KNOWLEDGE DISCOVERY AND MANAGEMENT WITHIN SERVICE CENTERS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Nazia Zaman

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

April 2016

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Knowledge Discovery and Management within Service Centers

By

Nazia Zaman

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Juan Li
Chair

Dr. Jun Kong

Dr. Kenneth Magel

Dr. Ying Huang

Approved:

04/15/2016	Dr. Brian Slator
Date	Department Chair

ABSTRACT

These days, most enterprise service centers deploy Knowledge Discovery and Management (KDM) systems to address the challenge of timely delivery of a resourceful service request resolution while efficiently utilizing the huge amount of data. These KDM systems facilitate prompt response to the critical service requests and if possible then try to prevent the service requests getting triggered in the first place. Nevertheless, in most cases, information required for a request resolution is dispersed and suppressed under the mountain of irrelevant information over the Internet in unstructured and heterogeneous formats. These heterogeneous data sources and formats complicate the access to reusable knowledge and increase the response time required to reach a resolution. Moreover, the state-of-the art methods neither support effective integration of domain knowledge with the KDM systems nor promote the assimilation of reusable knowledge or Intellectual Capital (IC). With the goal of providing an improved service request resolution within the shortest possible time, this research proposes an IC Management System. The proposed tool efficiently utilizes domain knowledge in the form of semantic web technology to extract the most valuable information from those raw unstructured data and uses that knowledge to formulate service resolution model as a combination of efficient data search, classification, clustering, and recommendation methods. Our proposed solution also handles the technology categorization of a service request which is very crucial in the request resolution process. The system has been extensively evaluated with several experiments and has been used in a real enterprise customer service center.

ACKNOWLEDGEMENTS

I am grateful to acknowledge and thank all those who assisted me in my graduate program at North Dakota State University. I would like to express my deepest appreciation and a bundle of thanks to my academic advisor Dr. Juan Li for her continuous guidance and support throughout my PhD study and research. I truly appreciate her great encouragement and effort in my research years. Without her careful supervision and persistent support this dissertation would not have been possible.

Special thanks to my graduate committee members, Dr. Jun Kong, Dr. Kenneth Magel, and Dr. Ying Huang for their valuable improvement suggestions. I would also like to thank Dr. Ammar Rayes, Distinguished Engineer, Cisco Systems Inc., for giving me the opportunity to apply my knowledge and gain experience on real-life data and problems.

Last but not the least, I would like to thank my family members for their unconditional support and encouragement throughout my years as a graduate student. They always have believed in me and supported me to pursue my dreams. This dissertation is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

These days, knowledge discovery and management (KDM) has become an important organizational tool for enterprise customer service centers. KDM systems allow organizations to manage their knowledge capitals effectively and efficiently. Specifically, for enterprise customer service centers, knowledge management system plays a very critical role. The growing business of service centers and IT consulting services is an indispensable part in customer-organization communications. These service centers generally receive an enormous number of product and technology service requests from their customers and partners on a daily basis. In a service center like this, customer support engineers offer assistance to the customers by addressing and solving their requests. To ensure the highest level of service, providing an accurate and appropriate solution in a timely manner for a service request is very vital for these service centers. The solution for a service request can either be delivered in the form of a pre-generated Intellectual Capital (IC) or can be used to pervade the problem request with knowledge on how to address that problem. Apart from serving customer requests, this information can also be used to form an automated processing of future service requests having similar issues, thereby avoiding reinvention of request resolution all together. Moreover, knowledge can be used to handle scenarios including proactive bug or problem fixing, preventing issues with similar devices, and software images in advance within customer networks. In other words, knowledge management may facilitate to avoid service requests altogether.

Researches have shown that the significant relationship between IC and the value it adds to the customer service productivity is gaining importance day by day (Phusavat 2013). Enterprise customer service centers should put more effort on exploiting the competitive advantages of Intellectual Capitals with the aim to increase their operating efficiency (Lu 2014).

The following subsections briefly represent the research challenges, the limitations of existing systems in solving those challenges, contributions of this research to address those limitations, and a short outline of this dissertation.

1.1. Research Challenges

The common practice of storing organizational knowledge assets includes various data sources like knowledge repositories, enterprise websites, white papers, and social networks of Subject Matter Experts (SMEs). In addition, the data stored in these diverse repositories may have heterogeneous and unstructured formats such as text files, command line interface output, and Web pages. These diverse data sources and huge volume of heterogeneous data formats not only complicate the request resolution procedure but also cause extensive processing time. As a matter of fact, researches have shown that 25% to 50% of the time spent by the technical support teams is for searching the solutions (Feldman 2004).

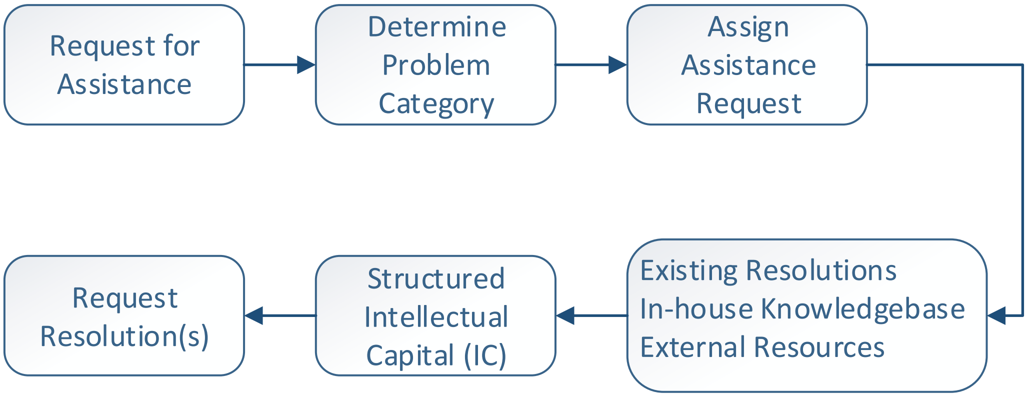


Figure 1. Request Resolution Process in an Enterprise Service Center

On the other hand, consulting services are expected to serve customers with the expertise to build, improve and scale their IT environment. Consulting engineers often execute repetitive tasks like performing network assessments and optimizations manually due to the lack of reusable knowledge or intellectual capital necessary to complete a task. So, the knowledge captured from customer engagements must be a part of the workflow of technical support teams to make effective use of it. Nevertheless, solutions to new service requests increase both the prospect of capturing reusable knowledge and the challenge of scaling the search and distribution of newly captured information.

In enterprise customer service centers, customer support engineers receive an enormous number of service queries from customers and partners regarding enterprise products and technologies. As depicted in Figure 1, upon receiving a customer's query requesting assistance a Service Request (SR) is created. After the SR is stored in enterprise service request management database, it assigned to a customer service engineer. While working on a service request, the engineer may quickly give suggestions based on his experiences if the request is a known one. However, for new problems, before organizing the solutions, he may need to do an extensive search related to the issue from different sources- such as the Internet / social network sites, related service requests, enterprise's white papers and/or technical reports. A service request system is then used to document the information traded between the customer and the service center. Furthermore, informal methods like manuals, binders, sticky notes, case histories, etc. are also used by the engineers to research inquiries and provide solutions (Rasooli 2007). But, these conventional methods are not optimal as they do not promote automation, knowledge sharing, and knowledge reuse or expedient resolutions.

1.2. State-of-the-art Methods

These days, to improve the request resolution efficiency there is an increasing demand of knowledgebase solutions among the customer support service centers. Knowledge management systems also help these service centers to reduce their need for in-house and business support escalations (Ashu 2012). In most cases, it is very likely that a service request being asked to a service center, has been asked before, and in all probability will be requested again. Hence, most service centers try to capture both the resolution, and the procedure followed in solving a previously posed request and build structured knowledge, which is called Intellectual Capital (IC), from this experience (Rasooli 2007), and (Heitz 2008). Therefore, after receiving a service request enterprise knowledge management systems should be capable of matching the service requests with similar cases that have been resolved before. The Intellectual Capitals are contributed by expert support engineers based upon their concrete experience and can be structured in the form of a knowledge database. These ICs promote faster access to enterprise knowledge repository and efficient service request response.

Although KM systems, built upon service engineers' previous experiences, facilitate efficient responses to customer inquiries and resolutions of similar requests, but such systems have limitations-

i) These systems suffer from cold start problem. In other words, new service requests cannot benefit from such system.

ii) Up-to-date information from sources like social network discussion threads or e-mail exchanges between service engineers and customers cannot be quickly integrated to the system.

iii) The existing systems lack in utilizing domain knowledge in granular level of service request resolution.

iv) The state-of-the-art methods do not consider incorporating multi-word lexicons and noise or information in their KM systems.

Research study has shown that with an overwhelming number of service requests, service center staffs endeavor to balance between the complexity of requests, existing tools or skills and quality service expectations by customers (Dimension Data 2013-2014). Due to these inevitable challenges, the request resolution rate is decreasing for consecutive years and as a result, the ratio of customers requesting for services and get their issues resolved has also dropped to 75%.

1.3. Contributions of the Dissertation

To address the aforementioned challenges, we propose an efficient knowledge management system to transform the huge amount of data into reusable knowledge or Intellectual Capital (IC). The IC can be utilized by service management solutions to make service processes more efficient, effective, and predictable. In particular, our effective IC mining system collects, processes and analyzes data from heterogeneous sources like enterprise data repositories and social network sites to extract and store ICs into machine-interpretable format to facilitate inference and reuse. This model offers better categorization of service request resolution data by integrating rich semantics, advanced search with data mining and machine learning technologies. The goal of this work is to make fundamental contributions towards realizing a usable, intelligent, and effective framework for IC mining.

Our goal is to help service engineers and customers find the right information they require and present the information in an understandable and reusable format. But before a service engineer can start working on a service request resolution, the very first step is to identify the problem category and if possible, the subcategory to make the service request assignment process smooth and accurate. To achieve this goal we build a technology-sub technology identifier which precisely outputs the problem category of a service request. In the next phase of service request resolution, we implement an integrated custom search engine by utilizing Google Custom Search. The search engine locates the documents related to service request, but with massive amount of noise. To remove noise and shape the extracted information into a powerful representation, we have utilized a multi-level classifier. At the third and the most important step of this process, we implement a semantics-guided classifier to categorize the information returned by the search engine to a structured Intellectual Capital which can be easily understood and absorbed by service center engineers or customers. Once the IC is extraction is complete, it is stored in an IC repository which clusters together similar ICs for making the knowledge reuse efficient and easy. Later, when a new service request comes in, the support engineers use the recommender module in our IC management system to search for already resolved similar cases before attempting the request resolution process.

We use ontology to incorporate domain knowledge to unstructured information for precise and intelligent KM. Ontology (Fensel 2001) provides a shared and common understanding of a domain that can be communicated across people and application systems, and thus facilitate interoperability, knowledge sharing and reuse. With the assistance of this semantic web technology our mining tool can automatically infer relationships between important concepts thus enabling accurate knowledge extraction and organization. There will also be a module for continuous learning to optimize the results where engineers' captured knowledge will be used as a feedback in areas such as query refinement, problem definition, results evaluation and verification to improve the results.

1.4. Dissertation Outline

The dissertation is organized as follows. Chapter 2 presents the related works in knowledge discovery and management with background knowledge. Chapter 3 gives an overview of the system architecture with the details of the methodologies. In Chapter 4 we evaluate the proposed methods and presented the effectiveness of the IC management system with a comprehensive set of experiments. Chapter 5 presents and concluding remarks with future research directions.

2. BACKGROUND AND RELATED WORK

The first section in this chapter briefly outlines the state-of-the-art methodologies in knowledge discovery and management (KDM) that have been investigated throughout this dissertation. Following the related works, the background knowledge related to KDM and our methodology, which is described in Chapter 3, has also been presented to better understand the problem domain.

2.1. Related Work

Knowledge is considered as a critical asset for organizations to leverage competitive advantage in today's economy. To stay competitive, organizations constantly seek for different data sources that can be used for new knowledge creation. However, business success is significantly dependent on organization's capabilities to acquire, manage, develop and use knowledge dynamically (Alavi 2001), (McEvily 2000), (Ravichandran 1999), (Sambamurthy 2005), and (Wu 2008). In other words, enterprise's capabilities are considered to be interrelated with its knowledge and the ability to manage it. In fact, knowledge management is a combination of knowledge assets and knowledge processes which are foundation of development, maintenance and renewing of enterprise proficiencies (Adler 1989), (Prahalad 2006), (Marr 2001), (Leonard-Barton 1995), and (Nelson 2009). In recent years, organizations' attempts to capture and reuse their intellectual assets have accelerated the use of knowledge management systems (KM). Knowledge management system has particularly abetted the customer support managers by reducing their support cost through the automation of complex support problems.

As defined by Nada K. Kakabadse et al., knowledge represents the organized and meaningful information which is accumulated through experience, communication and reasoning (Kakabadse 2003). Knowledge is acquired from raw facts or observations and classified or analyzed data utilizing human expertise and experience (Varun Grover 2001), (Kakabadse 2003), and (I. Nonaka 1994). Organizational knowledge and intangible assets determining an organization's value and competitiveness, often refer to the concept of Intellectual Capital (IC) (Magrassi 2002). Knowledge and innovation are considered as two very valuable intangible resources because of their key roles in organization's long-term business competitiveness. Proper management of organization's IC can reinforce the continuing productivity measurement efforts on an organization's intangible assets (Phusavat 2013). Researches have shown that operating efficiency of an organization is heavily dependent on its intellectual capital (Lu 2014). Grant (Grant 1996) and Spender (Spender 1996) have presented a knowledge-based view of organization where intangible resources have been categorized into different types of knowledge.

Enterprise customer support is one major area which requires management of data and information in a large context. Data like customer & business partners' contact information, products that require support, actions taken to resolve customer's request for a service, time spent on the service request resolution process—all these need to be captured and analyzed properly. Nevertheless, the knowledge of tackling and cracking complex service problems was not easily managed until lately. The enormous amount of knowledge available in electronic format requires a considerable amount of time & resources to search through. One solution to this problem is employing more experts as they can solve many customer problems. But, for organizations it is not always possible to appoint an adequate number of experts to keep up with the growing product knowledge and customers needing help. Another problem of expert dependent system is that when they leave the organization, part of the knowledge gets lost as the knowledge transfer process is not always easy. Moreover, the invasion of new, complex, and highly technical products comes with the requirement of high-end customer support as they may have very little knowledge about the products and their underlying technology. Enterprise customer service centers can have several thousand employees over the world and it is very unlikely that who takes service requests about problem and issues are all experts. These requirements incur huge costs both in infrastructures and in people. As a solution of these problems, enterprise support centers have started using knowledge management system to make the captured support knowledge available for their employees serving the customers. These customer support centers have already achieved quantifiable amount of benefits through the deployment of support knowledge management system. These knowledge management systems have enabled the enterprise customer support in reducing the call times, resolving problems without the involvement of in-person visit to customer's place. The ability to distribute, reuse, and apply the captured knowledge through the knowledge management systems has been beneficial for the organizations in reducing the dependency of expert & expensive support workforces.

Though the identification and management of an organization's intellectual capital is gaining importance but mining IC is inherently challenging and is often imprecise. The major challenge is that in most cases the knowledge related to IC is embodied in unstructured or semi-structured formats. The traditional approach of mining organization's IC is carried out manually through labor and time demanding tools like interviews, assessments, workshops, etc. (Yin 2003). Moreover, the quality of the results is greatly reliant on the experience & expertise of the persons involved in the process. The quality of the manually captured ICs also vary in quality due to the unavailability of uniform architecture for knowledge creation, acquisition and elicitation.

One key challenge in managing customer support knowledge is deciding whether to structure the information in advance prior to use or to formulate the structure in real time. In general, knowledge management involves various technologies which mostly involve relatively unstructured online repositories of enterprise product or service related documents (Davenport 1998). For instance, almost all customer service centers provide the facility to search for solutions to non-time-critical problems (Salton 1986). However, in most cases, this keyword based repository search can be quite time intensive and often results many documents that do not fulfill user's exact needs. On the other hand, the requirement is more vigorous for customer support environment requiring support analyst's assistance as customer's time is considered important. So engineers in customer support need instantaneous and accurate resolutions to solve customer queries in real time. In addition, after finding an appropriate document, the support engineer needs to read through the document to interpret it in the context of the customer's service request. Moreover, many of the customer support engineers may be apprentices at surfing through enterprise knowledge bases to address customer support problems. This process of request resolution process may be suitable to off-line problem-solution research, but it is inefficient & unacceptable for the real-time requirements.

As mentioned by Rasooli et al., the process of knowledge management consist of knowledge acquisition, knowledge creation, knowledge distribution, knowledge adaptation and knowledge utilization (Rasooli 2007). Ikujiro Nonaka defined knowledge creation not only as a process of making obtainable and intensifying knowledge created by individuals but also as a method to connect it to an enterprise knowledge system (I. G. Nonaka 2006). An organization's success in knowledge creation is closely linked to its proficiency to extract, convert, and combine implicit and explicit knowledge from various sources. Conversely, the amount of available electronic data within organization in increasing dramatically and most of these are either unstructured or semi-structured (Waters 2005). Unstructured data like e-mails, white papers, and other text-based documents contain important information like expert knowledge, details of customer relationships, common problem fixes, etc. Researchers have shown that the knowledge inherent in unstructured information can constitutes up to 80–98% of all the organization knowledge are very valuable for organization intellectual capital management (Cheung 2005). Researchers are working to provide more structured knowledge content which are capable in rapid and precise knowledge retrieval with more detailed solution descriptions. One of the most popular method for identifying intellectual capital-related information is content analysis (J. R. Guthrie 2004). The method presented by the authors is manual and requires codification of the qualitative and quantified IC-related information into some pre-defined IC indicator categories. These categories were compiled based on the literatures on government policy and professional policy announcements (J. R. Guthrie 1999).

However, manual method of IC extraction critically restricts the volume of texts that can be processed due to the labor demanding data assemblage process (Beattie 2007), and (Abeysekera 2006). The data extracted in this manual process is affected by personal bias even after the researchers' participation in assessment. Another issue is- the risk of inconsistency increases due to the different coding rules followed by different coders for interpretation. These inevitable disadvantages of manual IC creation inspired researchers to model the IC extraction problem to a machine solvable one. Nick Bontis utilized electronic search and a list of IC terminology to be used as the IC revelation references to capture the IC-related information in the electronic database of Canadian Corporations annual reports which contains approximately 11,000 records (Bontis 2003). Though this automatic method makes the identification of huge amount intellectual capitals conceivable but the low level of IC disclosure exhibits the fact that this kind of electronic search was not an efficient way of IC-related information extraction. The system could not recognize the synonyms and words with multiple meanings related to an IC (Beattie 2007). Furthermore, the number of matched IC related information was reduced as the machine-operated system cannot comprehend the background knowledge of the keywords.

Oliverira et al. investigated the level of IC disclosure in Spain using Concordance, a software program which aids in the study and analysis of textual data by improving the reliability, replicability, and objectivity of the extracted data. However, the performance of the system was not encouraging as the number of recognized ICs were less as compared to the manual identification. Lee and Guthrie exploited Factiva to identify the knowledge related to intellectual capitals in the business and analyst reports (Lock Lee 2010). Factiva (Dow Jones 2016) is an intelligent classification tool which support automatic organization of the IC related information and provides full-text access to current and archived news and business information. The authors applied electronic search to extract the IC-related data which were corrected using human assessments. After the correcting the gross errors, accepted IC terms were manually mapped to the Factiva intelligent taxonomy terms. Weng et al. presented their method of knowledge extraction and reuse using a text analytics system which includes hierarchical classifier and a recommender (C. R. Wang 2011) (C. R. Wang 2010). In their system, the authors formulate the classifier to label service center requests to well-defined categories, explicitly what, why, and how. The recommender module in the system recommends previously solved solutions for similar requests. A knowledge discovery framework utilizing a service oriented architecture (SOA) was proposed by *Klieber et al.* (Klieber 2009). The algorithm consists of three different steps- concept vectorization, concept clustering, and mapping finding and was designed with the primary goal to ease usage for non-knowledge discovery experts. *Suganya et al.* proposed a two-level model demonstration of textual data representation- syntactic and semantic (Suganya. S 2013). The syntactic information was presented using a tf-idf model and for semantic data representation, the authors have used Wikipedia. The authors employed three support vector machine, nearest neighbor classifiers on three different levels- syntactic, semantic, and the combined result of the two previous levels correspondingly.

Nevertheless, all of these existing classification approaches work on the single lexicon level and do not address these challenges efficiently-

i) How to deal with the dynamic social media data?

ii) How to provide support for never-seen service requests?

iii) How to properly incorporate the enterprise domain knowledge in service request resolution?

iv) How to effectively consider multi-word semantic entity, noise, and error information?

Our proposed IC Mining system effectively solved these problems utilizing a MaxEnt classifier. The output of the IC extractor (classifier) is then clustered together to create a reusable & machine understandable IC repository.

2.2. Background Knowledge

This sub-section provides preliminary terminologies related to the work and the literature survey of recent works on knowledge discovery and management.

Knowledge discovery (KD) is the process of generating knowledge from data using tools like artificial intelligence, mathematics, and statistics. According to Gregory Piatetsky & William Frawley, knowledge discovery is a method for extracting implicit, previously unknown, and potentially useful information from data (Piateski 1991). The process of discovering knowledge can be designed to exploit the underlying features and structures of various application domains- card analysis, customer analysis, and product analysis & enquiries. Table I represents some of the document management techniques, developed to reduce the workload of huge data handling.

Table 1. Document Handling Techniques

Function	**Method**	**Data Representation**	**Output**
Document Searching	Keyword extraction, Information Retrieval	Keywords, character strings	A set of documents
Document Organization	Keyword distribution analysis, classification, clustering	Set of keywords, features	Clusters of documents
Knowledge discovery	Semantic analysis, NLP, data mining	Semantic concepts	Concept

Management of data and information is an integral part of customer support. It is very crucial for the organizations to manage knowledge assets with intelligence in order to provide better service to the customers. However, for technology-intensive industries, it has always been a painstaking task to manage the knowledge of approaching & solving complex service problems. Searching through organization's technical white papers, internal databases, intranet & Internet for the knowledge required to solve the client service request has always been a time-consuming task. On the other hand, employing a good number of domain experts may reduce the request response time reasonably. But, for most of the technology-intensive call and service centers it is not always cost-effective to hire enough experts to match the increasing growth of both product knowledge and service requests. Moreover, losing even a single expert staff significantly impacts both the customer-organization relationships and company finance as the best practice knowledge seems to get lost when that expert leaves the organization (Nolan Norton 1998). Another challenge for cost effective business operations is to share & collaborate the knowledge among the stakeholders in order to establish the best practices (Kuehnast J 2009).

To handle these problems, organizations have started to capture, distribute and reuse support knowledge by deploying knowledge management systems (KMS). This reusable knowledge includes already solved service requests & their solutions, product enquires, answers and recommendations.

Table 2. Intermediate Forms and Corresponding Text Units

Intermediate Form (IF)	**Text Unit**
Bag of Words (BoW) N-grams	Word Token
Multi-term text phrase Paragraph	Paragraph
Concept Graph Semantic Graph Concept Hierarchy	Concept
Document	Document

In other words, knowledge management is the process of recognizing, capturing, and utilizing organizations' collective knowledge not only to help them compete (Von Krogh 1998) but also to increase the receptiveness and innovativeness at the same time (Hackbarth 1998). The activities involved in the process of knowledge management of call centers and service centers include knowledge acquisition, generation, distribution, adaptation and utilization (Rasooli 2007). The authors have used a case study approach to propose an abstract high-level knowledge management model for call centers based on the aforementioned actions. Knowledge in KMS is represented both in human and machine readable forms. The human-readable format of knowledge can be accessed using tools like search engines, browsers, etc. On the other hand, machine-understandable knowledge forms the knowledge base to aid in decision making of intelligent systems. These intelligent expert systems can be an essential part of KM systems. To formulate the machine readable knowledge base, it is very important to choose the form of knowledge representation. Ontology is one such form of representation where data conceptualization is explicitly specified (Guber 1993). In organizational knowledge management systems, ontology specification represents product or service taxonomy to define the knowledge for the system. Ontologies serves as a common resource in the KM system which facilitates knowledge search, storage, representation and reuse.

However, the quality of the knowledge varies depending on several factors including the source that has been used to produce that knowledge. For example, on discussion forums, users tend to rely more on the best practice solutions posted by the domain experts as compared to the ones suggested by the general users. So it is very crucial to filtering out the noisy and irrelevant data before knowledge extraction takes place. We will discuss the state of the art methodologies that are being used for knowledge extraction in the following subsections.

2.2.1. Text Mining

The process of automatic extraction of new, previously unknown information, from heterogeneous data sources is known as Text mining (Senellart 2008). Text is the most common form of information storage for most of the organizations. In fact, there is a rule of thumb, cited by Merrill Lynch (Shilakes 1998), which says that around 80% of potentially usable information may contained in unstructured textual documents, primarily text. As a multidisciplinary technique for knowledge discovery from unstructured text, text mining comprises of information retrieval, natural language processing, information classification, clustering, and visualization which provides computational intelligence (Sorensen 2009).

Text mining has been considered a variation of data mining (Navathe 2000) where the later aims to find interesting patterns from a large datasets. However, the fuzzy and unstructured nature of natural language text makes text mining more complex with a higher commercial potential as compared to data mining (Kano 2009). The aim of text mining is to finding out interesting and useful patterns in natural language text. To achieve the vision of text mining as data mining on raw unstructured data, text mining procedures aim to obtain structured datasets called *intermediate forms* (IF) to make data accessible for knowledge extraction techniques. Table 2 represents the intermediate forms and their corresponding atomic text units used by our IC mining system to extract the reusable knowledge from the raw data. Among these four IF, we briefly discuss BoW and N-gram techniques in the later subsections.

2.2.2. Natural Language Processing

One important step in machine understandable data preparation is to remove noisy information from the unstructured text/document used in knowledge capital extraction. To prepare the unstructured data to be used for knowledge extraction, Natural Language Processing (NLP) techniques are applied. In other words, knowledge discovery is achieved through NLP.

The origin of NLP was in the 1950s as the intersection of linguistics and artificial intelligence (AI). At the beginning, research areas of NLP and text information retrieval (IR) were diverged from each other. As stated by Manning et al. (Manning 2008), focus of IR is to index and search high volumes of textual data efficiently by deploying highly scalable statistics-based techniques. However, with time both these area have converged and NLP borrows from several other diverse fields as well. The following subsection describes the phases and the roles of natural language processing that has been used in our IC mining system to retrieve useful & reusable knowledge capital.

2.2.3. Named Entity Recognition

Named entity recognition (NER) is an essential component in applications like information extraction (Khalid 2008), machine translation (Babych 2008), question answering (Toral 2005), etc. NER aims to find all textual mentions of named entities (persons, organizations, locations, quantities, etc.) in the document. Named entities also include numerical expressions like dates, time, and percent. Moreover, named entities are domain dependent. In the domain of service centers, a named entity (NE) can be a product name or service name or a technology, etc. NER can either be gazetteer based, or trained.

2.2.3.1. Gazetteer-based NER

Gazetteer based approaches for named entity recognition systems utilize peripheral knowledge base to match text phrases via some dynamically constructed gazette to the names and entities. One useful feature of this approach is that gazetteers also provide a non-local model for resolving multiple names into the same entity. Gazetteer approaches performs better in certain domains (Torisawa 2007), (Richman 2008), and (Ritter 2011).

2.2.3.2. Trained NER

Trained NER performs better across the domains while performing predictive analysis for entities unknown to a gazette. These named entity recognition systems use statistical models to make predictions about named entities in document. However, systems like these require a huge amount of annotated training data to be effective in entity recognition. Also, trained NER don’t naturally provide a non-local model for entity resolution (Funayama 2009), (Florian 2003), (Chieu 2002), and (McCallum 2003).

Results achieved by the Statistical NERs are comparable to hand-coded systems. For instance, IdentiFinder (Bikel 1999), which is based on Hidden Markov Model (HMM) has achieved remarkably good performance. Apart from these two categories, NER can also be rule based. The entity extraction module in our IC mining system uses a hybrid approach of entity recognition using both n-gram (gazetteer) and rules.

Considering the underlying semantics of the natural language text is vital during the phase of preprocessing to convert raw unstructured text into machine understandable format. The entity extractor module in IC Mining system employs semantic web technology to retrieve this contextual information and to extract the machine understandable and reusable knowledge from the raw, unstructured natural language text. The following subsection briefly presents the role of semantic web technology, more specifically the role of ontology in knowledge discovery and management.

2.2.4. Ontology in Knowledge Management

This section outlines a short overview of the distinguished NLP model named Ontology Web Language (OWL) (McGuinness 2004) and Ontology in general. Sharing extracted knowledge among different platforms/applications remains a challenging task. Ontology, a domain specific hierarchical and conceptual representation of information, provides a way for knowledge sharing and reuse through logical interpretation of textual data (Hsieh 2011), (Marinica 2010), and (K. B. Ahmed 2014). According to Gruber, ontology is an explicit formal specification of a shared conceptualization (Wong 2012). The notion of explicit denotes that ontology should define the types of constraints used in model, formal means that it should be comprehensible to machines, shared specifies that it should be shared by group rather than being restricted to individuals. Ontology improves the interoperability by providing a shared and machine-executable meaning on concepts across users and systems. In addition, ontology supports inference mechanisms that can be used to enhance semantic matchmaking. For knowledge engineering and knowledge representation, ontologies provide a number of useful features (Gomez-Perez 2006).

Studies related to semantic web, especially ontologies, have advanced from the specific needs related to knowledge management within a computational environment, in particular from the challenge of knowledge sharing and reuse (Wong 2012). Ontologies facilitate domain specific knowledge sharing and reuse through semantics and hierarchical relationships between concepts and objects (Lin 1999), (Wimalasuriya 2010), (Vogrinčič 2011), (Marinica 2010), (Serra 2014), (Turney 2010), and (Glimm 2012). For instance, ontologies in the medical domain comprise concepts related to treatments of various diseases and clinical procedures that simplify the propagation of standard terminologies in the healthcare systems. Some of the most popular ontologies include Unified Medical Language system (UMLS), Basic Formal Ontology (BFO), Protein Ontology (PO), Suggested Upper Merged Ontology (SUMO), and Bio Investigation Ontology (BIO) (Turney 2010), (X.-Y. J.-H. Liu 2009), and (Domingos 1997). Flexible annotations and hierarchical conceptualization made ontologies instrumental in other application areas like semantic search, entity recognition and text mining (J. D.-P. Fernández 2010), (Subhashini 2011), and (K. B. Ahmed 2014).

OWL, an XML based vocabulary, provides a comprehensive ontology representation using class definitions, relationships between classes and constraint based class properties or attributes. It is an extension of Resource Description Framework (RDF) which supports the subject-predicate-object model to make assertions about a resource. Many ontology-based knowledge management systems have been designed & proposed by the researchers in recent years to facilitate effective knowledge sharing and reusing as ontology can efficiently handle heterogeneous data sources. A. Hotho et al. proposed a semantic method of document clustering (Hotho 2002) where the authors have used ontology to identify distinct interests. The authors had incorporated background knowledge to improve the clustering results where the system can make selection between results. Fernández et al. proposed a semantic search model which addresses the challenges of the massive and heterogeneous Web environment by utilizing ontologies (A. S. Fernández 2008). In conjunction of ontology, the authors have used a Fuzzy Rule Classification System to handle disproportionate data sets. Similarly, G. Rong et al., presented an ontology-based information retrieval system which retrieves and manages non-metallic pipe knowledge of oilfield (Rong 2012). Phoenix, an information management system, was proposed by *A.* Uszok et al. (Uszok 2013) which uses global ontology to manage knowledge in coalition environment domain and from documents of different format. An ontology-based KM approach which integrates a data quality component was presented by Sangodiah, Anbuselvan, and Lim Ean Heng for e-learning systems (Sangodiah 2012). However, the system does not provide a method to recognize the interaction of various technologies used within the component. Fan, Jing, et al. describes an ontology-based method for forest knowledge management (Fan 2012). But, during the system design, the authors did not incorporate the dynamic factors like resource competition and mutual benefits between species which influence the species configuration. As surveyed in (J. D. Huang 2010), ontology has also been extensively used for knowledge discovery and sharing in bioinformatics and medical informatics.

Several methodologies have been proposed for document annotation and classification based on features like pre-defined categories, domain knowledge or ontology. Labrou, Yannis, and Tim Finin presented a method which uses Yahoo!-Categories as a concept hierarchy and annotates Webpages using an n-gram classifier (Labrou 1999). The system has been designed to solve applications in the area of text mining. The authors used a notion of similarity matrices to test the new document topics and then at the final classification results were given based on a predefined threshold value. De Luca, Ernesto William, Andreas Nürnberger, and O. von-

Guericke have presented a system to classify search results into semantic classes by utilizing linguistic ontologies (specifically MultiWord-net (Pianta 2002)) (Luca 2004). For classification, the authors simply computed the cosine similarity between the search results and the multi-senses returned from the MultiWord-net ontology of the keyword. The system is not independent as the disambiguation has been after classification of the information was done. As a result, the systems does not benefit from the different semantics consideration. In another similar approach, Cheng et al. modeled knowledge management as a document clustering problem by utilizing Ontology-based Semantic Classifications framework (Cheng 2004). The two key components utilized by this framework are: context free text interpreter- for syntactic analysis & context-based categorization agent-for context model usability enhancement. But, the authors did not provide any implementation details to support the proposed framework. In (Bawakid 2010), the authors expand words in a document with the synonyms defined in WordNet claiming this would improve the classification accuracy for semantic categorization of text. However, the issue of expanding words with multiple semantic meanings was not addressed by them. Our IC mining system overcomes all these shortcomings of the existing systems by addressing the inherent domain knowledge with the help of semantic web technologies.

2.2.5. Classification

Data classification can be defined as the task of identifying the category for each of the given data based on a training data set whose category membership is known. Single-labeled classification is one of the two different categories of classification where each data belong to exactly one category and the categories do not overlap. One simple version of single-labeled classification is *Binary Classification* where each data point is assigned to one of the two predefined categories. Classification methods like Naïve Bayes and Support Vector Machine

(SVM) have been developed to address the single-labelled classification problem. The second category is multi-labelled classifiers where a single data point can belong to multiple categories concurrently. Multi-labelled classification problems are very common in the area of information retrieval. The following subsections represent the different types of classifiers used to classify textual data.

2.2.5.1. Maximum Entropy Classifier

The main idea behind this probabilistic classifier is that unknown model generating the sample data should be the model that is most uniform and satisfy all constrains from training data (A. Ratnaparkhi 1996). In other words, training data is used to constrain the conditional distribution in maximum entropy where each of the constraints defines a characteristic of the training. The model is represented by the following:

$$\mathrm{P(c|d)}=\frac{1}{\mathrm{Z(d)}}\exp(\sum_{i}\lambda_i \; f_i(d,c))$$

Here, $f_i(d, c)$ defines a feature as a real-valued function of the document d and the class c, and λ is a weight vector. The weight vector is found by numerical optimization of the lambdas so as to maximize the conditional probability which can be estimated using different methodologies. For example, some of the iterative calculation methods include Generalized Iterative Scaling (GIS) (Darroch 1970), Improved Iterative Scaling (IIS) (Berger 2005), and LBFGS Algorithm (Malouf 2002). We use the Stanford Classifier to perform MaxEnt classification. To train the weight in our system, we have used conjugate gradient ascent and added smoothing (L2 regularization) (Ng 2004). Z(d) is the normalization function which is computed as:

$$Z(d) = \sum_{c}\exp(\sum_{i}\lambda_i \; f_i(d,c))$$

The first step in using maximum entropy is to identify and select a set of feature functions to be used for classification. Expected value for each feature over the training data is then measured to be used as a constraint for the model distribution. Some examples of maximum entropy classifier include sentence boundary detection (Mikheev 2000), sentiment analysis (Pang 2008), and ambiguity resolution (A. Ratnaparkhi 1998).

2.2.5.2. Support Vector Machine

Support Vector Machine (SVM), a supervised method of classification, was proposed by Vapnik (Vapnik 2013) to solve two-class problems. SVM, which is very popular for text classifications, aims to measure a separation line between two hyperplanes defined by classes of data as shown in Figure 1. This goal of finding the margin separating two datasets rather than focusing on feature matching enables SVM to operate on fairly large feature set. In contrast to other classification methods, SVM algorithm uses both negative and positive training datasets to construct a hyper plane that separates the positive and negative data. The document that is closest to decision surface is called support vector (Baumer 2010).

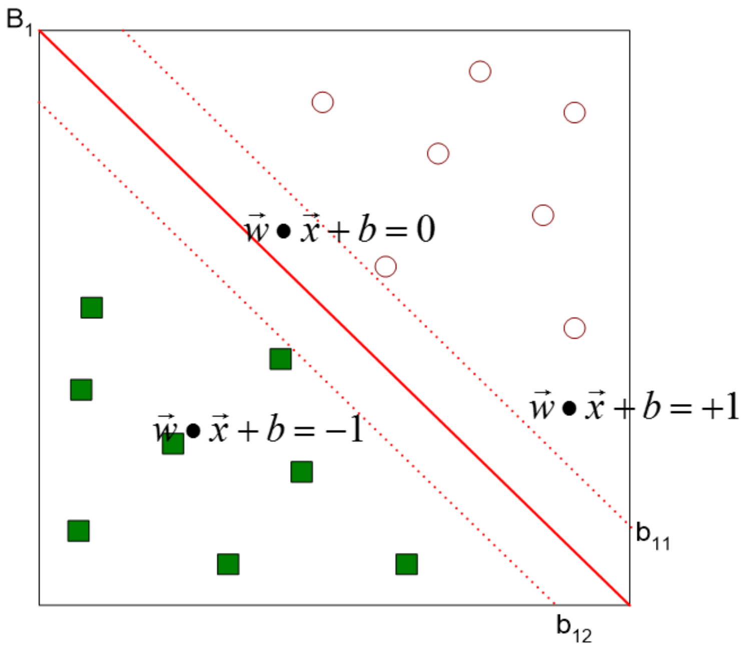

B1
$\vec{w} \bullet \vec{x} + b = 0$
$\vec{w} \bullet \vec{x} + b = -1$
$\vec{w} \bullet \vec{x} + b = +1$
b11
b12

Figure 2. Support Vector Machine

2.2.5.3. Naïve Bayes Classifier

Naïve Bayesian (NB) classifier is a probabilistic learning method for text classification (Narayanan 2013). NB classifier assigns the most likely class to a given document by utilizing feature vector and the assumption that features are independent. The probability of a document *d* being in class *C* is computed using the conditional probability *P(*t_k*|C)* for term t_k in a document d of class *C*-

$$P(C|d) = P(C) \prod_{1 \le k \le n_d} P(t_k|C)$$

Where *P(C)* is the prior probability of the document of being in class C. Bayesian classifier has proven success in text classification applications (Domingos 1997).

Ting Min et al. described a Support Vector Machine (SVM) based call-type classification and acoustic modeling for speech recognition in the context of a telephone-based call center corpus (Tang 2003). The authors modeled this problem as a text classification problem by manually labeling the topics and then used them to place calls in different loads. However, the proposed model focuses mainly on handling the contact center performance rather than focusing on the business metrics. K-nearest Neighbor is a simple classification algorithm which is very effective and used very widely for text classification problems (Lan 2009). K. Gayathri et al. studied the performance of KNN and SVM classifiers (Gayathri 2013). However, in terms of time require for classification, the efficiency of KNN algorithm decreases with the increase of data dimensions. In another study, authors have investigated the effect of SVM one-class approach in a non-stationary environment (Ho 2013).

2.2.6. Clustering

To statistically evaluate the occurrences of words and group similar documents in text corpora (K. B. Ahmed 2014), (Hu 2014), and (J. D.-P. Fernández 2010), machine learning techniques like clustering is very useful. Clustering is an unsupervised method of machine learning as the numbers, properties, class memberships are not known in advance (Luger 2005). The authors in (Baumer 2010) presented different clustering based approaches for document retrieval and compared those techniques for logical pattern extraction from unstructured text.

Document clustering refers to the grouping of semantically related text documents (Hayes 1963). These days, one important focus of document clustering is to provide an efficient way to browse large collections of enterprise documents and World Wide Web data. The system should also be capable of representing the data in a structured manner. However, the preliminary aim of document clustering was to improve precision and recall of information retrieval systems. Clustering methods can be categorized as: (a) hierarchical clustering, (b) partitional clustering, and (c) semantic based clustering. A brief description of these categories has been presented in the following subsections.

2.2.6.1. Hierarchical Clustering

Hierarchical clustering organizes the group of documents into a dendrogram (tree structure) with a topic/subtopic relationship between the documents (Chen 2009). Two common methods of achieving hierarchical clustering is to use either: (a) agglomerative or (b) divisive methods (Kavitha 2010). Agglomerative method of clustering follows a bottom up approach by successively merging adjoining pairs of clusters together until the whole dataset form single large cluster. The closeness of clusters is determined by calculating the distance between the objects. On the other hand, divisive method of hierarchical clustering follows a top-down approach by starting with the complete dataset as a single cluster and then recursively splitting the cluster into smaller clusters until each document is in a classified cluster.

Hierarchical clustering is very useful because of the structural hierarchal format. However, the approach may suffer from a poor performance adjustment once the merge or split operations are performed that generally leads to lower clustering accuracy (Jain 2010). Furthermore, the clustering approach is not reversible and the derived results can be influenced by noise. In (Yonghong 2010), the authors suggested agglomerative hierarchical clustering techniques for document clustering.

2.2.6.2. Partitional Clustering

Partitional clusters determine the relationship between objects using a feature vector matrix (Kavitha 2010). Features of every object are compared and objects having similar patterns are placed in a cluster (F. a. Liu 2011). The partitional clustering can be further categorized as iterative partitional clustering, where the algorithm repeats itself until a member object of the cluster stabilizes and becomes constant throughout the iterations. However, the number of clusters should be defined in advance (F. a. Liu 2011). Some common forms of the iterative partitional cluster-based approaches include K-mean, K-medoid, C-mean, C-medoid, single-pass, probabilistic methods, and nearest neighbor (Kavitha 2010), (Jain 2010), and (F. a. Liu 2011).

2.2.6.3. Semantic-based Clustering

In semantic-based clustering, the structured patterns are extracted from an unstructured natural language data by utilizing meaningful context analysis of contents for knowledge extraction. Researchers have proposed several algorithms for computing semantic similarities between text documents. For instance, Resnick and Lin algorithms (F. a. Liu 2011) are proposed to measure the semantic similarity of natural language text in a specific categorization. WC Chen & MS Wang present a detailed descriptions of these algorithms in (Chen 2009). WT Yu and CC Hsu introduce an innovative technique to automate the ontology construction process utilizing data clustering and pattern tree mining (Yu 2011). The authors evaluated their proposed method using weather news data collected form e-paper and revealed remarkable results by extracting the regions with high temperature.

Finding similar objects requires the notion of similarity measure which is computed between the objects to decide their closeness. The following subsections briefly describes the methods used to determine the similarity or closeness of service request.

2.2.7. Similarity Measures

One important step in determining the closeness or likelihood of two documents is to measure similarity/distance between them. The measure determines the degree of closeness/separation of the objects by mapping the similarity/distance between two objects into a single numeric value. The chosen measurement should correspond to the characteristics that distinguish the clusters embedded in the data. However, in some cases, these characteristics are dependent on the problem context and this makes it very difficult to select any single measure suitable all domains. Choosing a similarity measure appropriate for a problem domain is also very crucial for a clustering algorithm. For instance, DBScan, a density based clustering algorithms relies greatly on the similarity computation (Ester 1996).

Some of the most popular techniques used for similarity measurements are Euclidean Distance, Cosine Similarity, Jaccard Coefficient, and Pearson Correlation Coefficient. A brief description of these methods along with Semantic Similarity have been presented in the following subsections.

2.2.7.1. Euclidean Distance

Euclidean distance is a standard metric for geometrical problems which measures the distance between two points. For most clustering problems, specifically in the domain of document clustering, Euclidean distance is widely used. For example, it is the default distance measure used with the K-means algorithm (MacQueen 1967). The Euclidean distance of the two documents, d_a and d_b, represented by their term vectors t_a and t_b respectively, is defined as follows:

$$E_{d(t_a,t_b)=}\sqrt{\sum_{t=1}^{m}\left|w_{t_a}-w_{t_b}\right|^2}$$

Here, the term set is $T = \{t_1,...,t_m\}$ and the term weight w is defined as:

$$w_{t_a} = tfidf(d_a, t)$$

2.2.7.2. Cosine Similarity

The similarity of two documents, represented as term vectors, corresponds to the correlation between those vectors and can be enumerated as the cosine of the angle between vectors. Cosine similarity is one of the most popular similarity measures applied to text documents including information retrieval applications and clustering (Rong 2012). Given two term vectors, t_a and t_b, the cosine similarity can be calculated using the following equation where t_a and t_b are m-dimensional vectors over the term set is $T = \{t_1,...,t_m\}$:

$$sim_c(t_a, t_b) = \frac{t_a . t_b}{|t_a| \times |t_b|}$$

Each dimension represents a term with its corresponding non-negative weight in the document and this results a non-negative cosine similarity value ranging from 0 to 1.

2.2.7.3. Jaccard Coefficient

Jaccard coefficient measures similarity as the intersection of the two objects divided by their union. For text document, Jaccard coefficient compares weights of shared terms to the weights of terms which are present in either of the two document but not in the set of shared terms:

$$sim_j(t_a, t_b) = \frac{t_a . t_b}{|t_a|^2 + |t_b|^2 - \ t_a . t_b}$$

Here, t_a and t_b are m-dimensional term vectors for the term set $T = \{t_1,...,t_m\}$. Like cosine similarity, similarity measurement for Jaccard coefficient also ranges between 0 and 1. Jaccard coefficient can also be used as a distance measure.

2.2.7.4. Pearson Correlation Coefficient

Pearson correlation coefficient is another similarity measure to relate two vectors are related. One commonly used form of this coefficient formula among many others is as follows:

$$sim_p = \frac{m \sum_{t=1}^{m} w_{t_a} \times w_{t_b} - TF_a \times TF_b}{\sqrt{\left[m \sum_{t=1}^{m} w_{t_a}^2 - TF_a^2\right]\left[m \sum_{t=1}^{m} w_{t_b}^2 - TF_b^2\right]}}$$

$$TF_a = \sum_{t=1}^{m} w_{t_a} \, and \, TF_b = \sum_{t=1}^{m} w_{t_b}$$

The document clustering module in our proposed IC mining system uses semantic similarity (Zaman 2014) for similarity measurement as it performs better in terms of finding a balanced cluster (A. Huang 2008). The semantic similarity, *sim_s*, is calculated using the shortest path between two entities in the document with the following equation:

$$sim_s = 1 - \frac{1}{2}\left(\frac{\sum_{i \in path(E_a,E_p)} w_i dis(E_i, E_{i+1})}{\sum_{i \in path(E_a,E_{root})} w_i dis(E_i, E_{i+1})} + \frac{\sum_{i \in path(E_b,E_p)} w_i dis(E_i, E_{i+1})}{\sum_{i \in path(E_b,E_{root})} w_i dis(E_i, E_{i+1})}\right)$$

$$w(E_a, E_b) = 1 + \frac{1}{k^{d(E_b)}}$$

Here, E_p is the nearest common parent of E_a and E_b; *w* is the weighting factor. $d(E_b)$ represents the depth of entity E_b from the root in the hierarchy. *k*, an user defined factor with a value of 2 in our case, defines the decreasing behavior of weight values from root to leaves.

The problem of incorporating semantic information within the document representation has recently enticed a lot of research attention. Hotho et al. (A. Hotho 2003) integrated conceptual account of terms within WordNet to examine its effects for unsupervised document clustering. The authors in (Y. a. Wang 2006) used WordNet to define a sense disambiguation method based on the semantic relationships among the senses and used that in document clustering algorithms like k-means. They discovered that incorporating the semantic information can improve the clustering performance. However, they have used most frequently used terms to represent the clusters. But, it is sometimes challenging to find a term appropriate for representing the cluster as different users have incongruous views for the same word. The clustering module in our proposed intellectual capital mining system avoids this problem by utilizing the extracted semantic entities and k-means algorithm to cluster together similar service requests data.

3. METHODS AND PROCEDURES

This chapter presents a detail description of the Intellectual Capital mining system architecture along with the procedures followed in each step of knowledge mining. The following subsection provides a brief scenario of the typical process followed by the service engineers to resolve a service request. In the later subsections, a detail overview of the system modules and the architecture of the system are presented. Following to these subsections, the working methodology of our proposed knowledge mining system has been described with specifics.

3.1. System Overview

In enterprise service centers, after receiving a call or email from a customer, a Service Request (SR) gets inserted in the database. These service requests generally include information about the customer requesting a solution, the problem statement, the product information related to the service request (if possible), and some other metadata. A service request also records all in-house and customer interactions, the tests and procedures followed by the support engineers. Based on the service environment, it is not very uncommon for support stuff or engineers in a customer support center to get an overwhelming number of requests from their customers and partners every day. Each of these incoming requests are assigned to the support engineers based on several factors- engineer’s expertise on problem domain, problem solving rate, availability, and so on. However, it very likely that a SR can be re-assigned to other support engineers somewhere in the middle of request resolution process. So, to make this service request assignment process smooth, it is very important to identify the problem category upon receiving a service request.

After an engineer gets assigned with the service request, the very first thing that support personnel attempts to achieve is an understanding of the problem described in that service request. However, extracting the problem keywords is not easy as in most cases service requests data is either semi-structured or unstructured. So the manual process of extracting these keywords, which will help the engineer to get an idea about the issue customer is seeking help with, is quite time consuming and troublesome. Once the support engineer gets a good understanding of the problem domain, she attempts to explore the approaches that can be applied to solve the problem. Again, this manual step embroils a thorough search within an enormous number of documents related to the problem domain and can be very time-intensive. The traditional methodology also incurs large amount of operating overhead as compared to an automated web-based support system which provide support knowledge directly to customers. Moreover, the request resolution process can take a significant amount of time based on engineer's familiarity with the problem domain. The resolution process to a service request merely takes any time if the engineer has past experience in solving similar problem requests. On the other hand, it can take a substantial amount of time for engineers to solve completely new, in other words never-seen problems. In such cases, engineers may examine different knowledge sources including the Internet, enterprise social network sites, organization's support forum, related service request solutions, enterprise white papers or technical reports and then read the relevant documents before summarizing the resolution process.

In an effort to better customer service knowledge management, organizations' deploy service request system which keeps track of the customer-service center interactions. These systems not only capture the case history in conversation/action format but also include information like the workflow, the final resolution, etc. However, this process is semi-automatic as engineers working a problem often end digging deep through the unstructured text containing written phone or email conversations, output from devices' command line interfaces which neither encourages knowledge sharing and reuse nor promotes convenient and quick resolutions. To make things worse, these service requests often contain duplicate data from repetitive email threads, textual data with syntax, grammatical, and/or typographic error, data that contain an extensive amount of acronyms and abbreviations, etc. As a result, once the service request is solved, the length of that SR can range from couple of pages to hundred pages or more. These days, most organizations are equipped with knowledge management systems to manage the knowledge base & support solutions in a superior way. It has also been proved beneficial for these systems to have the ability to capture request resolution process of previously requested problems and build structured knowledge for future reuse. Thus, after receiving a service request, the system will attempt to match the service request with already solved similar request resolution processes. In such way, engineers can reduce both the time and cost related to the request resolution process considerably. Such knowledge management systems will also expedite efficient responses to customer inquiries and resolutions.

However, there are several challenges that need to be addressed in the existing systems. First of all, determining the problem category upon the arrival of a service request; which will help to propagate the SR to a support engineer having appropriate domain knowledge and expertise. Secondly, to extract the Intellectual Capital (IC) which describes the actual problem, the impacts of that problem, and possible recommended techniques to solve the problem, engineers have to search though those bulk textual data. The third challenge is cold-start problem handling- service requests which are not related to some previously solved service problems cannot be benefited from such system. Moreover, these systems do not have the option to utilize up-to-date information from other sources such as the social network discussions between users and experts on support problems. In most cases, the system may not have the capability quick incorporation of e-mail exchanges. IC management system in our work, handles these challenges efficiently and provide support solution using most relevant information related to the service requests, even if the users’ re-quests are new to the system.

Our proposed IC management system aims to help support engineers and customers finding accurate information related to the service problems. Another important feature of knowledge management systems is, the knowledge should be presented and managed easily to promote better understand and reuse of knowledge. To carry out this goal, we implemented an integrated custom search engine by customizing Google Custom Search which locates documents related to the service requests. However, these documents contain large amount of noise. The system then utilizes a multi-level classifier to remove noise, extract knowledge and contour the mined information into a useful representation. The next step implements a semantics-assisted classifier which classifies the clean, preprocessed data to a structured format which then can be easily utilized by service center engineers or customers.

Our IC management system annotates unstructured data with domain knowledge utilizing semantic web technologies. This marked up information empower the system to handle knowledge management in a more accurate and intelligent way. As mentioned in earlier sections, ontology gives a platform for sharing the common understanding of a domain and thus can be communicated across people and application systems, and thus simplify the processes like knowledge sharing and reuse. The ontology assisted IC mining tool in our knowledge management system deduces the relationships between key concepts automatically which in turns enables the system to perform an accurate knowledge extraction & management.

Our proposed IC management system comes with an interface which allows user feedback during query refinement, problem characterization and results verification. Moreover, the system is equipped with a re-training module which allows users to incorporate feedback to facilitate system learning thus improve overall system performance & accuracy. In other words, to discover & organize intellectual capitals efficiently, our IC management system attempts to employ machine learning techniques in a combination with the knowledge provided through human feedback. The following subsection briefly describes the overall system architecture.

3.2. System Architecture

As depicted in Figure 3, our proposed Intellectual Capital mining system is comprised of five main components: a technology/sub-technology identifier, a custom IC search module, entity extractor, intellectual capital extractor, and IC repository. Another significant component of our IC management system is the enterprise ontology. All of the five abovementioned components utilize the domain knowledge to improve system performance through this predefined & pre-generated semantic information.

After a service request is created in the database, it needs to be escalated to a customer support engineer who has expertise in the service request problem domain. This service request assignment process is very crucial as improper assignment can make the request resolution process longer than expected. One very important assignment criteria is knowing the technology/sub-technology group a service request problem belongs to. However, this field can be entered either by the customer when submitting an online support request or can be entered by the front-end support personal who might not have the proper knowledge to identify the category instantly. As a result, in most cases, the service requests do not get tied with the proper category which in turns results into an incorrect service request to service support engineer. To solve is issue we build a technology identifier which examines the problem description section of a service request thoroughly and then output the precise technology or sub-technology with the help of semantic relationships. The next step in mining intellectual capital from huge, heterogeneous data sources is to select the documents related to the service requests and having high intellectual value. The custom search module of our IC mining system aids the engineers in mining such data. The keywords entered by the service engineers can also be enhanced semantically to filter the search results. After retrieving the documents related to the search query, they are fed into the entity extractor module. Entity extractor removes the noisy and irrelevant portion of the data by performing data cleansing. The entity extractor module in our IC mining system extracts semantic entities which are then used in combination with the statistical features to classify the different IC segments. These features are then enhanced with the concepts mined from the domain specific ontologies. We have utilized a maximum entropy classifier to identify different IC categories. Maximum entropy classifier has proven to produce effective text classification results (Elder IV 2012). We attempt to decrease the size of the effective vocabulary and eliminate noisy features in the feature selection step. Moreover, we try to determine features most relevant to the classification process as some of the words are much more likely to be correlated to the class distribution than others. Besides commonly used preprocessing technologies such as tokenization, stop-word removal, stemming, and parts-of- speech tagging, our preprocessor also selects the significant keywords and phrases that carries important semantic meanings of the documents and contribute more to distinguish between documents.

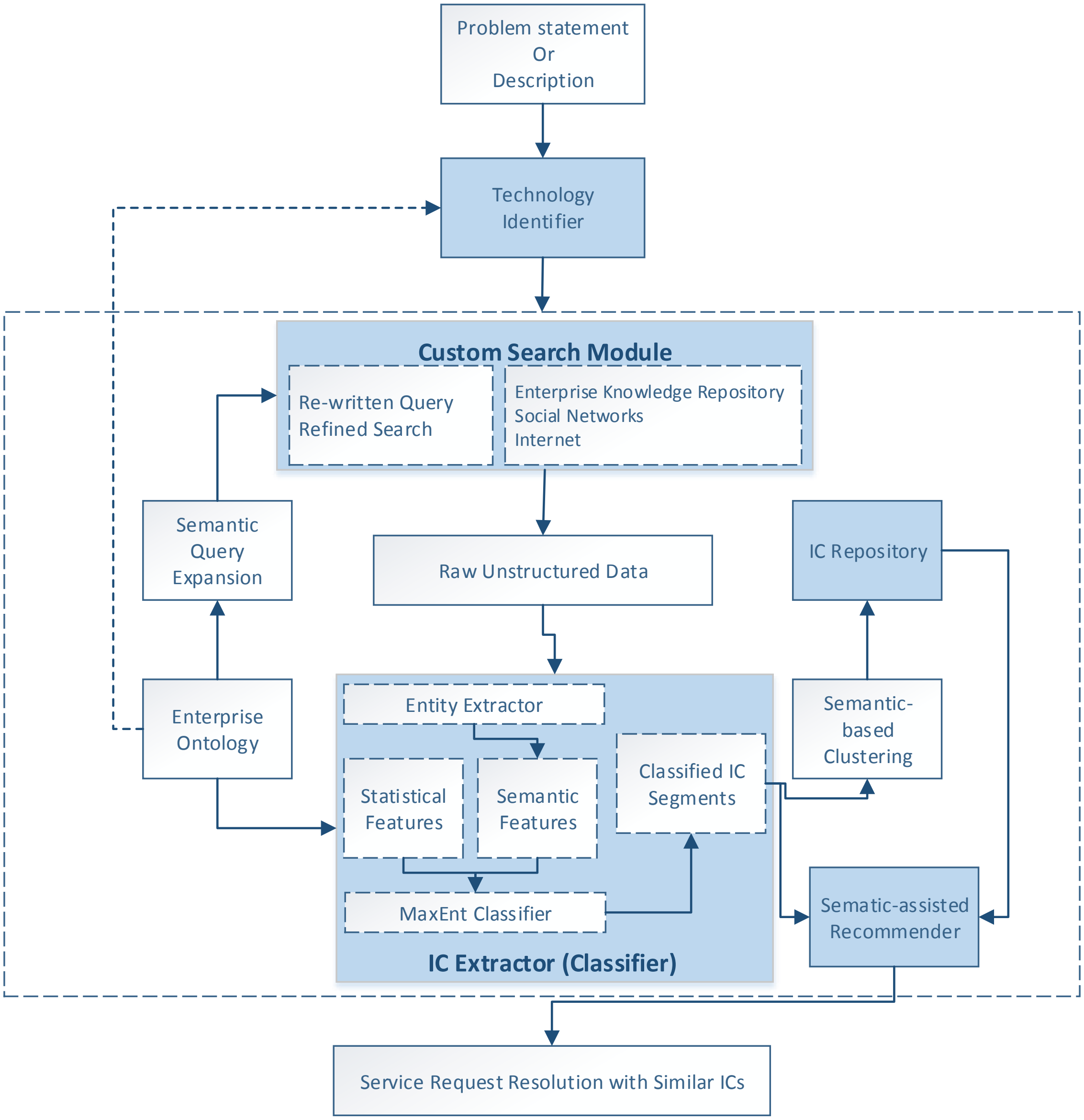


Figure 3. IC Management System Architecture

At the next phase of knowledge extraction, the engineers formulate the IC after verification and summarization of classified results and store it in the IC repository utilizing clustering algorithm to group similar problem cases. This helps users locate a group of similar problems and their best practice solution. Engineer's corrections of the classification will be observed and automatically forwarded as feedback to the classifier for continues learning.

The following subsections represents the working procedure for each of the components used in our knowledge extraction tool.

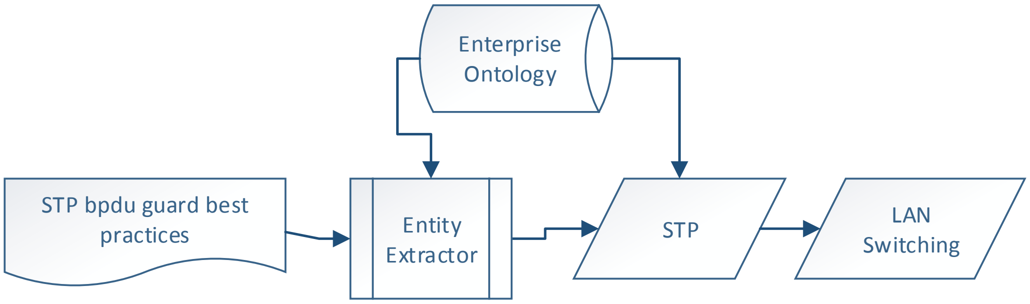


Figure 4. Tech/Sub-tech Identifier Example Scenario

3.3. Tech/Sub-tech Identifier

As mentioned in the previous section, to categorize the technology or sub-technology group for a service request, we exploited the semantic relationships between the entities presented in the problem description. Service request management system in an enterprise customer service center stores SR data as a combination of semi-structured and unstructured format. The SR metadata sometimes contains a field which stores a short description of the problem. Technology identifier in our IC management system accomplish its goal in two steps- firstly, it takes description text as an input and preprocess to discard the unnecessary clutters. Once the preprocessing is complete, technology identifier further processes the extracted entities to detect the problem or technology category they fall under. We use the enterprise ontology along with the entity extractor module, described in section 3.5, to find out the entities that help determining the technology/sub-technology group.

Figure 4 presents an example scenario how this identifier module process the problem description- “*STP bpdu guard best practices*”. The entity extractor takes this descriptive text and identifies STP as an important term. Later, we have followed a rule-based approach to find out the technology group STP belongs to. In enterprise ontology database, entities can be directly tied with one or more technologies using the property *prod_has_tech*. However, in most cases, such direct relationship cannot be found and to solve those cases, we need to traverse back towards the root in the ontology graph to determine the technology group if there is any. The latter case is applicable in our example scenario which has been depicted in Figure 5.

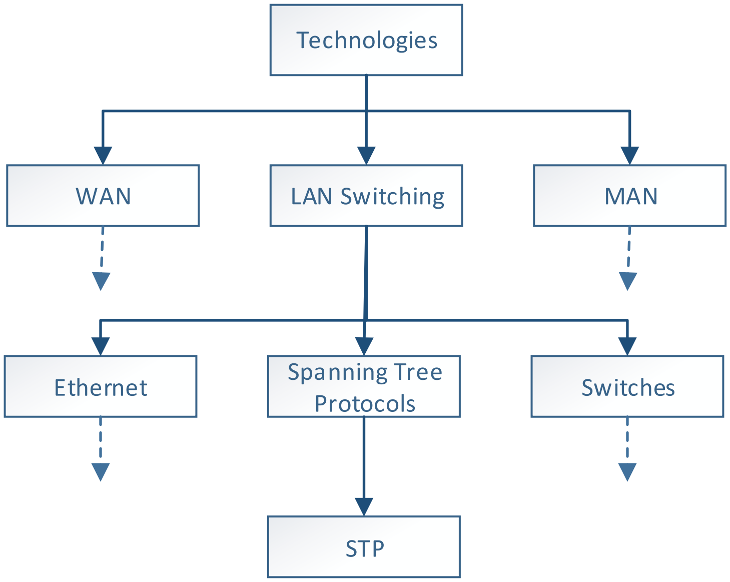


Figure 5. Part of the Ontology Graph

3.4. Custom Search Module

The custom search module in IC mining system searches through the world wide web data and categorizes the IC keywords relevant webpages and documents into the different groups- enterprise documents, social forums' data, and general. For instance, in the context of a network service center, a customer service engineer would likely be interested on how to configure different routing protocols on various router models. Those queries can be expanded to help users polish, and disambiguate their queries, thus finding the most relevant results. This expansion is achieved by combining Google's Custom Search API and enterprise ontology. Through the refinement function of Google Custom Search, the search module in our IC management system can associate the sites with topics by creating sophisticated tags. Moreover, the grouped search results also enable the users to customize the ranking of the search results, such as preferring webpages from a specific site/forum.

3.5. Entity Extractor

The documents and webpages, returned by the search module contain the information related to user queries. However, the data required by the users are buried under substantial amount of irrelevant and noisy information which needs to be discarded before making them machine understandable. The manual approach of relevant information mining out of this massive amount of irrelevant data involves reading & understanding the contents of these web pages and documents which is very time and energy consuming. In other words, mining the documents & webpages from organizational repository or social forums is not sufficient to collect the information required by service engineers to solve service requests. This leads to another problem- extracting specific chunk of the information from these documents. Another important task is to make a compact representation to the extracted information needs so that it can be easily understood and consumed by the service center engineers. This form of data representation also help to create machine understandable data. So these extracted information are then used as features by the IC extractor module to classify the IC segments. One important step in feature selection is to preprocess the massive data found in previous step. Before applying the preprocessing technologies such as tokenization, stop-word removal, parts of speech tagging, stemming, our preprocessor module performs a data deduplication & cleansing method. Moreover, the substantial keywords and phrases which carry essential semantic connotations are also get extracted by the preprocessor module. A brief description of these preprocessing techniques has been documented in the following subsections.

3.5.1. Preprocessing

The preprocessor module within entity extractor removes noisy, irrelevant data and obtains key features to enhance the relevancy between phrases and document and category. The very first step of our knowledge mining systems is to remove unnecessary data like document markups- HTML/XML tags, duplicate information, and so on. In next round of data preprocessing, sentence segmentation is done using the Punkt sentence segmenter proposed by Kiss et al (Kiss 2006). Segmented data is then tokenized using an adopted Penn Treebank Tokenizer (Marcus 1999). These tokens roughly correspond to "words". The preprocessor then removes stopwords if there is any and also performs word stemming. Improved Porters Algorithm (Willett 2006) has been used to perform word stemming. A brief description of the data deduplication method and preprocessing tools used by the IC mining system has been presented in the following subsections.

3.5.1.1. Tokenization

One frequently performed preliminary step in natural language processing & Information Extraction is splitting a text into sentences and then to words. This process of splitting is termed as segmentation or tokenization. Sentence segmentation is also called sentence boundary detection. We have introduced sentence segmentation and word tokenization in the following subsections.

3.5.1.1.1. Sentence Boundary Detection

Sentence boundary detection or sentence segmentation aims to divide the textual document into sentences. Detecting sentence boundaries is challenging as punctuations used to indicate the end of sentences can have other usage as well. For example, apart for marking sentence ending, a period is also used to mark abbreviations. Moreover, some periods concurrently indicate sentence termination and abbreviations. The preprocessor module in our IC mining system splits the document into sentences considering all these conditions. The next step involves tokenizing the sentences into words.

3.5.1.1.2. Word Tokenization

In general, word tokenization is an early step of processing where a text is segmented into basic units such as words, numbers and punctuations. Tokenization based on whitespace is inadequate for many applications because it bundles punctuation together with words. On the other hand, punctuation along cannot be used as the splitter. For instance, the text segment- *"…software release 12.4(24)T"* cannot be tokenized based on punctuation, period in particular, as we may lose some valuable information. Furthermore, different locale have different usage of punctuations. Thus, tokenization is considered to be a language specific task which can be solved either by a system trained with manually tokenized texts or by using some hand-crafted rules. Our preprocessor module's tokenization algorithm has been designed to accommodate all these characteristics of service center domain. More specifically, we have added certain predefined rules to enable the tokenizer handle these circumstance. So after tokenizing the above mentioned example sentence fragment, we will have *"12.4(24)T"* as a single token.

3.5.1.2. Data Deduplication

Another important task of our pre-processing module is to perform data deduplication. In most cases, problem requests contain redundant information which should be discarded to improve the performance of IC mining system. One such example of redundant data is the auto-embedded part of the original message that gets appended each time a response is created for a service request in customer service centers. We have adopted the document resemblance method proposed in (Muhammad Rafi 2010) where the authors defined resemblance *r(A, B)* of two documents, *A* and *B*, having a value in between 0 and 1. In other words, if the resemblance is close to 1 then it is very likely that the documents are roughly the same. Formally, resemblance, *r*, can be defined as:

$$r(A, B) = \frac{|t(A) \cap t(B)|}{|t(A) \cup t(B)|}$$

The word tokens, processed in the previous steps, are fed into the deduplication module before continuing with the remaining preprocessing tools. The preprocessor component is able to discard a significant amount of irrelevant data after performing this step.

3.5.1.3. Stopwords Removal

Before further processing of the documents and queries, a very usual & important step in almost all IR applications is stopwords removal. This is another step towards reducing the amount of unnecessary and context irrelevant information. Stopwords removal aims to eliminate function words, low-content words, very high frequency words in order to achieve an increased system performance. In addition to the common English language stopwords, the preprocessor module in our IC management system also eliminates the domain specific stopwords.

3.5.1.4. Parts of Speech Tagging

The ultimate research focus of NLP is to parse and understand the language. To achieve this goal, most researchers have focused on transitional task like inherent language structure identification. One such common preprocessing task of natural language processing is *part-of-speech tagging* (POS tagging) or simply *tagging* where the words are classified and labeled into their parts-of-speech. Some of the existing techniques of POS tagging include statistical (Manning 2008) (A. Ratnaparkhi 1996), memory-based (Brants 2000), rule-based (Daelemans

1996) (Brill 1992) methods. For our system, POS tagging plays an important role to identify the relevant feature. In our pre-processing module, we have used a statistical based POS tagger.

3.5.1.5. Stemming

Stemming involves linguistic normalization which removes the prefixes and suffixes from a word or token. Through stemming several terms are mapped onto one base form, which is then used as a term in the vector space model. This means that, on average, it increases similarities between documents or documents and queries because they have an additional common term after stemming. Though this increase the recall, but sacrifices the precision. However, in our case, stemming seems to have little or no effect on precision.

Stemming can be done using either one of the two available methodologies: (a) dictionary-based stemming, and (b) Porter-style stemming (Willett 2006). For our work, we have chosen to work with Porter-style stemming.

3.5.2. Named Entity Extraction

Named Entity Extractor module in the IC mining system aims to identify the features to be used in user query expansion, with our semantics assisted classifier, and also with the clustering module. These extracted entities improve classification performance as they contain the important contextual information about the documents being processed. The classification performance was then enhanced further by expanding the semantic representation of the documents. In other words, the identified entities were extended using the relationships defined by the domain ontology. In our system, we have used the most popular hierarchical specialization/generalization (or IS-A) relationship and the type relationship between a particular class and its corresponding instances. To efficiently locate related concepts bounded by the above-mentioned two relationships, two other inverse index tables were used where each keyword is associated with its corresponding super classes (or type class for individual objects). Each semantic entity within the document gets expanded with all of its ancestor concepts up to a maximal distance. It is important to note that the distance parameter needs to be chosen carefully as climbing up the taxonomy too far is likely to obfuscating the concept representation (Bloehdorn 2004). The following subsections describe the features used by the Entity extractor module of our IC Mining system.

3.5.3. Features of NER

Feature selection is the most important aspect of a named entity recognizer system. The machine understandable characteristics (a Boolean value, a numerical value, etc.) of a word, defined by the domain experts, comprise the feature set for a NER system. Features are also known as indication functions for the named entity recognizer systems. A brief description of some of the popular features of NER systems are presented in the following subsections:

3.5.3.1. Word Features:

This category of considers current word, previous word, next word and all words within the window as features for a NER system. In additional to these there can be other word features like- orthographic features (Fargo→ Xxxxx, ND-58103→ XX-#####), prefixes and suffixes (Fargo→ <F, <Fa, <Far... rgo>, go> o>), stem (stems, stemmer, stemming → stem), n-grams, length and so on.

3.5.3.2. Dictionaries:

Abbreviations, stop words also comprises the feature set for NER. Gazetteers containing entities like first names, last names, locations are also used as the dictionary based features for NER.

3.5.3.3. Metadata:

Metadata in a document like position of a word in a sentence, word frequency, words co-occurrences make feature set for NER systems.

3.5.4. Lookup Techniques

Lookup technique is one of the crucial tasks of gazetteer based NERs. Some common techniques used to lookup in the dictionary include the following:

- *Exact matching*- with this lookup technique, a dictionary entry is exactly matched with the phrases in the document.
- *Approximate matching*- method to find phrases having approximate match for a given pattern. This method is also known as fuzzy-text matching. The approximate matching uses edit distance as the similarity measure between the strings. The number of unit operations needed to convert one string into another determines the similarity measures between two strings.
- *Lemmatization-based matching*- lemmatization is a process to determine the root form of a word. For instance, the word “done" is lemmatized to its lemma “do". Lemmatization is done using the parts-of-speech tag of a word before matching the phrases in a dictionary. In other words, lemmatizer applies parts-of-speech specific rules to the word.
- *Soundex based matching*- Soundex refers to a string alteration method which transforms a word into its sound as uttered in English. In other words, English words are decoded such that the similarly spelled or pronounced words build the same code. Soundex helps with misspelled words during document processing.

Exact matching is the most used and simplest lookup technique among the 4 abovementioned methods. In contrast, approximate matching is not language dependent and complex in terms of implementation and running time. However, accuracy of approximate matching as a lookup technique outperforms the rest. For our IC mining engine, we have used both approximate & exact matching to extract the named entities.

3.5.5. Bag of Words

Bag of words (BoW) model is the representation of the text document as a set of words contained in that document. The BoW model is often used in natural language processing and information retrieval because of its simplifying approach. One common usage of BoW model is document classification where the words and their frequencies are used as features for the classifier. The *term frequency-inverse document frequency (tf-idf)* weighting scheme has been used for text representation in (Li 2003). In another research (Dumais 1991), authors improved the average performance by 30% by using global IDF and entropy weighting scheme along with tf-idf. Several other weighting schemes have been proposed in (Eikvil 1999), (Joachims 1997), and (Jones 2000) to improve the performance of general bag of words model. However, BoW model often suffers from issues like overfitting and poor system performance when the set of words or terms is huge. We have utilized the BoW model to capture the frequently used important word phrases to be used as a feature in our IC extractor module.

3.5.6. N-grams

N-gram is a sequence of *n* items (characters or words) extracted from the given text. N-grams can either be character based where n-gram is a set of *n* consecutive characters from a word in the document or it is a set of *n* words if it is word based. Word based n-grams are often called *shingles*. Character n-grams often make document preprocessing language independent and simple as they may span across word boundaries (Monz 2002). However, the cost of index size also increases with this approach. Our Named Entity Extractor module utilizes a tri-gram gazetteers which has been built using enterprise ontology and bag of words.

3.6. IC Extractor

Our IC extractor is based on an ontology-assisted classification approach in which semantic entities, enhanced with concepts extracted from the domain ontologies, are used as features.

3.6.1. Ontology-guided Feature Selection

So far, most existing text classification systems have adopted the Bag-of-Words (BoW) model where single words or word stems are used as features and word frequencies or weighting schemes like TF-IDF are used as feature values (Rong 2012). However, the BoW model ignores the conceptual relationships and domain knowledge. For example, using BoW model, the multi-word entity "catalyst 5000" will be treated as totally different things although semantically they are closely related. This problem is addressed by utilizing organization's ontology to identify important concepts and relationships between those concepts. These identified semantic entities were then extended with their semantically related concepts and select them as features. In this way, semantic meaning of the feature will be preserved and classification would be more accurate.

However, before utilizing semantic entities as features for classification, the non-trivial problem of Entity Recognition (ER) needs to be addressed. Traditional NER works to identify all textual references of named entities- noun phrases referring to specific individuals like persons, organizations, location and so on and do not consider the usage of the ontology as a reference.

Algorithm 3.1 The semantic entity extraction algorithm

```
Input: document $d=\{t_1, t_2, ..., t_n\}$, // $t_i$: token
          ngram_dictionary $n=\{ n_1, n_2, ..., n_n\}$,
          entity set $s=\{\}$

for each $t_i$ in $d$ do
    for each $n_j$ in $n$ do
        if isValid($t_i$) and $e_j$ contains $t_i$ then // $e_i$: reversed index of n-gram entity
            tag $t_i$ with $e_j$'s ID
for each $t_i$ in $d$ do
     phrase p=null
     returnedPhrase rp = AddToEntityPhrase($t_i$, p)
    if rp != null
      p = rp
    else
      printEntity(p)
 return entity
```

Besides, the form of a named entity in free text can be significantly different from its ontology version. For example, the semantic entity ‘Cisco Catalyst 5000 Series Switches’ in the ontology might be referred to as ‘Catalyst 5000’ in the text. Since entities in the ontology are represented as strings, the ontology-guided named entity extraction problem can be modeled as an approximate string matching problem. Our proposed approximate semantic entity recognition utilizes the well-known string-based dissimilarity measure – Levenshtein distance (W. C. Wang 2009). The potential entity extraction methodology is given in Algorithm 3.1 - 3.3.

Algorithm 3.2 AddToEntityPhrase Algorithm

```
Input: token $t_i$
          phrase p

if TaggedAsEntity($t_i$) and TaggedAsEntity(p)
    if TokensFromSameEntity($t_i$, p)
        return merge($t_i$, p)
    else
       return null
```

Algorithm 3.3 PrintEntity Algorithm

```
Input: phrase p

entityList = []
if ngramAcronymDictionary contains p
    entityList.add(p)
if ngramSynonymDictionary contains p
    entityList.add(p)
if entityList is not empty and generalNgramDictionary contains p
    entityList.add(p)

return entityList
```

3.6.2. Semantic Extension

The performance of classification is further enhanced by expanding the identified entities with semantically relevant entities based on the most important relationships: the hierarchical specialization/generalization (or IS-A) relationship and the type relationship between a particular class and its corresponding instances. Classification with ontology enhancement will capture the semantics of the text by overcoming the shortcomings in the syntax level. For example, in the training data, we have a case on how to configure a router. In the testing data, we have a similar case which is about configuring a switch. Without considering the semantics of the data, the classifier may fail to catch the semantic relationship between these two cases as entities “router” and “switch” share the same semantic ancestor entity ‘Network Device’.

Table 3. Feature List

Semantic Features	**Statistical Features**
expanded semantic entities	*n*-most frequent words
type of data source	presence of query keywords
bag-of-hit-words	length of paragraphs
	relative location of a paragraph

3.6.3. Classification

As mentioned in earlier section, we choose to deploy maximum entropy classifier (Banerjee 2007) for classifying the IC segments in our system as. Table III lists the semantic and statistical features that have been used by the classification module in our IC Mining system. For semantic entities, we use their presence instead of using their frequency count. In other words, if the feature is present, the value is 1, but the value is 0 if that feature is absent in the document. Document source should also be incorporated with the feature list as they require different processing methods. For example, social network discussions should be processed differently from whitepaper documents. A document, especially an enterprise whitepaper may include multiple topics/sub-topics. IC-relevant problem may be contained within a minor part of the document. Therefore, whether a paragraph contains the query keywords should also be considered as a feature related to determine if that paragraph is IC associated.

The document containing ICs are disproportionate in most cases in terms of the number of IC-relevant paragraphs and IC-irrelevant paragraphs: on average the IC relevant information is one tenth of the irrelevant information. In most cases, the classifier's performance degrades considerably on imbalanced data-sets as they are designed to minimize the global error rate (A. S. Fernández 2008). To address the issue of biased data, a multi-level hybrid-sampling classification mechanism is proposed in our system. At the first level of classification, we identify and remove noisy information. Then at the next level, we distinguish IC-relevant information from irrelevant ones by utilizing two effective methods- under-sampling (Chawla 2002) & over-sampling (X.-Y. J.-H. Liu 2009). We then further classify them to different IC categories- criteria, impact and recommendation after IC-relevant data have been identified. Here, *Criteria* is the principle or standard by which the problem or the service request may be judged or decided, *Impact* is the influence caused by the problem and *recommendation* is the suggestions to solve the problem. So as an output from the IC extractor, we retrieve an intellectual capital related to the search query which specifies the problem area followed by the impact they might have on the system and the possible solution to that problem. Once we have all the ICs related to specific problem or service request, the next step in our IC management system is to make a repository to make these mined ICs reusable for future use.

3.7. IC Repository

The task of document clustering can be divided into two sub-tasks: *first*, the semantics of the documents need to be represented in a machine understandable way, and *second*, a similarity measure needs to be defined based on the semantic representation such that it documents having higher semantic relationship get higher numerical values (Muhammad Rafi 2010). The techniques proposed by various authors differ in terms of document representation, semantic measure, and usage of background semantic information.

Semantic similarity plays very vital role in tasks like natural language processing, information retrieval, text categorization, document clustering and so on. To store the ICs in the repository, we applied k-means algorithm for document clustering using semantically enhanced data sets. The findings from performance evaluation shows that incorporation of semantics with the dataset significantly improves the clustering performance. A brief discussion on the result of this experiment evaluation has been presented in section 4.6.

3.8. IC Recommender

For customer support engineers, it is sometime very helpful to go through the request resolution process of some similar ICs he currently is working on. This not only helps the support engineers to generate idea on the possible solutions for current service request, but also assist with a better understanding the problem domain.

Our proposed IC management system includes a recommender module which takes the IC stored in the repository as an input along with the new request and utilizes semantic enhanced k-nearest neighbor approach to find out most similar service request(s). We have considered only the problem description paragraph(s) of the stored knowledge capital for similarity measurement. Instead of using traditional similarity measurements to determine the closeness between service request problem descriptions, we have calculated the semantic similarity using the approach described under the section 2.4.5.

4. EXPERIMENT RESULTS

This chapter describes the experimental evaluations of our implemented IC Management system which consists of five different components- technology identifier, a custom content search module, entity extractor, IC extractor (auto and manual), and an IC repository. Each of these five major components utilizes the contextual information in the form of semantic entities.

Several experiments are performed to evaluate the performance of the implemented IC management system. In all of these experiments, the minimum IC unit is a paragraph from a service request and each paragraph is classified as one of the three categories:

i) *Criteria (BP Problem):* The description text used by the customer or the support engineer to define the problem. This is termed as BP (Best Practice) problem in the implemented toolkit.

ii) *Impact:* Possible influences that problem can have.

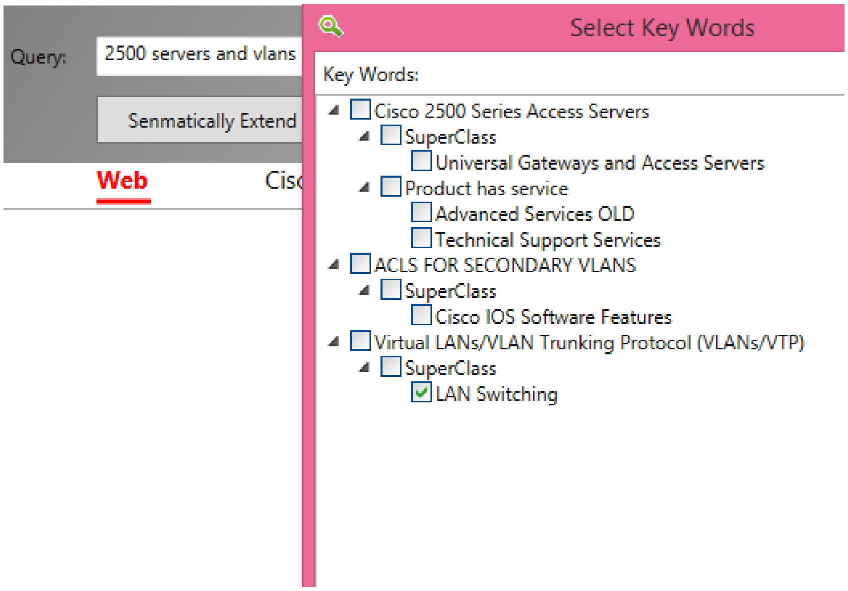
iii) *Recommendation:* The request resolution process suggested to the customer.

iv) Irrelevant: A paragraph is considered as irrelevant if it does not any information related to the problem context.

4.1. Data Set

We conducted experiments for the technology identifier module using a dataset containing problem descriptions of 270 service requests data. On an average, the length of the problem description data ranges from 3 lines to 10 lines.

In the next set of experiments, to evaluate the performance of the classifier, we consider 109 manually tagged Best Practice (BP) use cases of Cisco Intellectual Capital Mining team. The documents returned by the IC search module were cleaned & preprocessed to remove noisy, irrelevant, and clutter data. After pre-processing, human experts read the documents and tag each paragraph of the documents as one of the four categories- irrelevant, criteria, impact, and recommendation.

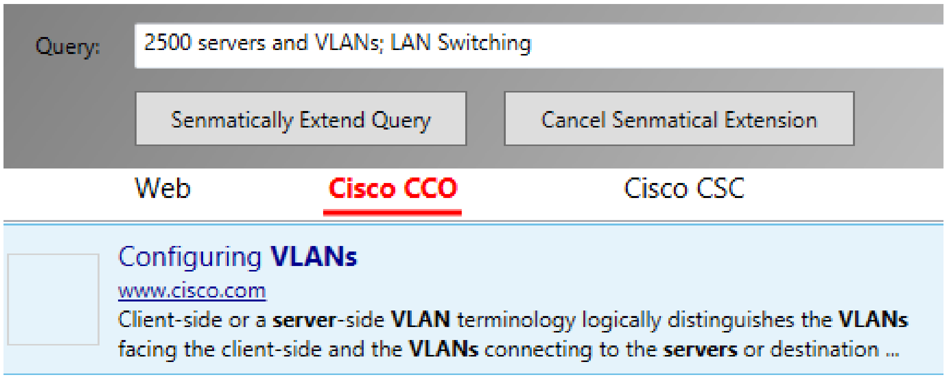


Query: 2500 servers and vlans

Key Words:

- □ Cisco 2500 Series Access Servers
  - □ SuperClass
    - □ Universal Gateways and Access Servers
  - □ Product has service
    - □ Advanced Services OLD
    - □ Technical Support Services
- □ ACLS FOR SECONDARY VLANS
  - □ SuperClass
    - □ Cisco IOS Software Features
- □ Virtual LANs/VLAN Trunking Protocol (VLANs/VTP)
  - □ SuperClass
    - ☑ LAN Switching

Figure 6. Query Expansion (a)



Query: 2500 servers and VLANs; LAN Switching

Configuring **VLANs**
www.cisco.com
Client-side or a **server**-side **VLAN** terminology logically distinguishes the **VLANs** facing the client-side and the **VLANs** connecting to the **servers** or destination ...

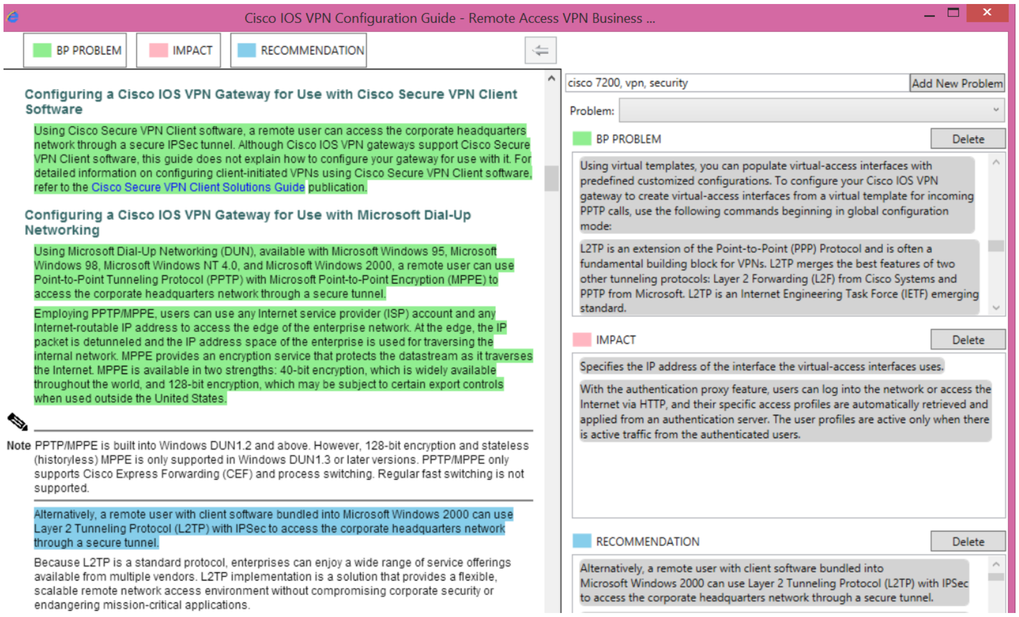
Figure 7. Query Expansion (b)

One very productive method for classifier feature refinement is to execute error analysis which we adopted during the second phase of the experiments. A development set containing the corpus data is selected to create the model at the very first level of this method. In the next step, development set is further divided into the initial training set and the development test set. The first data set is used to build the model and then error analysis is prepared using the development test set. Error analysis is very helpful to fine tune the classifier model by adjusting the feature set. Once the model is adjusted for a better performance, it can be applied on the original test data set. We split the development data set with a random partition of 80% data in initial training set and the rest 20% is the development test set.

At the third level of experiment, we evaluated the performance of the IC repository module to see how effectively it can cluster together related service request data. For this experiment we have considered the paragraphs tagged as problem description/criteria of the 109 manually tagged datasets. At the very last experimental phase, we evaluated the performance of IC recommender model using the same datasets.

4.2. Use Case Scenario

At the very first step of service request resolution process using our IC management system, the customer support employee enters the problem description in technology identifier module to determine the technology/sub-technology group of the service request. After this step, the service request is assigned to a domain expert in that identified technology/sub-technology group. Once the assignment is complete, the support engineer enters the keywords to the IC custom search module. For instance, Figure 6 represents a scenario where the support engineer is searching with the keywords “*2500 servers and VLANs”*. Figure 7 also displays that the entered query can also be extended using related semantic concepts and relationships. This feature is very useful as it allows the support engineers to narrow down the problem domain and to refine the query. To use this feature, support engineer click the Semantically Extend Query button in the page and then he can select the appropriate concepts/relationships from the window that popped-up after the button click.



Configuring a Cisco IOS VPN Gateway for Use with Cisco Secure VPN Client Software

Using Cisco Secure VPN Client software, a remote user can access the corporate headquarters network through a secure IPSec tunnel. Although Cisco IOS VPN gateways support Cisco Secure VPN Client software, this guide does not explain how to configure your gateway for use with it. For detailed information on configuring client-initiated VPNs using Cisco Secure VPN Client software, refer to the Cisco Secure VPN Client Solutions Guide publication.

Configuring a Cisco IOS VPN Gateway for Use with Microsoft Dial-Up Networking

Using Microsoft Dial-Up Networking (DUN), available with Microsoft Windows 95, Microsoft Windows 98, Microsoft Windows NT 4.0, and Microsoft Windows 2000, a remote user can use Point-to-Point Tunneling Protocol (PPTP) with Microsoft Point-to-Point Encryption (MPPE) to access the corporate headquarters network through a secure tunnel.

Employing PPTP/MPPE, users can use any Internet service provider (ISP) account and any Internet-routable IP address to access the edge of the enterprise network. At the edge, the IP packet is detunneled and the IP address space of the enterprise is used for traversing the internal network. MPPE provides an encryption service that protects the datastream as it traverses the Internet. MPPE is available in two strengths: 40-bit encryption, which is widely available throughout the world, and 128-bit encryption, which may be subject to certain export controls when used outside the United States.

Note PPTP/MPPE is built into Windows DUN1.2 and above. However, 128-bit encryption and stateless (historyless) MPPE is only supported in Windows DUN1.3 or later versions. PPTP/MPPE only supports Cisco Express Forwarding (CEF) and process switching. Regular fast switching is not supported.

Alternatively, a remote user with client software bundled into Microsoft Windows 2000 can use Layer 2 Tunneling Protocol (L2TP) with IPSec to access the corporate headquarters network through a secure tunnel.

Because L2TP is a standard protocol, enterprises can enjoy a wide range of service offerings available from multiple vendors. L2TP implementation is a solution that provides a flexible, scalable remote network access environment without compromising corporate security or endangering mission-critical applications.

cisco 7200, vpn, security

Problem:

BP PROBLEM

Using virtual templates, you can populate virtual-access interfaces with predefined customized configurations. To configure your Cisco IOS VPN gateway to create virtual-access interfaces from a virtual template for incoming PPTP calls, use the following commands beginning in global configuration mode:

L2TP is an extension of the Point-to-Point (PPP) Protocol and is often a fundamental building block for VPNs. L2TP merges the best features of two other tunneling protocols: Layer 2 Forwarding (L2F) from Cisco Systems and PPTP from Microsoft. L2TP is an Internet Engineering Task Force (IETF) emerging standard.

IMPACT

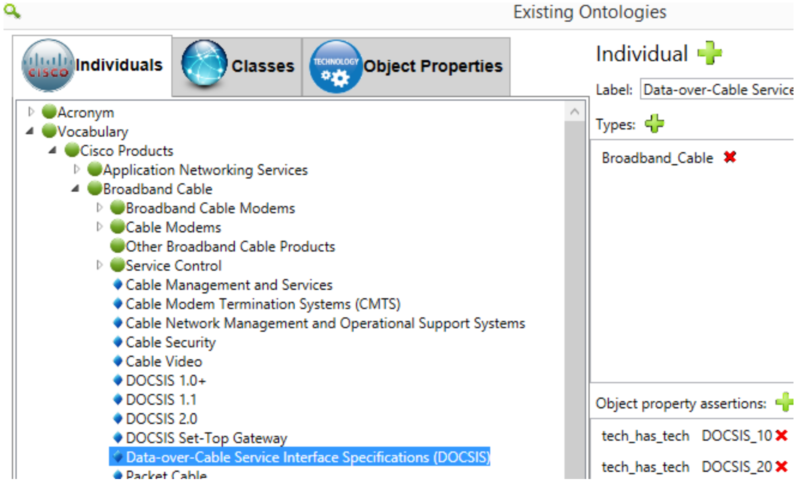
Specifies the IP address of the interface the virtual-access interfaces uses.

With the authentication proxy feature, users can log into the network or access the Internet via HTTP, and their specific access profiles are automatically retrieved and applied from an authentication server. The user profiles are active only when there is active traffic from the authenticated users.

RECOMMENDATION

Alternatively, a remote user with client software bundled into Microsoft Windows 2000 can use Layer 2 Tunneling Protocol (L2TP) with IPSec to access the corporate headquarters network through a secure tunnel.

Figure 8. Screenshot of IC Extractor



- Acronym
- Vocabulary
  - Cisco Products
    - Application Networking Services
    - Broadband Cable
      - Broadband Cable Modems
      - Cable Modems
      - Other Broadband Cable Products
      - Service Control
      - Cable Management and Services
      - Cable Modem Termination Systems (CMTS)
      - Cable Network Management and Operational Support Systems
      - Cable Security
      - Cable Video
      - DOCSIS 1.0+
      - DOCSIS 1.1
      - DOCSIS 2.0
      - DOCSIS Set-Top Gateway
      - Data-over-Cable Service Interface Specifications (DOCSIS)
      - Packet Cable

Individual

Label: Data-over-Cable Service

Types:

Broadband_Cable

Object property assertions:

tech_has_tech DOCSIS_10

tech_has_tech DOCSIS_20

Figure 9. Screenshot of Enterprise Ontology Manager

After the user clicks the “Search” button, custom search module in our IC management system searches through multiple data sources (the Internet, the Cisco Intranet and Social

Network sites) for the query keywords. The customer support engineer either can manually choose from the search results for next step knowledge mining, or can use the system's selected top k results to extract IC.

Figure 8 presents a snapshot of the IC extractor tool in our system. The IC management system promotes easy management of enterprise knowledge base with the enterprise ontology management user interface. As depicted in Figure 9, the engineers can add, update, and delete the ontology records to better manage the classes and properties through this interface.

4.3. Experiments

4.3.1. Entity Extractor

In our first experiment, we evaluated the performance of the Entity Extraction by comparing the extracted entities with manually tagged entities. We use precision and recall measurements for performance evaluation of our algorithm using the following formulas:

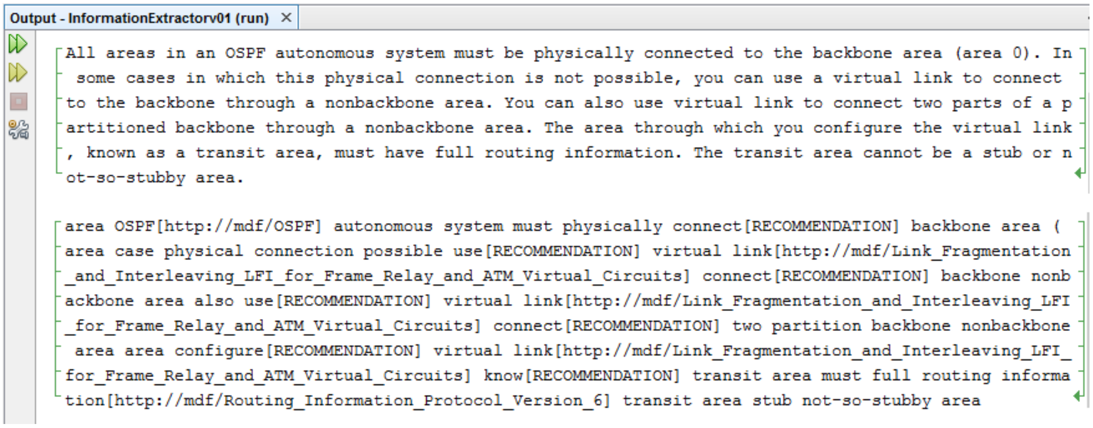
$$precision = \frac{|relevantEntries \cap retrievedEntries|}{retrievedEntries}$$

$$recall = \frac{|relevantEntries \cap retrievedEntries|}{relevantEntries}$$

Table 4. Performance of Entity Extractor

	1-Word	**2-Word**	**3-Word**	**4-Word**
Precision	98.10%	100%	100%	100%
Recall	96.30%	94.50%	94.40%	88.90%

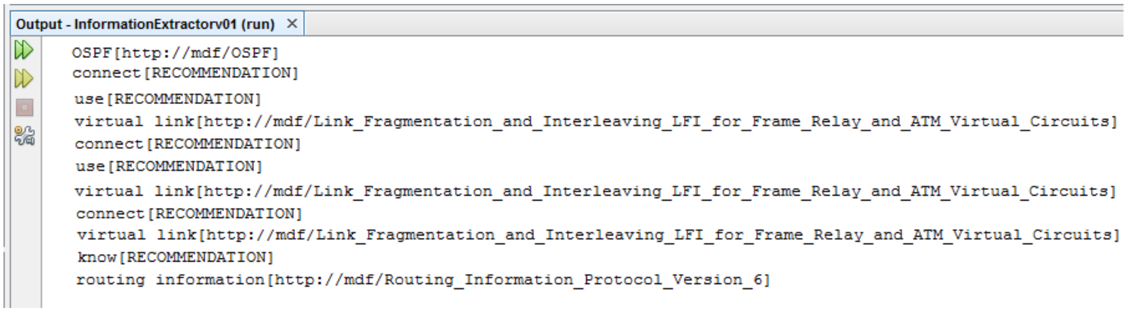
We should note that extracted entities are mapped with the semantic entities are using one-to-many relationships. An extracted entity is considered as a *relevantEntity* if it belongs to one of those mapped entities in the ontology. Our simplified approach of entity extraction exploits domain knowledge in a form of semantic entities and thus generates precise output. It can be seen from Table 4 that precision and recall measurements of the entity extractor are very high for entities with different length. Moreover, we carefully considered both word-level and character-level variations of the semantic entities within the ontology database while designing the extraction algorithm. As a result, the entity extractor in our proposed IC management system effectively captures variations due to word-level insertion, substitution, deletion, permutation, and abbreviations. Figure 10 and 11 represent the entity extractor input-output and the extracted entities with the identified categories for that input sample respectively.



```
All areas in an OSPF autonomous system must be physically connected to the backbone area (area 0). In
 some cases in which this physical connection is not possible, you can use a virtual link to connect
to the backbone through a nonbackbone area. You can also use virtual link to connect two parts of a p
artitioned backbone through a nonbackbone area. The area through which you configure the virtual link
, known as a transit area, must have full routing information. The transit area cannot be a stub or n
ot-so-stubby area.

area OSPF[http://mdf/OSPF] autonomous system must physically connect[RECOMMENDATION] backbone area (
area case physical connection possible use[RECOMMENDATION] virtual link[http://mdf/Link_Fragmentation
_and_Interleaving_LFI_for_Frame_Relay_and_ATM_Virtual_Circuits] connect[RECOMMENDATION] backbone nonb
ackbone area also use[RECOMMENDATION] virtual link[http://mdf/Link_Fragmentation_and_Interleaving_LFI
_for_Frame_Relay_and_ATM_Virtual_Circuits] connect[RECOMMENDATION] two partition backbone nonbackbone
 area area configure[RECOMMENDATION] virtual link[http://mdf/Link_Fragmentation_and_Interleaving_LFI_
for_Frame_Relay_and_ATM_Virtual_Circuits] know[RECOMMENDATION] transit area must full routing informa
tion[http://mdf/Routing_Information_Protocol_Version_6] transit area stub not-so-stubby area
```

Figure 10. Snapshot of Pre-processor Output



```
OSPF[http://mdf/OSPF]
connect[RECOMMENDATION]
use[RECOMMENDATION]
virtual link[http://mdf/Link_Fragmentation_and_Interleaving_LFI_for_Frame_Relay_and_ATM_Virtual_Circuits]
connect[RECOMMENDATION]
use[RECOMMENDATION]
virtual link[http://mdf/Link_Fragmentation_and_Interleaving_LFI_for_Frame_Relay_and_ATM_Virtual_Circuits]
connect[RECOMMENDATION]
virtual link[http://mdf/Link_Fragmentation_and_Interleaving_LFI_for_Frame_Relay_and_ATM_Virtual_Circuits]
know[RECOMMENDATION]
routing information[http://mdf/Routing_Information_Protocol_Version_6]
```

Figure 11. Extracted Entities by Pre-processor Module

4.3.2. Technology Identifier

In our next experiment, we examined the performance of the technology identifier module using a data set of 270 problem descriptions. For entities having multiple technologies, we choose the one at a lower level in the hierarchy to make the output as specific as possible. As compared to the manual process of verifying whether the Technology/Sub-technology metadata field in the Service Request is a correct one, our identifier module categorizes the SR instantly. Figure 12 summarizes the findings for this experiment. It can be clearly seen that our algorithm achieves high precision and recall with a high F-1 measure and thus generates precise output.

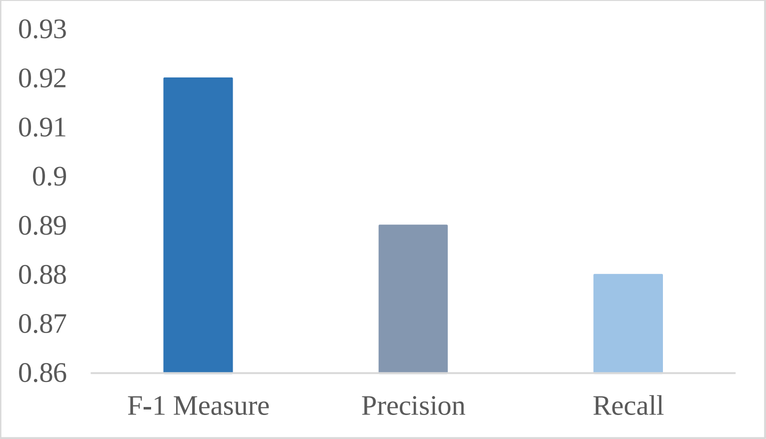


Figure 12. Performance of Technology Identifier

The performance of this module has also been tested in a large scale environment where technology identifier was deployed using MapReduce (Dean 2008). More specifically, we have implemented the system on Hadoop cluster, which gave us a convenient framework for distributed computing of technology/sub-technology identification. We also have a plan to implement and measure the scalability performance of the IC extractor segment on a similar distributed clustered platform.

4.3.3. IC Extractor

We also conducted experiment to evaluate the performance of the classifier in the IC management system. The feature set in our MaxEnt classifier includes extended semantic entities, top words, type of websites, query keywords, length of the paragraph, relative location of the paragraph, and the bag-of-hit words. As mentioned earlier, we considered paragraph as the granular data level for evaluation and each of these features help the classifier determining the class label for a particular paragraph. Each data in training data set is represented as two tab-separated columns where first column indicates the class label of a paragraph and the last column is a comma-separated list of features for that paragraph. The classifier in our IC management system also maintains a file to handle several configuration parameters. These parameters aid in tuning the model by adjusting different variables of the classifier for performance optimization. The variables we have used for model tuning include regularization, convergence tolerance for parameter optimization, smoothing method.

To evaluate the efficiency of our IC mining system, we examined how this toolkit aid in productivity enhancement for Best Practice IC extraction in the domain of enterprise service request resolution, specifically for Cisco Service Request management. The average time needed for manually extracting IC related to a particular topic was compared with the time taken by IC Extractor in our toolkit. In the manual process of Intellectual Capital mining, support engineers input the IC topic as a set of keywords to external search engines and Cisco internal search tools. To further process the documents and/or webpages returned by the search engines, customer support engineers are required to read and comprehend those documents. Once they have a better understanding on those Best Practice IC related problem related documents, the engineers mark (copy/paste) the relevant information as "*criteria*", "*impact*", and "*recommendation*" and finally they can summarize the IC based on the documented information. On the other hand, our IC mining tool assists the support engineers by allowing them to simply enter the keywords in the toolkit and relevant categorized information will be automatically returned to them as output. In addition, for incorrect or incomplete knowledge, the engineers can use our tool to highlight the documents/web pages to correct the returned IC.

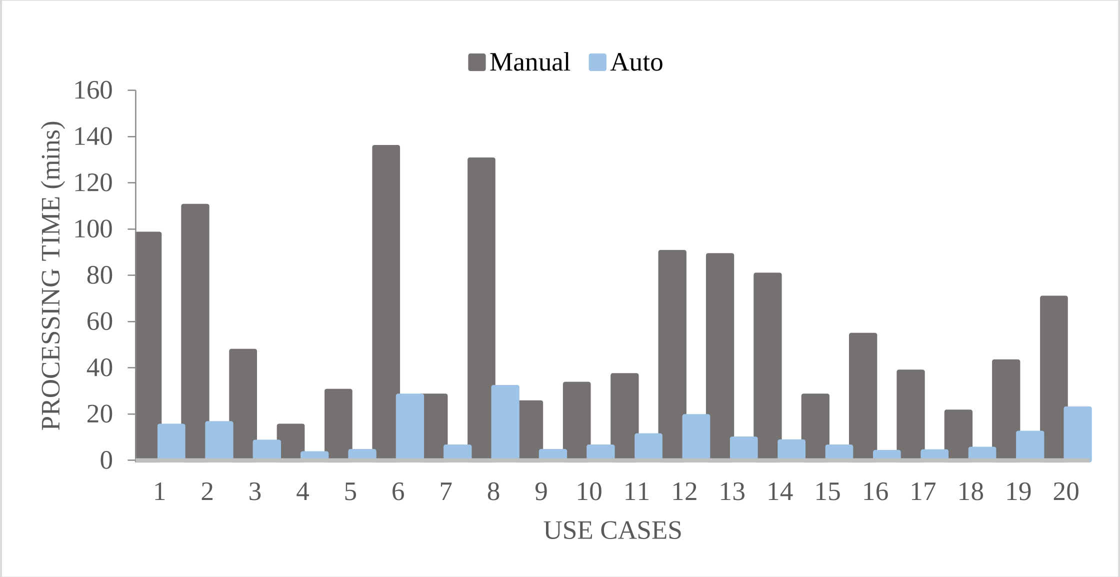


Figure 13. Comparison of Time Consumption of Manual & Auto IC Extraction

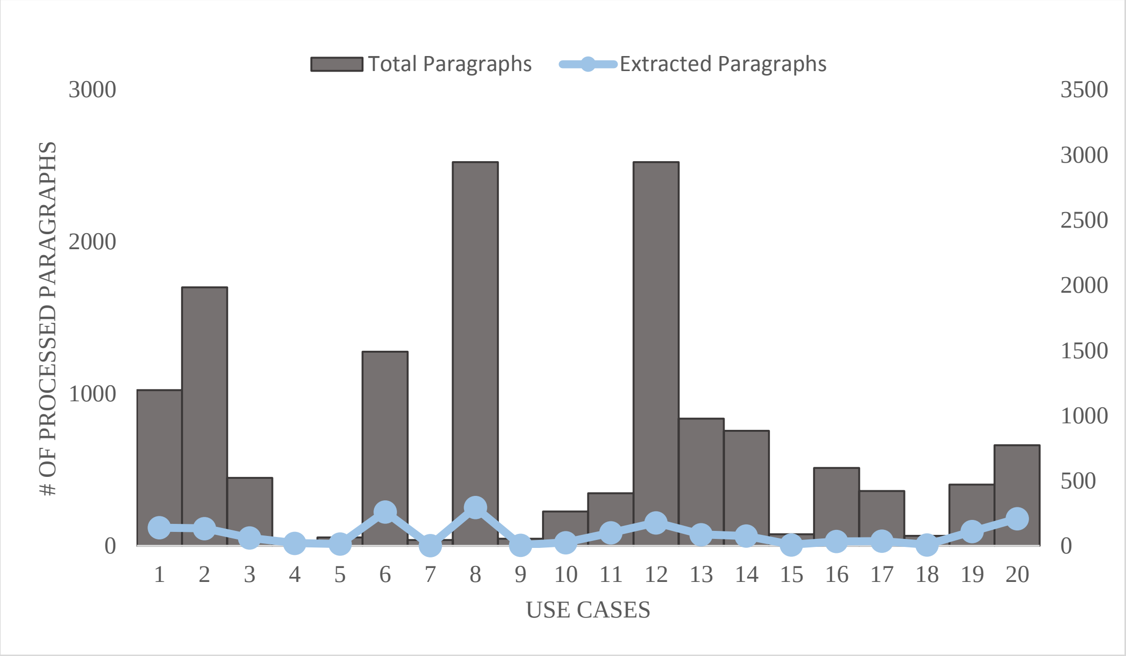
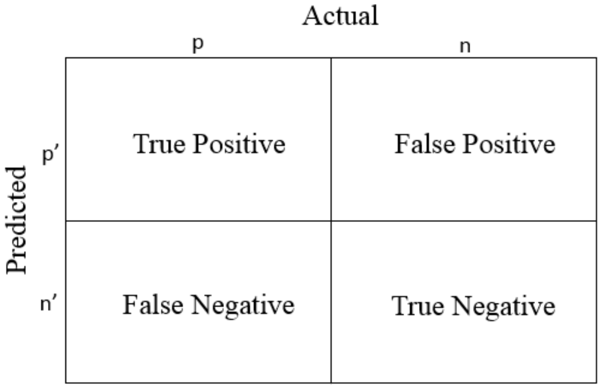


Figure 14. Workload Comparison of Manual & Auto IC Extraction

The effectiveness of the proposed system is observed with a comparison of the average time used for extracting IC related to a particular topic manually and with the assistance of IC mining toolkit. Figure 13 plots the time required for these two scenarios to process the same problem set data including the time required for inaccurate IC correction. It is very obvious that our tool dramatically outperforms manual IC extraction in terms of time reduction.



		Actual	
		p	n
Predicted	p′	True Positive	False Positive
	n′	False Negative	True Negative

Figure 15. Confusion Matrix

In the next phase of performance evaluation, we compared the information load for both the manual and automatic IC mining process and Figure 14 presents our finding. The information load is computed based on the number of paragraphs that needs to be read or processed when working on a request resolution. Like the previous experiment, if the paragraphs returned by the tool are incorrect then all the manually processed paragraphs will be added in the system.

The performance of the IC mining classifier is measured using the aforementioned feature sets with the goal of determining whether inclusion of semantically enhanced entities along with other features actually help improving the performance. The feature set of the IC extractor includes frequent words (W), extended semantic entities (S), presence of search keywords (K), type of the documents (T), length of the paragraph being processed (L), and relative location of that paragraph in the document (R). For measuring the performance, we use macro-averaged F1, accuracy, precision and recall and which are defined as follows:

$$recall = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

$$micro - avg.\ F1 = \frac{TP}{num}$$

$$macro - avg. F1 = \frac{2 \times recall \times precision}{recall + precision}$$

The confusion matrix in Figure 15 was used for the calculation of these measurements based on the two possible outcomes – positive (p: the result is present as expected) and negative (n: the result is not present). To conduct this experiment, we have performed 10 folds cross validation on the dataset. Moreover, we shuffled the dataset and took the average of 10 different runs before subdividing them for cross-validation. We attained similar results for each of these folds which clearly indicates the stability of the score performed by the system.

Table 5. Performance Measurement for Different Feature Set – 10 Folds Cross Validation (Maximum Entropy Classifier)

Feature Set	**Micro-avg. F1**	**Macro-avg. F1**	**Accuracy**	**Precision**	**Recall**
W	0.74	0.61	0.76	0.66	0.58
W+S	0.80	0.71	0.80	0.73	0.68
W+S+K	0.81	0.72	0.81	0.73	0.70
W+S+K+T+L+R	0.82	0.73	0.82	0.74	0.72

A summary of these performance measurements for the above mentioned feature sets can be found in Table 5. The findings in this table clearly indicates that semantic entities dramatically improves performance and the combined feature set performs best as compared to others.

Table 6. Performance Measurement for Different Feature Set – 10 Folds Cross Validation (Naïve Bayes Classifier)

Feature Set	Micro-avg. F1	Macro-avg. F1	Accuracy	Precision	Recall
W	0.45	0.43	0.42	0.43	0.42
W+S	0.33	0.41	0.32	0.40	0.44
W+S+K	0.34	0.50	0.35	0.50	0.50
W+S+K+T+L+R	0.69	0.64	0.69	0.80	0.53

We also have tested our IC mining toolkit for Naïve Bayes classifier and the outcomes are listed in Table 6. However, for Naïve Bayesian we see that the performance is lower as compared to Maximum Entropy classifier. Also, we can see a performance degradation after the inclusion of semantic entities. But we got a better measurement for Macro-averaged F1, precision and recall after adding the feature 'presence of search keywords'.

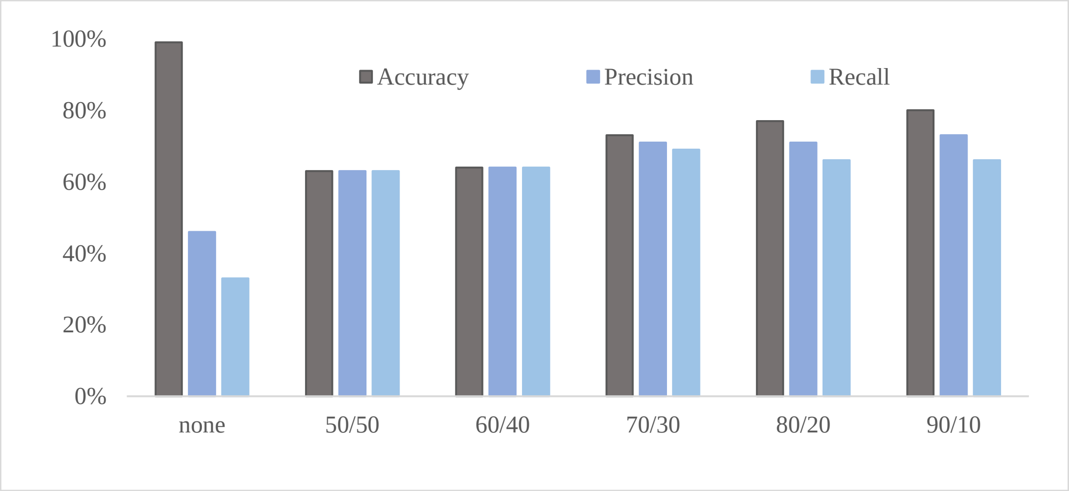


Figure 16. IC Extraction with Different Under-sampling Rate

The next experiment, we executed is for measuring the performance of under-sampling the imbalanced data and the results are presented in Figure 17 which demonstrates system performance under different ratio of IC relevant and irrelevant data samples. As we can see, the first set of data, which has been labeled as *none*, denotes the samples without under-sampling, has a very high accuracy but with the cost of low precision and recall. The underlying reason is that the classifier try to minimize the global error and classify more instances as IC-irrelevant which is the majority class in our case. However, in situation like ours, the precision and recall rate are more important than the accuracy as identifying irrelevant data as relevant is more acceptable than recognizing relevant data as irrelevant. In later part of this experiment, we under-sampled the majority IC-irrelevant paragraphs. It can be seen that although the accuracy decreases, the system attains a higher precision and recall after under-sampling was applied on irrelevant data.

4.3.4. IC Repository

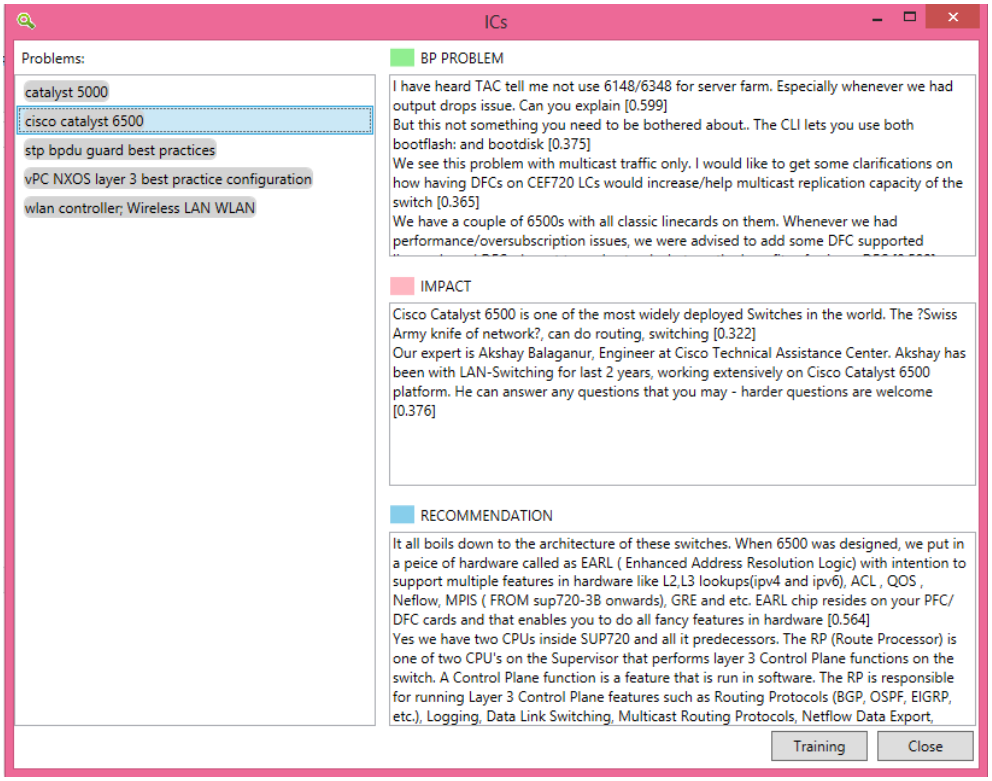
Our IC management module is equipped with an IC repository which not only stores already resolved service requests but also facilitates the search for similar Best Practice problem resolutions. Figure 17 represents a snapshot of this repository in our system. For evaluating the performance of this module, we compared the Silhouette Coefficient (Rousseeuw 1987) value for K-means algorithm, for unstructured data and extracted entities. We have used the description section of best practice ICs for this experiment and Figure 18 plots the finding for this evaluation. It should be noted that a higher score indicates a model with better defined clusters. Silhouette Coefficient (SC) uses the model itself to perform the evaluation for each sample using the following two scores:

d_1: mean distance between a sample and all other points in the class.

d_2: mean distance between a sample and all other points in the next nearest cluster.

The Silhouette Coefficient for a set of samples is calculated as the mean of the SC for each sample where SC for a single sample is defined as:

$$sc = \frac{d_2 - d_1}{\max(d_1, d_2)}$$



Problems:

- catalyst 5000
- cisco catalyst 6500
- stp bpdu guard best practices
- vPC NXOS layer 3 best practice configuration
- wlan controller; Wireless LAN WLAN

🟩 BP PROBLEM

I have heard TAC tell me not use 6148/6348 for server farm. Especially whenever we had output drops issue. Can you explain [0.599]
But this not something you need to be bothered about.. The CLI lets you use both bootflash: and bootdisk [0.375]
We see this problem with multicast traffic only. I would like to get some clarifications on how having DFCs on CEF720 LCs would increase/help multicast replication capacity of the switch [0.365]
We have a couple of 6500s with all classic linecards on them. Whenever we had performance/oversubscription issues, we were advised to add some DFC supported

🟥 IMPACT

Cisco Catalyst 6500 is one of the most widely deployed Switches in the world. The ?Swiss Army knife of network?, can do routing, switching [0.322]
Our expert is Akshay Balaganur, Engineer at Cisco Technical Assistance Center. Akshay has been with LAN-Switching for last 2 years, working extensively on Cisco Catalyst 6500 platform. He can answer any questions that you may - harder questions are welcome [0.376]

🟦 RECOMMENDATION

It all boils down to the architecture of these switches. When 6500 was designed, we put in a peice of hardware called as EARL (Enhanced Address Resolution Logic) with intention to support multiple features in hardware like L2,L3 lookups(ipv4 and ipv6), ACL , QOS , Neflow, MPIS (FROM sup720-3B onwards), GRE and etc. EARL chip resides on your PFC/ DFC cards and that enables you to do all fancy features in hardware [0.564]
Yes we have two CPUs inside SUP720 and all it predecessors. The RP (Route Processor) is one of two CPU's on the Supervisor that performs layer 3 Control Plane functions on the switch. A Control Plane function is a feature that is run in software. The RP is responsible for running Layer 3 Control Plane features such as Routing Protocols (BGP, OSPF, EIGRP, etc.), Logging, Data Link Switching, Multicast Routing Protocols, Netflow Data Export,

Figure 17. IC Repository

4.3.5. IC Recommender

IC recommender facilitates the search for finding ICs with similar problem description. The experiment dataset includes the same dataset that was used in the performance measurement of technology identifier. To find out the closeness of two problem request data, we have considered semantic similarity calculation described in section 2.4.5. The finding, which has been plotted on Figure 19, shows that our simple approach of recommendation using semantic concept distance outperforms the traditional KNN approach in terms of standard Mean Absolute Error (MAE). The lower MAE values are the higher is the recommendation accuracy. Given the datasets of actual and predicted values *(a, p)* for all the *n* problem descriptions in the test set, the MAE is computed as follows:

$$MAE = \frac{\sum_{i=1}^{n} |a - p|}{n}$$

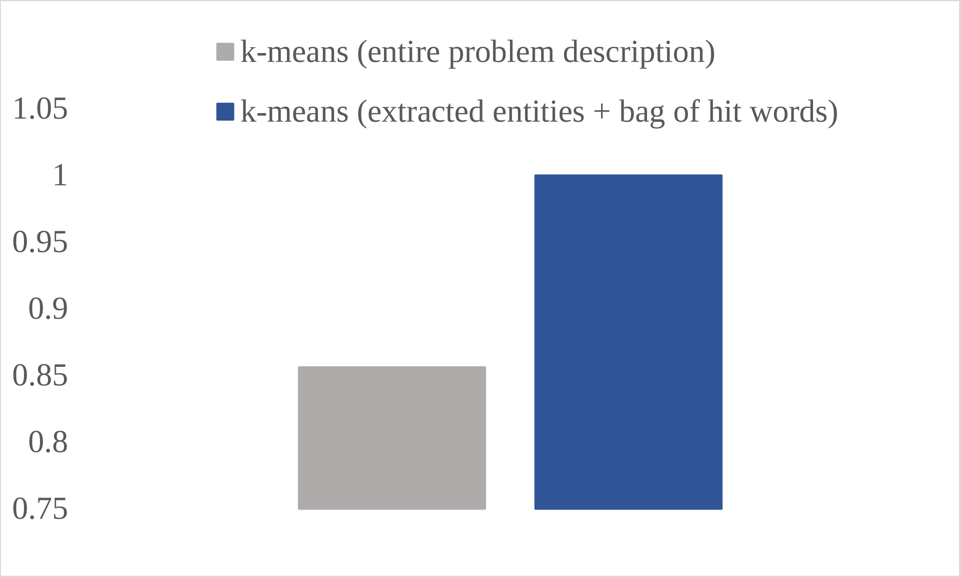


Figure 18. Comparison of Silhouette Coefficients

It can be noted that though for this experiment, we have only used the problem description segment of the intellectual capital. However, the other related information, such as the impacts, recommendations can easily be tracked from the IC repository if the support engineers are interested in exploring how those similar ICs were resolved.

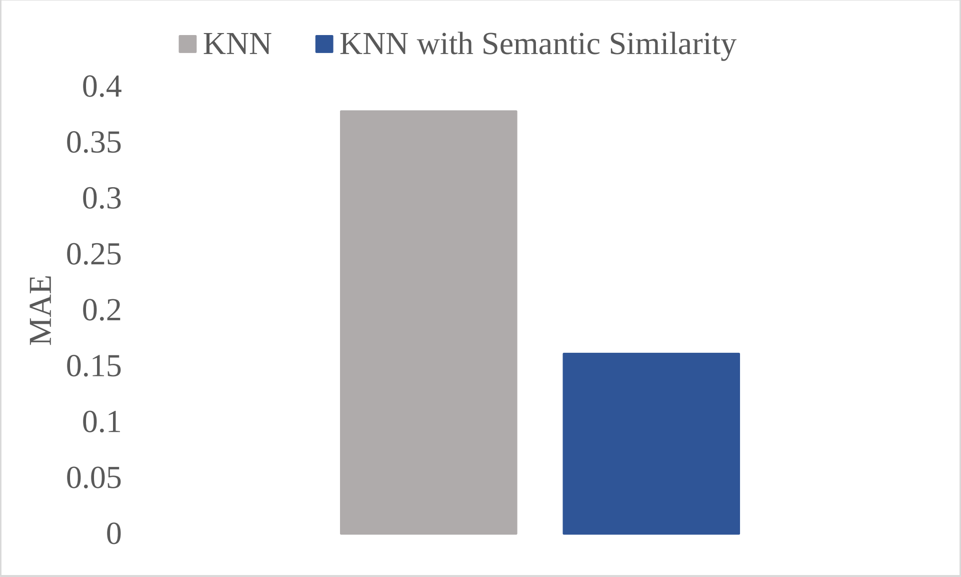


Figure 19. Performance Comparison of IC Recommender

5. CONCLUSION

In this dissertation, we study knowledge discovery and management methods for enterprise service centers. The issues with existing knowledge management systems in this domain and our contributions to address those challenges have been summarized in this chapter with the hope that this will inspire new approaches and enlighten other researchers in better understanding the challenges in knowledge discovery and management in the context of service centers.

5.1. Contribution Summary

In order to provide a quality service to the customers, knowledge discovery and management is essential for enterprise service centers. A knowledge enabled infrastructure allows customer service centers to measure their customer support performance through automation of Intellectual Capital (IC) acquisition and automation. Along with expedite and efficient request resolution methodology, identification of an organization’s IC also provides insights on management action. These insights often relate to organizations’ goal of enhancing their transparency which is considered beneficial for both internal and external stakeholders as well as for beneficiaries. Traditional approaches of IC mining include manual methods like interviews, surveys, workshops which rely comprehensively on human participation and thus are time consuming and costly.

To assist enterprise customer centers resolving the service requests and accelerate the time need to resolve cases using online and in-house data, we propose an efficient Intellectual Capital management system. Our proposed system converts enormous amount of semi-structured and unstructured data into reusable knowledge or Intellectual Capital. We model the service resolution problem as a combination of five different tasks- categorizing problem description, online search for related documents, classification of IC sections, clustering similar service requests, and recommending previously solved request resolutions for new service requests based on their similarity. The knowledge base in our IC management system is data collected from enterprise data repositories and the Internet using a custom search module. Once a support engineer is assigned to a service request based on the problem domain category, he searches for documents related to customer’s service request in this knowledgebase. After that, the search results are pre-processed and classified to extract IC which includes the problem definition, possible impacts, and the recommended resolution steps to solve that problem. A novel classifier has been used for extracting IC from the mountain of data, which utilized the enterprise domain ontology to direct the classification process. Experimental results in Chapter 4 show that our semantics-assisted classifier dramatically enhances the system performance. In addition to enhanced specification and matching techniques, this proposed model offers improved method for service request resolution data categorization. The proposed Intellectual Capital Mining system achieves a precise result for identifying problem category, extracting, classifying different IC categories, and clustering and recommending similar service problems. A plugin based on the proposed strategy has been used in real enterprise service centers which efficiently improves the service engineer’s request resolution performance and intensifies the amount of reusable knowledge. Our proposed system is also equipped with an IC repository which helps grouping similar service requests and promotes knowledge reusability. The IC recommender module uses this repository to while recommending previously answered similar service requests.

5.2. Future Directions

In our research we tried to focus on the knowledge extraction and reuse in the form of Intellectual Capital as this is very crucial for enterprise customer service centers in terms of reputation and business competitiveness. However, this is merely a fragment in the comprehensive area of knowledge mining in service centers context. We briefly listed some of these interesting research directions which we believe, if solved, will help improving the performance of request resolution process further:

i) Utilizing enterprise service requests repository, how can we extract the information about the customer support engineer's level of expertise on solving service requests on specific domain. This information is very useful when building tools to measure employee performance.

ii) Significant research should be conducted on how topic modeling (Blei 2012) can be incorporated in the research of intellectual capital mining.

iii) Performance of IC management system can be improved further with the assimilation of external ontology. For instance, Cisco IOS software routers implement Maintenance Operations Protocol (MOP), developed by Digital Equipment Incorporation, to collect configuration information. However, Cisco enterprise ontology lacks this information and as a result the entity extractor in our IC management systems cannot measure the semantic relationship for those missing terms.

iv) In future, like the technology identifier, we also plan to include the parallelization of the IC extraction procedure to help expedite the computation process.

6. REFERENCES

A. Hotho, S. Staab, and G. Stumme. 2003. "Wordnet improves text document clustering." *Proceedings of the Workshop on Semantic Web, SIGIR-2003.* Toronto, Canada.

Abeysekera, Indra. 2006. "The project of intellectual capital disclosure: researching the research." *Journal of intellectual capital 7.1* 61-77.

Adler, Paul S. 1989. "When knowledge is the critical resource, knowledge management is the critical task." *Engineering Management, IEEE Transactions on 36, no. 2* 87-94.

Ahmed, Khalida Bensidi, Adil Toumouh, and Dominic Widdows. 2014. "Lightweight domain ontology learning from texts: graph theory–based approach using Wikipedia." *International Journal of Metadata, Semantics and Ontologies 9, no. 2* 83-90.

Alavi, Maryam, and Dorothy E. Leidner. 2001. "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues." *MIS quarterly,* 107-136.

Ashu, Roy. 2012. *Business Value of Contact Center Knowledge Management: A Strategic Perspective.* eGain Communications.

Babych, Bogdan, and Anthony Hartley. 2008. "Improving machine translation quality with automatic named entity recognition." *In Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT.* Association for Computational Linguistics. 1-8.

Banerjee, Arindam. 2007. "An Analysis of Logistic Models: Exponential Family Connections and Online Performance." SDM.

Baumer, Eric PS, Jordan Sinclair, and Bill Tomlinson. 2010. "America is like Metamucil: fostering critical and creative thinking about metaphor in political blogs." *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM. 1437-1446.

Bawakid, Abdullah, and Mourad Oussalah. 2010. "A semantic-based text classification system." *Cybernetic Intelligent Systems (CIS), 2010 IEEE 9th International Conference on.* IEEE.

Beattie, Vivien, and Sarah Jane Thomson. 2007. "Lifting the lid on the use of content analysis to investigate intellectual capital disclosures." In *Accounting Forum. Vol. 31. No. 2*. Elsevier.

Berger, Adam. 2005. "The improved iterative scaling algorithm: A gentle introduction." citeseer.ist.psu.edu/berger97improved.html.

Bikel, Daniel M., Richard Schwartz, and Ralph M. Weischedel. 1999. "An algorithm that learns what's in a name." *Machine learning 34, no. 1-3* 211-231.

Blei, David M. 2012. "Probabilistic topic models." *Communications of the ACM 55.4* 77-84.

Bloehdorn, Stephan, and Andreas Hotho. 2004. "Boosting for text classification with semantic features." *WebKDD* 149-166.

Bontis, Nick. 2003. "Intellectual capital disclosure in Canadian corporations." *Journal of Human Resource Costing & Accounting 7.1* 9-20.

Brants, T. 2000. "A statistical Part-of-Speech tagger." *In Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP-2000).* 224-231.

Brill, Eric. 1992. "A simple rule-based part of speech tagger." *In Proceedings of the workshop on Speech and Natural Language.* Association for Computational Linguistics. 112-116.

Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research 16, no. 1* 321-357.

Chen, W. & Wang, M. 2009. "A fuzzy c-means clustering-based fragile watermarking scheme for image authentication." *Expert Systems with Applications, 36(2)* 1300-1307.

Cheng, Ching Kang, Xiaoshan Pan, and Franz Kurfess. 2004. "Ontology-based semantic classification of unstructured documents." *Adaptive Multimedia Retrieval.* Springer Berlin Heidelberg. 120-131.

Cheung, Chi Fai, W. B. Lee, and Y. Wang. 2005. "A multi-facet taxonomy system with applications in unstructured knowledge management." *Journal of knowledge management 9.6* 76-91.

Chieu, Hai Leong, and Hwee Tou Ng. 2002. "Named Entity Recognition: A Maximum Entropy Approach Using Global Information." *COLING '02 Proceedings of the 19th International Conference on Computational Linguistics, vol. 1.* Stroudsburg, PA. 1-7.

Daelemans, Walter, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. "MBT: A Memory-Based." *arXiv preprint cmp-lg/9607012* .

Darroch, John N., and Douglas Ratcliff. 1970. "Generalized iterative scaling for log-linear models." *The annals of mathematical statistics*, 1470-1480.

Davenport, Thomas H., and Laurence Prusak. 1998. *Working knowledge: How organizations manage what they know.* Harvard Business Press.

Dean, Jeffrey, and Sanjay Ghemawat. 2008. "MapReduce: simplified data processing on large clusters." *Communications of the ACM 51, no. 1* 107-113.

Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. "Indexing by latent semantic analysis." *Journal of the American society for information science 41, no. 6, 391.*

Dimension Data. 2013-2014. "Global Contact Centre Benchmarking Report." http://www.dimensiondata.com/Global/Global-Microsites/CCBenchmarking.

Domingos, Pedro, and Michael Pazzani. 1997. "On the optimality of the simple Bayesian classifier under zero-one loss." *Machine learning 29, no. 2-3* 103-130.

Dow Jones. 2016. *Factiva.* Accessed April 8, 2016. https://global.factiva.com/factivalogin/login.asp?productname=global.

Dumais, Susan T. 1991. "Improving the Retrieval of Information from External Sources." *Behavior Research Methods, Instruments, & Computers 23, no. 2* 229-236.

Eikvil, K. Aas and L. 1999. *Text Categorisation: A Survey.* Technical Report Raport NR 941, Norwegian Computing Center.

Elder IV, John, and Thomas Hill. 2012. *Practical text mining and statistical analysis for non-structured text data applications.* Academic Press.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. "A density-based algorithm for discovering clusters in large spatial databases with noise." *Kdd, vol. 96, no. 34* 226-231.

Fan, Jing, Xiuying Liu, Ying Shen and Tianyang Dong. 2012. "Ontology-based Knowledge Management for Forest Channel‖." *In Proc. 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012).* IEEE. 1523-1527.

Feldman, Susan. 2004. "The high cost of not finding information." *Information Today, Incorporated.*

Fensel, Dieter. 2001. "Ontologies." *Springer Berlin Heidelberg.*

Fernández, Alberto, Salvador García, María José del Jesus, and Francisco Herrera. 2008. "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets." *Fuzzy Sets and Systems 159, no. 18* 2378-2398.

Fernández, Javier D., Claudio Gutierrez, and Miguel A. Martínez-Prieto. 2010. "RDF Compression: Basic Approach." *In Proceedings of the 19th international conference on World wide web.* ACM. 1091-1092.

Florian, Radu, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. "Named Entity Recognition Through Classifier Combination." *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4.* Association for Computational Linguistics. 168-171.

Funayama, Hirotaka, Tomohide Shibata, and Sadao Kurohashi. 2009. "Bottom-Up Named Entity Recognition Using a Two-Stage Machine Learning Method." *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications.* Association for Computational Linguistics. 55-62.

Gayathri, K., and A. Marimuthu. 2013. "Text document pre-processing with the KNN for classification using the SVM." *In Intelligent Systems and Control (ISCO), 2013 7th International Conference on.* IEEE. 453-457.

Glimm, Birte, Ian Horrocks, Boris Motik, Rob Shearer, and Giorgos Stoilos. 2012. "A Novel Approach to Ontology Classification." *Web Semantics: Science, Services and Agents on the World Wide Web 14* 84-101.

Gomez-Perez, Asuncion, Mariano Fernández-López, and Oscar Corcho. 2006. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web.* Springer Science & Business Media.

Grant, Robert M. 1996. "Toward a knowledge-based theory of the firm." *Strategic management journal 17.S2* 109-122.

Guber, T. 1993. "A Translational Approach to Portable Ontologies." *Knowledge Acquisition 5, no. 2* 199-229.

Guthrie, J., R. Petty, F. Ferrier, and R. Wells. 1999. "There is no accounting for intellectual capital in Australia: Review of annual reporting practices and the internal measurement of intangibles within Australian organisations." *In International Symposium Measuring and Reporting Intellectual Capital: Experiences, Issues and Prospects* 9-10.

Guthrie, James, Richard Petty, Kittiya Yongvanich, and Federica Ricceri. 2004. "Using content analysis as a research method to inquire into intellectual capital reporting." *Journal of intellectual capital 5, no. 2* 282-293.

Hackbarth, Gary. 1998. "The impact of organizational memory on IT systems." *AMCIS 1998 Proceedings.* 197.

Hayes, Robert M. 1963. "Mathematical models in information retrieval." In *Natural Language and the Computer (Edited by PL Garvin)*. New York 287: McGraw-Hill.

Heitz, Christoph, Geoffrey Ryder, and Kevin Ross. 2008. "Knowledge Management in Call Centers: How Routing Rules Influence Expertise and Service Quality." *In MSOM Conference Proceedings.* Washington DC: MSOM. 1-7.

Ho, Anh Khoi Ngo, Nicolas Ragot, Jean-Yves Ramel, Véronique Eglin, and Nicolas Sidere. 2013. "Document Classification in a non-stationary environment: A One-Class SVM

Approach." *In Document Analysis and Recognition (ICDAR), 2013 12th International Conference on.* IEEE. 616-620.

Hotho, Andreas, Alexander Maedche, and Steffen Staab. 2002. "Ontology-based text document clustering." *KI 16, no. 4* 48-54.

Hsieh, Shang-Hsien, Hsien-Tang Lin, Nai-Wen Chi, Kuang-Wu Chou, and Ken-Yu Lin. 2011. "Enabling the Development of Base Domain Ontology through Extraction of Knowledge from Engineering Domain Handbooks." *Advanced Engineering Informatics 25, no. 2* 288-296.

Hu, Fanghuai, Zhiqing Shao, and Tong Ruan. 2014. "Self-Supervised Chinese Ontology Learning from Online Encyclopedias." *The Scientific World Journal.*

Huang, Anna. 2008. "Similarity measures for text document clustering." *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008).* Christchurch, New Zealand.

Huang, Jingshan, Dejing Dou, Lei He, Jiangbo Dang, Hayes, P. 2010. "Ontology-based knowledge discovery and sharing in bio-informatics and medical informatics: A brief survey." *Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD).*

Jain, Anil K. 2010. "Data clustering: 50 years beyond k-means." *Pattern recognition letters 31, no. 8* 651-666.

Joachims, Thorsten. 1997. "A Probabilistic Analysis of the Rocchio Algorithm with tfidf for Text Categorization." *Proc. 14th Int'l Conf. Machine Learning (ICML '97).* 143-151.

Jones, K. Sparck, Steve Walker, and Stephen E. Robertson. 2000. "A probabilistic model of information retrieval: development and comparative experiments: Part 2." *Information Processing & Management 36, no. 6* 809-840.

Kakabadse, Nada K., Andrew Kakabadse, and Alexander KouzminJournal of knowledge management 7, no. 4. 2003. "Reviewing the knowledge management literature: towards a taxonomy." *Journal of knowledge management 7, no. 4* 75-91.

Kano, Y., W. A. Baumgartner, L. McCrohon, S. Ananiadou, K. B. Cohen, L. Hunter, and T. Tsujii. 2009. "Data Mining: Concept and Techniques." *Oxford Journal of Bioinformatics 25, no. 15* 1997-1998.

Kavitha, V., and M. Punithavalli. 2010. "Clustering time series data stream - A literature survey." *International Journal of Computer Science and Information Security, 8(1)* 289-294.

Khalid, Mahboob Alam, Valentin Jijkoun, and Maarten De Rijke. 2008. "The impact of named entity normalization on information retrieval for question answering." 705-710. Springer Berlin Heidelberg.

Kiss, Tibor, and Jan Strunk. 2006. "Unsupervised multilingual sentence boundary detection." *Computational Linguistics 32, no. 4* 485-525.

Klieber, Werner, Vedran Sabol, Markus Muhr, Roman Kern, Georg Öttl, and Michael Granitzer. 2009. "Knowledge discovery using the KnowMiner framework." *Proc. IADIS 9.*

Kuehnast J, and Hengeveld W. 2009. "Enterprise application integration (white paper)." GmbH, Berlin: T-systems enterprise services.

Labrou, Yannis and Tim Finin. 1999. "Yahoo! as an ontology: using Yahoo! categories to describe documents." *Proceedings of the eighth international conference on Information and Knowledge Management.* Kansas City.

Lan, Man, Chew Lim Tan, Jian Su, and Yue Lu. 2009. "upervised and traditional term weighting methods for automatic text categorization." *Pattern Analysis and Machine Intelligence, IEEE Transactions on 31, no. 4* 721-735.

Leonard-Barton, Dorothy. 1995. *Wellspring of knowledge.* Boston, MA: Harvard Business School Press.

Li, Xiaoli, and Bing Liu. 2003. "Learning to Classify Texts Using Positive and Unlabeled Data." *In IJCAI, vol. 3* 587-592.

Lin, Dekang. 1999. "MINIPAR: a minimalist parser." *Maryland linguistics colloquium.*

Liu, Fasheng, and Lu Xiong. 2011. "Survey on text clustering algorithm." *Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on.* IEEE. 901-904.

Liu, Xu-Ying, Jianxin Wu, and Zhi-Hua Zhou. 2009. "Exploratory undersampling for class-imbalance learning." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 39.2* 539-550.

Lock Lee, Laurence, and James Guthrie. 2010. "Visualising and measuring intellectual capital in capital markets: a research method." *Journal of intellectual capital 11.1* 4-22.

Lu, Wen-Min, Wei-Kang Wang, and Qian Long Kweh. 2014. "Intellectual capital and performance in the Chinese life insurance industry." *Omega 42.1* 65-74.

Luca, De, Ernesto William, Andreas Nürnberger, and O. von-Guericke. 2004. "Ontology-based semantic online classification of documents: Supporting users in searching the web." *Proc. of the European Symposium on Intelligent Technologies (EU-NITE 2004).* Aachen.

Luger, George F. 2005. *Artificial Intelligence: Structure and Strategies for Complex Problem.* Pearson education.

MacQueen, James. 1967. "Some methods for classification and analysis of multivariate observations." *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, no. 14.* 281-297.

Magrassi, Paolo. 2002. *A taxonomy of Intellectual capital.* Wikimedia Foundation Inc.

Malouf, Robert. 2002. "A comparison of algorithms for maximum entropy parameter estimation." *In proceedings of the 6th conference on Natural language learning-Volume 20.* Association for Computational Linguistics.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval.* Cambridge: Cambridge university press.

Marcus, Mitchell, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. *Treebank-3.* Accessed April 5, 2016. https://catalog.ldc.upenn.edu/LDC99T42.

Marinica, Claudia, and Fabrice Guillet. 2010. "Knowledge-based interactive postmining of association rules using ontologies." *Knowledge and Data Engineering, IEEE Transactions on 22, no. 6* 784-797.

Marr, Bernard, and Giovanni Schiuma. 2001. "Measuring and managing intellectual capital and knowledge assets in new economy organisations." In *Handbook of performance measurement*. Gee, London.

McCallum, Andrew, and Wei Li. 2003. "Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction, and Web-Enhanced Lexicons." *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4.* Association for Computational Linguistics. 188-191.

McEvily, Susan K., Shobha Das, and Kevin McCabe. 2000. "Avoiding competence substitution through knowledge sharing." *Academy of Management Review 25, no. 2* 294-311.

McGuinness, Deborah L., and Frank Van Harmelen. 2004. "OWL web ontology language overview." *W3C recommendation 10, no. 10.*

Mikheev, Andrei. 2000. "Tagging sentence boundaries." *In Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference.* Association for Computational Linguistics. 264-271.

Monz, Christof, Jaap Kamps, and Maarten de Rijke. 2002. "The University of Amsterdam at CLEF 2002." *In CLEF (Working Notes).*

Muhammad Rafi, M. Shahid Shaikh, Amir Farooq. 2010. "Document Clustering based on Topic Maps." *International Journal of Computer Applications (0975 – 8887), Vol. 12– No.1.*

Narayanan, Vivek, Ishan Arora, and Arjun Bhatia. 2013. "Fast and Accurate Sentiment Classification Using Enhanced Naïve Bayes Model." 194-201. Springer Berlin Heidelberg.

Navathe, Shamkant B., and Elmasri Ramez. 2000. "Data warehousing and data mining." *Fundamentals of Database Systems* 841-872.

Nelson, Richard R., and Sidney G. Winter. 2009. *An evolutionary theory of economic change.* Harvard University Press.

Ng, Andrew Y. 2004. "Feature selection, L1 vs. L2 regularization, and rotational invariance." *In Proceedings of the twenty-first international conference on Machine learning.* ACM. 78.

Nolan Norton, Institute. 1998. *Putting the Knowing Organization to Value.* Nolan Norton Institute.

Nonaka, Ikujiro. 1994. "A dynamic theory of organizational knowledge creation." *Organization science 5, no. 1* 14-37.

Nonaka, Ikujiro, Georg Von Krogh, and Sven Voelpel. 2006. "Organizational knowledge creation theory: Evolutionary paths and future advances." *Organization studies 27, no. 8* 1179-1208.

Pang, Bo, and Lillian Lee. 2008. "Opinion mining and sentiment analysis." *Foundations and trends in information retrieval 2, no. 1-2* 1-135.

Phusavat, Kongkiti, Narongsak Comepa, Agnieszka Sitko-Lutek, and Keng-Boon Ooi. 2013. "Productivity management: integrating the intellectual capital." *Industrial Management & Data Systems 113, no. 6* 840-855.

Pianta, Emanuele, Luisa Bentivogli, and Christian Girardi. 2002. "Multiwoednet: Developing an aligned multilingual database." *In Proc. 1st Int'l Conference on Global WordNet.*

Piateski, Gregory, and William Frawley. 1991. *Knowledge discovery in databases.* MIT press.

Prahalad, Coimbatore K., and Gary Hamel. 2006. *The core competence of the corporation.* Springer Berlin Heidelberg.

Rasooli, Pooya, and Amir Albadvi. 2007. "Knowledge Management in Call Centres." *Electronic Journal of Knowledge Management 5, no. 3* 323-332.

Ratnaparkhi, A. 1996. "A Maximum Entropy Model for Part-of-Speech Tagging." *In Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP-96.* Philadelphia, PA.

Ratnaparkhi, Adwait. 1998. "Maximum entropy models for natural language ambiguity resolution." *PhD diss.* University of Pennsylvania.

Ravichandran, Thiagarajan, and Arun Rai. 1999. "Total quality management in information systems development: key constructs and relationships." *Journal of Management Information Systems 16, no. 3* 119-155.

Richman, Alexander E., and Patrick Schone. 2008. "Mining Wiki Resources for Multilingual Named Entity Recognition." *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies.* Stroudsburg, PA. 1-9.

Ritter, Alan, Sam Clark, and Oren Etzioni. 2011. "Named Entity Recognition in Tweets: An Experimental Study." *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics. 524-1534.

Rong, Guo, and Wu Jun. 2012. "Design and implementation of domain ontology-based oilfield non-metallic pipe infor-mation retrieval system." *Computer Science and Information Processing (CSIP), 2012 International Conference on. IEEE.*

Rousseeuw, Peter J. 1987. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." *Journal of computational and applied mathematics 20* 53-65.

Salton, Gerard, and Michael J. McGill. 1986. *Introduction to modern information retrieval.*

Sambamurthy, V., and Mani Subramani. 2005. "Special issue on information technologies and knowledge management." *MIS quarterly 29, no. 1*, 1-7.

Sangodiah, Anbuselvan, and Lim Ean Heng. 2012. "Integration of data quality component in an ontology based knowledge management approach for e-learning system." *In Computer & Information Science (ICCIS), 2012 International Conference on, vol. 1.* IEEE. 105-108.

Senellart, Pierre, and Vincent D. Blondel. 2008. "Automatic discovery of similarwords." 25-44. Springer London.

Serra, Ivo, Rosario Girardi, and Paulo Novais. 2014. "Evaluating Techniques for Learning Non-Taxonomic Relationships of Ontologies from Text." *Expert Systems with Applications 41, no. 11* 5201-5211.

Shilakes, Christopher C., and Julie Tylman. 1998. *Enterprise information portals.* Merrill Lynch, November 16.

Sorensen, L. 2009. "User managed trust in social networking-Comparing Facebook, MySpace and Linkedin." *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace&Electronic Systems Technology.*

Spender, J-C. 1996. "Making knowledge the basis of a dynamic theory of the firm." *Strategic management journal 17.S2* 45-62.

Subhashini, R., and J. Akilandeswari. 2011. "A Survey on Ontology Construction Methodologies." *International Journal of Enterprise Computing and Business Systems 1, no. 1* 60-72.

Suganya. S, Gomathi. C and Mano Chitra. S. 2013. "Syntax and Semantics based Efficient Text Classification Framework." *International Journal of Computer Applications 65(15):18-21.*

Tang, Min, Bryan Pellom, and Kadri Hacioglu. 2003. "Call-type classification and unsupervised training for the call center domain." *Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on* (IEEE) 204-208.

Toral, Antonio, Elisa Noguera, Fernando Llopis, and Rafael Munoz. 2005. "Improving question answering using named entity recognition." In *Natural language processing and information systems*, 181-191. Springer Berlin Heidelberg.

Torisawa, Jun'ichi Kazama and Kentaro. 2007. "Exploiting Wikipedia as External Knowledge for Named Entity Recognition." *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* 698-707.

Turney, Peter D., and Patrick Pantel. 2010. "From Frequency to Meaning: Vector Space Models of Semantics." *Journal of artificial intelligence research 37, no. 1* 141-188.

Uszok, Andrzej, Larry Bunch, Jeffry M. Bradshaw, Thomas Reichherzer, James Hanna and Albert Frantz. 2013. "Knowledge-Based Approaches to Information Management in Coalition Environments." *Intelligent Systems, IEEE Vol. 28, Issue 1* 34-41.

Vapnik, Vladimir. 2013. "The nature of statistical learning theory." *Springer Science & Business Media.*

Varun Grover, Thomas H. Davenport. 2001. "General perspectives on knowledge management: Fostering a research agenda." *Journal of management information systems 18, no. 1* 5-21.

Vogrinčič, Sergeja, and Zoran Bosnić. 2011. "Ontology-based Multi-Label Classification of Economic Article." *Computer Science and Information Systems 8, no. 1* 101-119.

Von Krogh, Georg. 1998. "Care in knowledge creation." *California management review 40, no. 3* (California management review 40, no. 3) 133-153.

Wang, Chunye, Ram Akella, and Srikant Ramachandran. 2010. "Hierarchical service analytics for improving productivity in an enterprise service center." *In Proceedings of the 19th ACM international conference on Information and knowledge management.* ACM. 1209-1218.

Wang, Chunye, Ram Akella, Srikant Ramachandran, and David Hinnant. 2011. "Knowledge Extraction and Reuse within "Smart" Service Centers." *In SRII Global Conference (SRII) Annual.* IEEE. 163-176.

Wang, Wei, Chuan Xiao, Xuemin Lin, and Chengqi Zhang. 2009. "Efficient approximate entity extraction with edit distance constraints." *In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data.* ACM. 759-770.

Wang, Yong, and Julia Hodges. 2006. "Document clustering with semantic analysis." *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on, vol. 3.* IEEE. 54c-54c.

Waters, John K. 2005. "Managing unstructured information." *Application Development Trends Articles 2, no. 1.*

Willett, Peter. 2006. "The Porter stemming algorithm: then and now." *Program: electronic library and information systems 40.3* 219-223.

Wimalasuriya, Daya C., and Dejing Dou. 2010. "Ontology-based Information Extraction: An Introduction and a Survey of Current Approaches." *Journal of Information Science.*

Wong, Wilson, Wei Liu, and Mohammed Bennamoun. 2012. "Ontology Learning from Text: A Look Back and into the Future." *ACM Computing Surveys (CSUR) 44, no. 4.*

Wu, Chuni. 2008. "Knowledge creation in a supply chain." *An International Journal 13, no. 3* 241-250.

Yin, Robert K. 2003. *Case Study Research: Design and Methods, 3rd edn. Applied Social Research Methods Series, vol. 5.*

Yonghong, Yu, and Bai Wenyang. 2010. "Text clustering based on term weights automatic partition." *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on, vol. 3.* IEEE. 373-377.

Yu, Yao-Tang, and Chien-Chang Hsu. 2011. "A structured ontology construction by using data clustering and pattern tree mining." *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on. Vol. 1.* IEEE.

Zaman, Nazia, and Juan Li. 2014. "Semantics-Enhanced Recommendation System for Social Healthcare." *In Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on.* IEEE. 765-770.