen in this figure, we automate “Incident Resolution Process” by accommodating the following components.

1) *SDA Role Categorized*: As described, SDAs provide an effective coordination of activities to solve a given incident via various levels of specialized support. One distinct activity is *creating* an incident description for a given incident call. However, this description may be written in various ways, according to which SDA is dealing with this incident call.

A SDA can be often represented by its *characteristics*, which reflect the SDA’s role or position well in the domain.

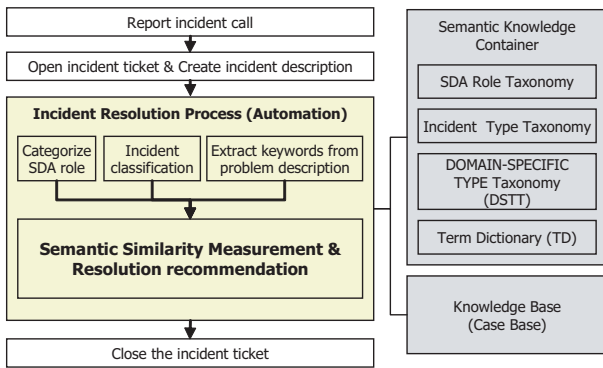


Fig. 2. The proposed computer-facilitated incident management process.

An example of SDA’s characteristics used in this work is shown in Table I. Such characteristics may be differently defined according to specific criteria by the domain experts. Based on these characteristics, we define SDA’s *role*. The aim is to identify a simple form of SDA’s role based on its characteristics, thereby better utilizing it in our “Incident Resolution Process”.

The SDA’s role is generated by using the technique proposed in [10], which is based on fuzzy logic. Every characteristic of SDA is defined by a particular membership function which helps to take the crisp input values and transform them into certain degree of membership values. Then, we assign predefined role categories to SDAs, according to their corresponding membership values using a set of fuzzy rules. Table I also shows a fuzzy set for membership functions of SDA characteristics. A fuzzy rule example used in our approach is: “IF AGE IS *young* AND EDUCATION IS *college* AND WORKGROUP IS *software* OR DOMAIN KNOWLEDGE IS *medium* ... THEN ROLE IS *SD\_R1*.”

TABLE I  
SDA ROLE CHARACTERISTICS AND FUZZY SET.

TYPES	CHARACTERISTICS	FUZZY SET
Demographic	Age	young, middle, old
	Education	school, college, graduate, professional
	Workgroup	hardware, software, application, printer, etc
Knowledge	Domain knowledge	low, medium, high
	Training	low, medium, high
Proficiency	Work experience	novice, medium, senior, expert
	IT Speed	low, medium, high
	Skill	low, medium, high

2) *Tokens/Keywords Extracted from Problem Description:*

The meaningful terms contained in the problem description belonging to a given incident description play significant role in matching similar incidents. This is because these terms are mainly used as main criteria for measuring similarity between a given incident and experienced incidents stored in a knowledge base. A crucial issue raised here is how to capture those terms from the problem description. To extract the terms, we present these two types of knowledge sources:

- *Term Dictionary (TD):* TD maintains common standardized terms and their alias terms, which are used in a given

domain. The structure of TD consists of (‘standardized term’, ‘alias terms’, and ‘term type’). The ‘term type’ can be categorized into ‘DOMAIN-INDEPENDENT TYPE’ and ‘DOMAIN-SPECIFIC TYPE’. Further, the ‘DOMAIN-SPECIFIC TYPE’ is subcategorized into more details, based on their domain-specific features (e.g., internet, database, printer, etc). Since we cannot list all possible terms and their alias terms that can be used to describe problem descriptions, the terms not found in TD are assumed to be categorized as ‘DOMAIN-INDEPENDENT TYPE’.

The premise of using TD is that if problem descriptions are always described using the set of standardized terms, we can avoid the ambiguity caused from the following cases: (1) some heterogeneous terms can be used to describe a same term; (2) alias terms often cannot be interpreted properly or missed; and (3) non-keywords are often recognized terms used in the given domain.

- *DOMAIN-SPECIFIC TYPE Taxonomy (DSTT):* DSTT describes the subcategorized types of the ‘DOMAIN-SPECIFIC TYPE’ and their relationships using a simple ontological description, based on “is-a” relationships. The aim is to improve the possibility of capturing implicit semantic knowledge about those subcategorized types by using this ontological description. A part of the ontological description of DSST used in this paper is seen in Fig. 3(c).

The proposed term extraction process from a given problem description is described as follows:

- 1) Possible terms are extracted by removing stop-words (e.g., punctuation, ‘a’, ‘is’, ‘the’, etc);
- 2) The standardized terms corresponding to the extracted terms and their term types are identified by referring TD. First, we transform the extracted terms, categorized as ‘DOMAIN-INDEPENDENT TYPE’, into their corresponding standardized terms using TD. For example, “*cant, cannot, can’t*” terms are transformed into ‘*unable*’ term as seen in Table II. We refer to all of the terms generated by this process, as *tokens*. Second, we also transform the extracted terms, categorized as ‘DOMAIN-SPECIFIC TYPE’, into their corresponding standardized terms using TD. We call these transformed terms as *keywords* in the rest of this paper. Examples of a TD used in this paper and simple text processing using this TD are presented in Table II.

3) *Incident Classification Defined by SDA:* As in the case of the typical incident management, we assume that categorizing incident classification is done by SDA when creating the incident description for a given incident call. In our “Incident Resolution Process”, we just use the categorized incident classification for computing similarity measurement to match experienced incidents.

4) *Semantic Similarity Measurement:* The key component of the proposed “Incident Resolution Process” is measuring semantic similarity between a given incident call and



TABLE II  
TERM DICTIONARY AND TEXT PROCESSING.

STANDARDIZED TERM	ALIAS TERMS	TERM TYPE
unable	unable, cannot, cant, can't	DOMAIN-INDEPENDENT
print	print, printer, printing problem, printing error	printer
HPPrinter	HP 4250n, HP 4300n, HP 4500n	LaserJet
w3	w3, world wide web, www	internet
sql	sql, query language,	database
Bizflow	N/A	custom application

A problem description: "printer tray cannot print from Bizflow."  
 -tokens: "printer", "tray", "unable(← cannot)", "print", "Bizflow".  
 -keywords: "print(← printer)", "print", "Bizflow"

experienced incidents. After measuring this similarity, the experienced incident having the highest similarity score will be recommended to the customer who reports the incident call. The details of the proposed similarity measurement is presented in the next section.

#### IV. OUR SEMANTIC SIMILARITY MEASUREMENT

As the key technique for the proposed automated incident management process, we propose a new semantic similarity measurement. The central intuition underlying in our similarity measurement is that the quality of similarity measurement can be improved by (1) incorporating additional useful information outside of the problem description of a given incident description, and (2) exploiting as much domain knowledge as possible.

##### A. Incorporated additional information: "Incident Classification" and "SDA Role"

No doubt, we believe that incorporating additional useful information which can have a great influence on improving the traditional similarity measurement may be viewed as a desirable scheme. Though the perfect determination of which information needs to be incorporated is very difficult, we consider a minimal set of these two information types.

1) *Incident Classification*: Incident classification information, categorized by SDA, provides useful support for resolving a given incident call. Some benefits of using incident classification information are helping to estimate services with which incidents are related; to properly route incidents to the correct support group; to speed up diagnoses by collecting the right information; and to improve the efficiency of technical groups. Likewise, we believe that incident classification information may be also used as valuable information to assist to find the most similar incident case for a given incident call in our similarity measurement.

2) *SDA Role*: Determining an appropriate incident classification for a given incident is still one of the most important issues that many IT organizations may encounter[14]. An inappropriate incident classification leads to more downtime and decrease service quality, since diagnosis for a given incident may become complex, bouncing between multiple SDAs. The main cause is due to that incident classification is primarily determined by SDA's characteristics. Thus, when we utilize incident classification information, we may also need

to consider the SDA's characteristics. As discussed in Section III-B, in our approach, since SDA's characteristics are represented by SDA's *role*, we consider SDA's role information.

##### B. Semantic Similarity using Domain Knowledge

Many proposal focusing on similarity measurement have popularly taken the advantages of ontological description scheme to model domain knowledge [9], [18]. One merit is the ability of utilizing *semantic knowledge* of the relationships of objects, being compared, that helps to make more complete similarity measurement. Typically, the most important relationship in ontologies is, namely, "is-a" relationship [17]. An important nature of "is-a" relationship lies in its *hierarchical* structure and its transitivity, logically inferred by navigating the hierarchical relationships between entities specified by the "is-a" ontology<sup>3</sup>.

Intuitively, in a taxonomy, highly related entities are grouped together and the path between two different entities in the hierarchy reflects how these are semantically related in the application domain. As a consequence, using semantic knowledge inherent in taxonomy can provide useful insights of the underlying domain knowledge, thereby facilitating comparative analysis of the entities for similarity measurement. In this paper, we will use the term *semantic similarity measurement*, which is a particular similarity measurement using semantic knowledge (more precisely, taxonomic knowledge) about structural relationships among their entities belonging to a taxonomy.

In order to take advantage of using taxonomic knowledge, we exploit three taxonomies in our similarity measurement (see also Fig. 3): a taxonomy of the set of SDA's roles ( $T_{sr}$ ), a taxonomy of the classified incident types ( $T_{inc}$ ), and DSTT ( $T_{kt}$ ).

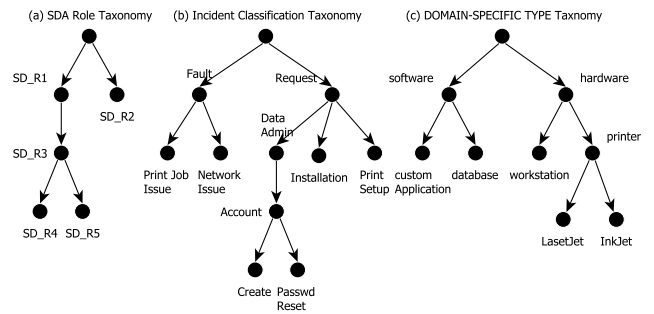


Fig. 3. Three taxonomies used in our similarity measurement

##### C. Object Model in Our Similarity Measurement

To incorporate additional identified information, described in Section IV-A, into our similarity measurement, we first need to define an object model to be compared. In our approach, we use a set (or bag)-based representation to modeling the object, due to its simplicity.

<sup>3</sup>In the remaining of the paper, we call this "is-a" ontology as simply *taxonomy*

In our approach, objects are treated as an ordered set of three types of *attribute values*, which describe different types of distinguishing values or properties of these three attributes: (1) *SD role* attribute ( $A_{sr}$ ), (2) *incident classification* attribute ( $A_{inc}$ ), and (3) *problem description* attribute ( $A_{desc}$ ).

Again, note that unlike the traditional similarity measures, the two sets of attribute values of  $A_{sr}$  and  $A_{inc}$  are additionally described in taxonomies  $T_{sr}$  and  $T_{inc}$ , respectively. Regarding attribute  $A_{desc}$ , we cannot predefine its values in a taxonomy, since these values are described using free text. Instead, the subcategorized types of ‘DOMAIN-SPECIFIC TYPE’ terms, defined in Term Dictionary, are built in taxonomy  $T_{kt}$ .

An example of an incident  $O_1$  is represented as  $O_1 = \{\text{“SD\_R1”, “Print Job Issue”, “Printer is unable to print Bizflow”}\}$ . In the rest of this paper, the object representation is only restricted to be based on this object model.

#### D. Similarity Measurement Modeling

We propose a new similarity measurement that computes semantic similarity between two given objects (i.e., one is newly reported incident object - simply, *incident object*, the other is stored incident object - simply, *case object*) by exploiting semantic knowledge about attribute values belonging to these objects.

The goal of the proposed semantic similarity measure is to compute the similarity  $sim(O_1, O_2)$ , a real number  $[0,1]$ , where  $O_1$  and  $O_2$  are given two objects which are described by three types of attribute values. Given incident object  $O_1 = \{a_{11}, a_{21}, a_{31}\}$  and case object  $O_2 = \{a_{12}, a_{22}, a_{32}\}$ , similarity  $sim(O_1, O_2)$  between  $O_1$  and  $O_2$  is defined to be

$$sim(O_1, O_2) = w_1 * s_{sr}(a_{11}, a_{12}) + w_2 * s_{inc}(a_{21}, a_{22}) + w_3 * s_{desc}(a_{31}, a_{32}) \quad (1)$$

where  $s_{sr}(a_{11}, a_{12})$  denotes the similarity between two attribute values of  $a_{11}, a_{12} \in T_{sr}$  (the set of SDA roles),  $s_{inc}(a_{21}, a_{22})$  means the similarity between two attribute values of  $a_{21}, a_{22} \in T_{inc}$  (the set of classified incident types), and  $s_{desc}(a_{31}, a_{32})$  represents the similarity between two attribute values of  $a_{31}, a_{32}$ , where  $a_{31}$  is the problem description of  $O_1$ , and the problem description of  $O_2$ . In order to express the different importance of these three similarities  $s_{sr}$ ,  $s_{inc}$ , and  $s_{desc}$ , we define weights on them, denoted by  $w_1$ ,  $w_2$ , and  $w_3$ , respectively, where  $w_1 + w_2 + w_3 = 1.0$ .

Now, our main concern is how to define similarity measures  $s_{sr}$ ,  $s_{inc}$ , and  $s_{desc}$ , exploiting semantic knowledge of their relationships described in the taxonomies  $T_{sr}$ ,  $T_{inc}$  and  $T_{kt}$ , respectively. Computing these similarities is divided into the following two methods.

1) *Semantic Similarity Measure for  $s_{sr}$  and  $s_{inc}$* : The semantic similarity measure for  $s_{sr}$  and  $s_{inc}$  is divided into two approaches.

The first approach is *distance-based* approach that estimates semantic similarity between two attribute values in a given taxonomy by calculating distance between them [9]. For example, some representative work is found in [11], [24].

The second approach is *node-based* approach which estimate the amount of sharing *information content* between two attribute values [18]. In this approach, given a taxonomy, a node represents a unique attribute value consisting of a certain amount of information, and an edge represents a relation between two attribute values. The similarity between two attribute values is then the extent to which they *share* information in common. Namely, the more information they have, the more similar they are. Based on the notation in information theory, Resnik [18] quantifies the information content of attribute value  $a$  by considering the negative log likelihood, as follows:

$$ic_{tc}(a) = -\log p(a), \quad (2)$$

where  $p(a)$  is the probability of encountering  $a$  in a given corpus. Intuitively, as  $p(a)$  increases,  $a$  becomes more abstract and the informativeness of the concept  $a$  decreases. In [18],  $p(a)$  is defined as using maximum likelihood estimation,  $p(a) = \frac{freq(a)}{N}$ , where  $freq(a)$  is the sum of the frequency counts of its all subsumed attribute values occurred in the corpus, and  $N$  is the total number of attribute values in the taxonomy.

Based on eq(2), there have been several similarity measures proposed. Given two attribute values  $a_1$  and  $a_2$ , some representative measures are

$$\begin{aligned} (a) \quad sim_{Res}(a_1, a_2) &= ic_{tc}(lcs(a_1, a_2)) \\ (b) \quad sim_{in}(a_1, a_2) &= \frac{2 * ic_{tc}(lcs(a_1, a_2))}{ic_{tc}(a_1) + ic_{tc}(a_2)} \\ (c) \quad sim_{jcn}(a_1, a_2) &= \frac{1}{ic_{tc}(a_1) + ic_{tc}(a_2) - 2 * ic_{tc}(lcs(a_1, a_2))} \end{aligned} \quad (3)$$

where (a), (b), and (c) are proposed by Resnik [18], Lin [13], and Jiang [9], respectively, and  $lcs(a_1, a_2)$  is the *least common subsumer (LCS)* that subsumes  $a_1$  and  $a_2$  in the taxonomy.

One important issue with regard to the node-based approach is how to measure the information content of each attribute value in a taxonomy. Resnik’s method [18] relies on the combined knowledge of the hierarchical structure from the taxonomy with statistics on actual usage of the attribute values from a given corpus. As an alternative approach, Seco [20] exploits only the structural information of the taxonomy. The premise underlined in this approach is that a taxonomic structure is organized in a meaningful and principled way, where more general attribute values convey less information than attribute values lower down, not relying on corpora analysis. Given taxonomy  $T$  and attribute value  $a$ , this measure  $ic_t$  is formally defined as,

$$ic_t(a) = \frac{\log\left(\frac{|sc(a)|+1}{|T|}\right)}{\log\left(\frac{1}{|T|}\right)} = 1 - \frac{\log(|sc(a)|+1)}{\log(|T|)}, \quad (4)$$

where  $sc(a)$  the set of subsumed attribute values of  $a$  and  $|T|$  is the total number of attribute values that exist in taxonomy  $T$ .

For our similarity measurement for  $s_{sr}$  and  $s_{inc}$ , we use eq(5), which is a combination of eq(3c) and eq(4), given by [20]. The main reason is that several studies [9], [16], [20] commonly found that eq(3c) is the most effective and outperforms either of those two pure approaches. Moreover, since we do not need to consider a corpus, we choose eq(4)

for measuring the information content of the attribute value.

$$\begin{aligned} sim(a_1, a_2) = \\ 1 - \left( \frac{ic_t(a_1) + ic_t(a_2) - 2 * ic_t(lcs(a_1, a_2))}{2} \right), \text{ where} \quad (5) \\ ic_t(a_k) = 1 - \frac{\log(|sc(a_k)| + 1)}{\log(|T|)} (k = 1, 2), \end{aligned}$$

where  $lcs(a_1, a_2)$  is the *LCS* that subsumes  $a_1$  and  $a_2$  in taxonomy  $T$  and  $|sc(a_k)|$  the cardinality of the set of subsumed attributes of  $a_k$ .

To illustrate, let us consider the taxonomies shown in Fig.3 with two objects  $O_1 = \{\text{"SD\_R1", "Print Job Issue", "Printer is unable to print Bizflow"}\}$  and  $O_2 = \{\text{"SD\_R5", "Fault", "Bizflow printing error"}\}$ . Given  $O_1$  and  $O_2$ ,  $s_{sr}(\text{"SD\_R1", "SD\_R5"})=0.613$  and  $s_{inc}(\text{"Print Job Issue", "Fault"})=0.770$ .

2) *Semantic Similarity Measure for s\_desc*: Given incident object  $O_1 = \{a_{11}, a_{21}, a_{31}\}$  and case object  $O_2 = \{a_{12}, a_{22}, a_{32}\}$ , similarity  $s_{desc}(a_{31}, a_{32})$  can be computed by these two substeps:

**Substep1: Intersection-based similarity between two token sets** - After text processing using the proposed method in Section III-B, we have two token sets and keyword sets of  $O_1$  and  $O_2$ . Then, as a basic similarity measure, we first compute the syntactical differences between two token sets, denoted by  $T_{a_{31}}$  and  $T_{a_{32}}$ , of  $O_1$  and  $O_2$  using their set intersection. That is, the more terms  $T_{a_{31}}$  and  $T_{a_{32}}$  have in common, the more similar they are considered. One of the well-known measures for this similarity is Jaccard coefficient [21]. Given two token sets  $T_{a_{31}}$  and  $T_{a_{32}}$ , the Jaccard coefficient is represented by

$$sim_{jc}(T_{a_{31}}, T_{a_{32}}) = \frac{|T_{a_{31}} \cap T_{a_{32}}|}{|T_{a_{31}} \cup T_{a_{32}}|}. \quad (6)$$

This measure is normalized between interval [0,1] and reach 0 when  $T_{a_{31}}$  and  $T_{a_{32}}$  have no terms in common, and reach 1 when  $T_{a_{31}}$  and  $T_{a_{32}}$  consist of the same set of terms.

To illustrate, consider again these two object  $O_1 = \{\text{"SD\_R1", "Print Job Issue", "Printer is unable to print Bizflow"}\}$  and  $O_2 = \{\text{"SD\_R5", "Fault", "Bizflow printing error"}\}$ . Given  $O_1$  and  $O_2$ , their two token sets are obtained as  $T_{a_{31}} = \{\text{"Printer", "unable", "print", "Bizflow"}\}$  and  $T_{a_{32}} = \{\text{"Bizflow", "printing", "error"}\}$ . Thus,  $s_{desc}(T_{a_{31}}, T_{a_{32}}) = \frac{1}{6} = 0.167$ .

**Substep2: Semantic similarity between keywords** - In this step, given  $a_{31}$  and  $a_{32}$ , we compute a similarity based on their keyword sets, denoted by  $K_{a_{31}}$  and  $K_{a_{32}}$ , exploiting semantic knowledge resided in keyword type taxonomy  $T_{kt}$ . This similarity is based on the premise that the more the types of given keywords are similar, the more similar these keywords are semantically.

For this similarity, we extend the typical intersection-based similarity measure by combining node-based similarity approach. One common limitation of the typical intersection-based similarity measures is that if  $K_{a_{31}} \cap K_{a_{32}} = \emptyset$ , then

their similarity is zero. However, this may be unreasonable in a situation where two keywords have terms that are siblings in a given keyword type taxonomy. In the situation, these two sets should have nonzero similarity even though they have no identical keywords in common.

The inclusion of semantic knowledge inherent in a keyword type taxonomy can allow the intersection-based similarity to avoid such inaccurate zero similarity measure. The key idea of the node-based approach is that the similarity of two entities belonging to a taxonomy is defined by the information content of their *LCS*. By incorporating this concept, we first create two sets of keyword types, denoted as  $KT_{a_{31}}$  and  $KT_{a_{32}}$ , corresponding to two keyword sets  $K_{a_{31}}$  and  $K_{a_{32}}$ , respectively, by referring Term Dictionary. Then, based on the concept of *LCS* inherent in taxonomy  $T_{kt}$ , we augment  $KT_{a_{31}} = \{k_{11}, \dots, k_{n1}\}$  and  $KT_{a_{32}} = \{k_{12}, \dots, k_{m2}\}$  by including the set of the *LCS* that subsumes every pair of  $(k_{i1}, k_{j2})$ , where  $k_{i1}(1 \leq i \leq n) \in KT_{a_{31}}$  and  $k_{j2}(1 \leq j \leq m) \in KT_{a_{32}}$ . Formally, given  $KT_{a_{31}}$  and  $KT_{a_{32}}$ , their corresponding *augmented* keyword types, which denote  $KT_{a_{31}}^+$  and  $KT_{a_{32}}^+$ , can be defined to be

$$\begin{aligned} KT_{a_{31}}^+ &= KT_{a_{31}} \cup \{lcs(k_{i1}, k_{j2})\}, \\ KT_{a_{32}}^+ &= KT_{a_{32}} \cup \{lcs(k_{i1}, k_{j2})\}, \end{aligned} \quad (7)$$

where  $lcs(k_{i1}, k_{j2})$  is the *LCS* of two keyword types  $k_{i1}$  and  $k_{j2}$ . Using this augmented intersection, the *augmented* Jaccard's similarity, can be defined to be

$$sim_{jc^T}(KT_{a_{31}}, KT_{a_{32}}) = \frac{|KT_{a_{31}}^+ \cap KT_{a_{32}}^+|}{|KT_{a_{31}}^+ \cup KT_{a_{32}}^+|}. \quad (8)$$

Based on the above measure, we then assign *weight* to every keyword type in  $KT_{a_{31}}^+$  and  $KT_{a_{32}}^+$ . One promising way to estimate the weight of the keyword type is to use the value of information content of itself. Let  $w(k)$  be the weight of keyword type  $k$ , we then define  $w(k)$  to be  $ic_t(k)$  using eq(4). Finally, the *weighted augmented* Jaccard coefficient, which denotes  $sim_{w(dc^T)}$  is defined by

$$sim_{w(jc^T)}(KT_{a_{31}}, KT_{a_{32}}) = \frac{\sum_{\{i|k_i \in KT_{a_{31}}^+ \cap KT_{a_{32}}^+\}} w(k_i)}{\sum_{\{j|k_j \in KT_{a_{31}}^+ \cup KT_{a_{32}}^+\}} w(k_j)}. \quad (9)$$

To sum up, given two descriptions  $a_{31}$  and  $a_{32}$ , the similarity  $s_{desc}(a_{31}, a_{32})$  can be computed by combining eq(6) and eq(9)

$$\begin{aligned} s_{desc}(a_{31}, a_{32}) = \\ cc * sim_{jc}(T_{a_{31}}, T_{a_{31}}) + (1 - cc) * sim_{w(jc^T)}(KT_{a_{31}}, KT_{a_{32}}) \end{aligned} \quad (10)$$

where  $cc$  is a combination coefficient which represents an acceptable reliability degree of those two similarities.

As the final step of the proposed "Incident Resolution Process", based on the proposed similarity measurement proposed in eq(1), the stored incidents having the highest similarity score is recommend to the customer who reports a incident call.

## V. EXPERIMENT

In this section, we present an experiment to evaluate the *accuracy* of our new approach for automated incident resolution

process. In particular, in order to show that our approach yields more accurate similarity result between an incident object and a stored case object, we have measured four different combinations of similarity measures proposed in this paper. Then, we perform comparative analysis based on the results of those measures to show how our proposed similarity measure (the following **M4**) outperforms the other three measures.

Formally, given incident object  $O_1 = \{a_{11}, a_{21}, a_{31}\}$  and case object  $O_2 = \{a_{12}, a_{22}, a_{32}\}$ , the four similarity measures between  $O_1$  and  $O_2$  are represented by

$$\begin{aligned}
 \mathbf{M1} : sim_3(O_1, O_2) &= s_{desc}(a_{31}, a_{32}), \\
 \mathbf{M2} : sim_{23}(O_1, O_2) &= w_2 * s_{inc}(a_{21}, a_{22}) + w_3 * s_{desc}(a_{31}, a_{32}), \\
 \mathbf{M3} : sim_{13}(O_1, O_2) &= w_1 * s_{sr}(a_{11}, a_{12}) + w_3 * s_{desc}(a_{31}, a_{32}), \\
 \mathbf{M4} : sim_{123}(O_1, O_2) &= w_1 * s_{sr}(a_{11}, a_{12}) + w_2 * s_{inc}(a_{21}, a_{22}) + \\
 &w_3 * s_{desc}(a_{31}, a_{32})
 \end{aligned} \tag{11}$$

where  $s_{desc}$  is computed by eq(10),  $s_{sr}$  and  $s_{inc}$  are computed by eq(5). The detailed explanation of the notations used in the above measures is found in Section IV-D.

#### A. Experimental Configuration

We configure our experimental environment, mainly composed of three components, as follows: (1) Incident case base, (2) Three taxonomies (i.e., SDA Role taxonomy, Incident classification taxonomy, and DOMAIN-SPECIFIC TYPE taxonomy (DSTT)), and (3) Term Dictionary (TD).

1) *Incident case base*: As incident case base, we have chosen 9 incident cases that are extracted from a real life incident management dataset (4350 incident cases) collected through an installation of HP ServiceManager (formerly HP ServiceCenter). Among these 9 incident cases, 7 incident cases are related to ‘printing problem’ and 2 incident cases are associated with ‘application problem’, but the problem descriptions of all of these 9 incident cases have a few keywords in common. Since this dataset was collected from the typical incident management process, that is, solutions are generated by SDA’s manual work, there is no information about SDA’s role in the dataset.

To run our approach, we need to create this role information in the dataset. Thus, we made use of SDA’s *workgroup* information contained in the dataset which indicates a team or group that SDA belongs to. Since this workgroup is represented by numeric values (e.g., 1, 2, ...), we created SDA’s role by simply merging the prefix term, “SD\_R”, with the workgroup value. For instance, given a workgroup “1”, its corresponding role is expressed by “SD\_R1”. The premise of using this substitution is that even though the workgroup may not fully represent SDA’s role quite enough, it would reflect SDA’s role to some extent. Fig. 4 shows the 9 cases used in our experiment.

2) *Three taxonomies*: The three types of taxonomies were constructed as follows:

- Regarding SDA role taxonomy ( $T_{sr}$ ), we used the simple one, shown in Fig. 3(a), which was made by roughly

ID	SDA Role	Incident Classification	Problem Description	Solution Description
1	SD_R5	Fault	Bizflow running slow again	Link in Orpington has been upgraded and an extra server has been installed into the Bizflow farm.
2	SD_R3	Network Issue	Bizflow running very slowly	The master ticket is resolved. Checked with Barbara. There is a new server being set up in Orpington to solve this issue. 20 users so far have access to it as of 2nd October. She said that the users on the new server have seen a big improvement in the per
3	SD_R1	Print Setup	Break-Fix- user is unable to print from Bizflow	Confirmed with Lucy printing OK now. - Tony
4	SD_R3	Fault	Bizflow printing error	Have assisted the user and her colleagues with printing issue and tested all working fine. Think the spooler service need to be restarted on the citrix server
5	SD_R2	Print Job Issue	unable to print from bizflow to the printer uktr7940	Richard Walker : no problem found
6	SD_R5	Fault	Bizflow printing issue	Restarted print spooler on Citrix42
7	SD_R3	Fault	User cannot find printer	User can print but can't select the tray. User will not give me login details for security reason. Can Jason please check what exactly is going on. User is trying to print from different Trays. Printer is OK. Printer don't recognise the Tray she selecte
8	SD_R2	Print Job Issue	User cannot print from any printer	Lucy confirmed all printing OK now. TW
9	SD_R5	Print Job Issue	Printing problem	This was caused by network and server errors yesterday. These services have been restored.

Fig. 4. The 9 cases used in the experiment.

observing the relationships of the SDA roles (more precisely, SDA workgroups) in the randomly selected 100 cases out of the given 4350 dataset. Granted, in the future, this taxonomy will be built with the help of the domain expert;

- Incident classification taxonomy ( $T_{inc}$ ) was constructed on the basis of the given 4350 dataset, consisting of 51 types. However, we merely used the taxonomy which is a part that taxonomy seen in Fig. 3(b), since this taxonomy covered all incident types being appeared in the 9 incident cases used in the experiment;
- In fact, it is generally recognized that keywords used in a certain application domain rely highly on the characteristic of the domain. Thus, since the complete set of definition for the taxonomy of subcategorized types of ‘DOMAIN-SPECIFIC TYPE’ terms may be impossible, we defined a taxonomy using a minimal set of the keyword types used in the given 4350 dataset. For this experiment, a small part of the defined taxonomy ( $T_{kt}$ ) was used which is shown in Fig. 3(c).

3) *Term Dictionary*: To construct Term Dictionary, we observed terminologies frequently used in the areas of ITSM and Information Technology (IT) by referring some websites.<sup>4</sup>

#### B. Experimental Scenario and Results

As an input incident call, we decided to use an actual incident case extracted from the given 4350 incident cases,

<sup>4</sup>(1) Glossary for ITSM: [http://www.covestic.com/downloads/glossary\\_itmsm\\_terms.pdf](http://www.covestic.com/downloads/glossary_itmsm_terms.pdf), <http://www.itsmf.ca/about/glossary.html>, <http://www.e-help-desk.com/glossary.html>; (2) Glossary for IT: [http://www.gartner.com/6\\_help/glossary/GlossaryMain.jsp](http://www.gartner.com/6_help/glossary/GlossaryMain.jsp), <http://www.cryer.co.uk/glossary>, <http://www.itcom.itd.umich.edu/glossary.html>



as seen in Fig. 5.

SDA Role	Incident Classification	Problem Description	Solution
SD_R5	Network Issue	Unable to Print from Bizflow	Restarted Print Spooler on CITRIX53.

Fig. 5. The selected input incident call.

Let  $I$  be the object corresponding to the input incident call, and  $CC = \{C_1, C_2, \dots, C_9\}$  be the collection of the 9 incident cases used in the experiment, where  $C_k(1 \leq k \leq 9)$  is the  $k^{th}$  object, corresponding to the incident case having ‘ID’= $k$ . To measure similarity between  $I$  and  $C_k(1 \leq k \leq 9)$ , we first need to extract the terms of every object  $C_k \in CC$ . Those extracted tokens and keywords are listed as follows:

ID	tokens	keywords and keyword types	
		keywords	keyword types
$O_{in}$	unable, bizflow, print	bizflow: print:	custom application printer
$O_1$	bizflow, slow, running	bizflow:	custom application
$O_2$	slowly, bizflow, running	bizflow:	custom application
$O_3$	unable, break-fix, bizflow, user, print	bizflow: print:	custom application printer
$O_4$	error, bizflow, printing	bizflow: print:	custom application printer
$O_5$	unable, bizflow, uktr7940 printer, print	bizflow: print:	custom application printer
$O_6$	issue, bizflow, printing	bizflow: print:	custom application printer
$O_7$	unable, user, find, printer	print:	printer
$O_8$	unable, user, printer, print	print:	printer
$O_9$	problem, printing	print:	printer

Fig. 6. Extracted tokens/keywords/keyword types.

Given  $I$  and  $C_k(1 \leq k \leq 9)$  being compared, let us denote  $I$  as  $O_1 = \{a_{11}, a_{21}, a_{31}\}$  and  $C_k$  as  $O_2 = \{a_{12}, a_{22}, a_{32}\}$  to help understand of our equations provided in this paper. In this representation,  $a_{11} \in T_{sr}$  and  $a_{12} \in T_{sr}$  represent the attributes values of the SDA roles of  $O_1$  and  $O_2$ , respectively, and  $a_{21} \in T_{inc}$  and  $a_{22} \in T_{inc}$  denote the attribute values of the classified incident types of  $O_1$  and  $O_2$ , respectively. Again let  $T_{a31}$  and  $T_{a32}$  be two token sets extracted from the attribute values  $a_{31}$  and  $a_{32}$ , which are problem descriptions of  $O_1$  and  $O_2$ , respectively. Further, we denote again  $KT_{a31}$  and  $KT_{a32}$  as the corresponding to two keyword sets, extracted from the attribute values  $a_{31}$  and  $a_{32}$ . Then, the results of the four different similarity measures we focused in the experiment are shown in Table III which ranked 5-top incident cases having the highest similarity scores.

One criterion for analyzing the accuracy of these results, shown in Table III, is to compare the solution descriptions of  $O_1$  and  $O_2$  being compared. Since incident object  $I$  (or  $O_1$ ) used in the experiment was selected from the given 4350 dataset, comparing two solution descriptions between  $I$  and  $C$  (or  $O_2$ ) are useful for determining the accuracy of these results. Referring to Fig. 5, we can easily understand that the solution description of  $I$  is about “restarting printer spooler”. The incident cases having the most similar solution descriptions with  $I$  in  $CC$  may be regarded as  $C_4$  and  $C_6$ . Therefore, the accuracy can be determined by looking at how incident cases

$C_4$  and  $C_6$  are highly ranked in each of the four similarities used in the experiment.

As seen in Table III, first, **M1** ranked  $C_3$  and  $C_5$  as the top-ranked cases, since these cases have highest similarity scores, that is,  $sim(I, C_3) = sim(I, C_5) = 0.840$ , while  $sim(I, C_4) = sim(I, C_6) = 0.680$ . To measure **M1**, we assigned  $cc$  (combination coefficient for combining  $sim_{jc}(T_{a31}, T_{a32})$  and  $sim_w(jc^T)(KT_{a31}, KT_{a32})$ ) as 0.4. This means we put the more weight (0.6) on the similarity between keywords than the similarity between tokens (0.4). This  $cc$  value was consistently used in all the measures (**M1** - **M4**).

Second, to measure **M2**, we applied  $w_2 = 0.3$  and  $w_3 = 0.7$  to make the combination  $w_2 * sim_{inc}(a_{21}, a_{22}) + w_3 * sim_{desc}(a_{31}, a_{32})$ . This means we viewed the similarity  $sim_{desc}$  much higher than the similarity  $sim_{inc}$ . This measure ranked  $C_5$  and  $C_4$  as the first and second top ranked incident cases, respectively, while  $C_3$  (top-ranked case in **M1**) ranked fifth case because the  $sim(“Network Issue”, “Print Setup”) = 0$ .

For **M3**, we used the weights  $w_1 = 0.3$  and  $w_3 = 0.7$  to combine  $w_1 * sim_{sr}(a_{11}, a_{12})$  and  $w_3 * sim_{desc}(a_{31}, a_{32})$  as the case of **M2**. In this measure,  $C_6$  and  $C_3$  are ranked as the first and second incident cases, while  $C_5$  has the fifth ranking score, since the similarity  $sim_{sr}(“SD_R5”, “SD_R2”) = 0$ .

To compute the measure **M4** (our approach), we used the weight  $w_1 = 0.2$ ,  $w_2 = 0.3$ , and  $w_3 = 0.5$  to make the combination  $w_1 * sim_{sr}(a_{11}, a_{12}) + w_2 * sim_{inc}(a_{21}, a_{22}) + w_3 * sim_{desc}(a_{31}, a_{32})$ . This measure ranked  $C_4$  and  $C_6$  as the top-ranked two incident cases in which are the most similar solution descriptions are contained. In our experiment, we changed the weights  $w_1$ ,  $w_2$ , and  $w_3$  in various ways, such as, ( $w_1 = 0.25$ ,  $w_2 = 0.25$ , and  $w_3 = 0.5$ ) and ( $w_1 = 0.2$ ,  $w_2 = 0.2$ , and  $w_3 = 0.6$ ). But, the result of ranked cases maintained in the same with the result with **M4**. From these results, we found that our approach (**M4**) is more accurate than the other three measures.

## VI. FUTURE WORK AND CONCLUSION

In this paper, we presented a new computer-facilitated incident management approach to improve the manual-based typical incident resolution process. Unlike the typical incident management, our approach automated conventional incident resolution process by presenting a new semantic similarity measurement between a given incident call and incident cases stored in a case base. The distinctive feature of our similarity measurement is to extend decision space for similarity measurement by incorporating additional useful information outside of problem description of incident cases. As the additional information, service-desk agent role concept was employed using fuzzy logic and classified incident type information was utilized in our similarity measurement. Based on the extended decision space, we proposed a new semantic similarity measurement by combining typical intersection-based similarity measurement and node-based semantic similarity measurement. The experimental results show how our proposed similarity measurement performed well based on our experimental scenario.

TABLE III  
THE EXPERIMENTAL RESULTS.

SIM	Ranked Case ID	$sim_{jc}(T_{a31}, T_{a32})$	$sim_w(jc^I)(KT_{a31}, KT_{a32})$	$s_{desc}(a_{31}, a_{32})$	$s_{inc}(a_{21}, a_{22})$	$s_{sr}(a_{11}, a_{12})$	Similarity Results
$sim_3(O_1, O_2)$ (M1)	Rank1:3	0.6	1.0	0.840	N/A	N/A	0.840
	Rank1:5	0.6	1.0	0.840	N/A	N/A	0.840
	Rank3:4	0.2	1.0	0.680	N/A	N/A	0.680
	Rank3:6	0.2	1.0	0.680	N/A	N/A	0.680
	Rank5:1	0.2	0.667	0.480	N/A	N/A	0.480
$sim_{23}(O_1, O_2)$ (M2)	Rank1:5	0.6	1.0	0.840	0.541	N/A	0.751
	Rank2:4	0.2	1.0	0.680	0.771	N/A	0.707
	Rank2:6	0.2	1.0	0.680	0.771	N/A	0.707
	Rank3:2	0.2	0.667	0.480	1.0	N/A	0.636
	Rank4:3	0.6	1.0	0.840	0.0	N/A	0.588
$sim_{13}(O_1, O_2)$ (M3)	Rank1:6	0.2	1.0	0.680	N/A	1.0	0.776
	Rank2:3	0.6	1.0	0.840	N/A	0.613	0.772
	Rank3:4	0.2	1.0	0.680	N/A	0.693	0.684
	Rank4:1	0.2	0.667	0.480	N/A	1.0	0.636
	Rank5:5	0.6	1.0	0.840	N/A	0.0	0.588
$sim_{123}(O_1, O_2)$ (our approach) (M4)	Rank1:6	0.2	1.0	0.680	0.771	1.0	0.771
	Rank2:4	0.2	1.0	0.680	0.771	0.693	0.706
	Rank3:1	0.2	0.667	0.480	0.771	1.0	0.683
	Rank4:2	0.2	0.667	0.480	1.0	0.693	0.663
	Rank5:3	0.6	1.0	0.84	0.0	0.613	0.573

In the future, we plan to evaluate our approach on the whole set of 4350 incident cases, showing how well our approach could match ideal incident cases, which are agreed with human intuition. We will also evaluate how our approach could provide better results as to finding ideal incident cases by comparing other relevant similarity measures.

#### REFERENCES

- [1] A. Aamodt and E. Plaza, "Case-based reasoning : Foundational issues, methodological variations, and system approaches," *AI Communications*, vol. 7, no. 1, pp. 39 – 59, 1994.
- [2] O. Alm, E. Hyvonen, and A. Vehvilainen, "Opas : An ontology-based library help desk service," *4th European Semantic Web Conference 2007 (ESWC 2007)*, 2007.
- [3] C. Bartolini, C. Stefanelli, and M. Tortonesi, "Symian: A simulation tool for the optimization of the it incident management process," *Managing Large-Scale Service Deployment, LNCS*, pp. 83–94, 2008.
- [4] K. H. Chang, P. Raman, W. H. Carlisle, and J. H. Cross, "A self-improving helpdesk service system using case-based reasoning techniques," *Computers in Industry*, vol. 30, no. 2, pp. 113–125, 1996.
- [5] P. Ganesan, H. Garcia-Molina, and J. Widom, "Exploiting hierarchical domain structure to compute similarity," *ACM Transactions on Information Systems*, vol. 21, no. 1, pp. 64–93, 2003.
- [6] L. M. Gonzalez, R. E. Giachetti, and G. Ramirez, "Knowledge management-centric help desk: specification and performance evaluation," *Decision Support System*, vol. 40, no. 2, pp. 389–405, 2005.
- [7] R. Gupta, K. Prasad, and M. Mohania, "Automating itsm incident management process," *Autonomic Computing, 2008. ICAC '08. International Conference on*, pp. 141–150, June 2008.
- [8] R. Heckman and A. Guskey, "Sources of customer satisfaction and dissatisfaction with information technology help desks," *Journal of Market-Focused Management*, vol. 3, no. 1, pp. 59–89, 1998.
- [9] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *In Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, 1997.
- [10] Y. Kang, A. Zaslavsky, and S. Krishnaswamy, "Help-desk agent recommendation system based on three-layered user profile," *European Conference on Artificial Intelligence (ECAI) 2008 Workshop on Recommender Systems*, 2008.
- [11] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," *In: Fellbaum, MIT Press*, pp. 265–183, 1998.
- [12] R. Leopoldi, "It service management - incident/problem management methods and service desk implementation best practices," *White Paper of RL Infomation Consulting LLC.*, 2003.
- [13] D. Lin, "An information-theoretic definition of similarity," *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 296–304, 1998.
- [14] H. Marquis, "How to classify incident," *itSM Solutions, DITY Newsletter*, 2006.
- [15] D. L. Media, R. L. oddstoac, and D. Gentncr, "Respects for similarity," *Psychological review*, vol. 100, no. 2, pp. 254–278, 1993.
- [16] T. Pedersen, S. V. S. Pakhomov, S. Patwardhan, and C. G. Chute, "Measures of semantic similarity and relatedness in the biomedical domain," *J. of Biomedical Informatics*, vol. 40, no. 3, pp. 288–299, 2007.
- [17] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *IEEE Trans.Syst.Man Cybern*, vol. 19, no. 1, pp. 17–30, 1989.
- [18] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research.*, vol. 11, pp. 95–130, 1999.
- [19] C. Rudd, "An introductory overview of itil," *The IT Service Management Forum (itSMF)*, 2004.
- [20] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in wordnet," *Techn. report, University College Dublin, Ireland*, 2004.
- [21] C. J. van Rijsbergen, in *Information Retrieval. 2nd ed.*, 1979.
- [22] I. Watson, "Case-based reasoning is a methodology not a technology," *Knowledge-Based Systems*, vol. 12, no. 5-6, pp. 303 – 308, 1999.
- [23] Q. Yang, E. Kim, and K. Racine, "Caseadvisor: Supporting interactive problem solving and case base maintenance for help desk applications," *In Proceedings of the IJCAI 97 Workshop on Practical Applications of CBR*, 1997.
- [24] M. P. Zhibiao Wu, "Verb semantics and lexical selection," in *Proceeding of the 32nd Annual Meeting of the Association for Computational Linguistics*, 1994, pp. 133–138.