This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

Semantics-enhanced Online Intellectual Capital Mining Service for Enterprise Customer Centers

Juan Li, Nazia Zaman, Ammar Rayes, Ernesto Custodio, *Member, IEEE*

Abstract— One of the greatest challenges of an enterprise's service center is to ensure that their engineers and customers are provided with the right information in a timely fashion. For this purpose, modern organizations operate a wide range of information support systems to assist customers with critical service requests and to provide proactive monitoring, where possible, to prevent service requests from occurring in the first place. It is often the case that relevant information is scattered over the Internet and/or maintained on disparate systems, buried in large amount of noisy data, and in heterogeneous formats, thereby complicating the access to reusable knowledge and extending the response time to reach a resolution. To address these challenges, in this paper we propose an effective knowledge mining solution to improve the quality of service request resolution. We model the service resolution problem as an online search and classification problem, and use domain knowledge in the form of ontology to guide effective machine learning. Our proposed solution has been extensively evaluated with experiments and has been used in a real enterprise customer center.

Index Terms— Knowledge management service, semantic web, ontology, data mining, machine learning, natural language processing, business rules, event processing, production rule system

1 Introduction

ENTERPISE services centers and IT consulting services are a growing business in today's fast-paced marketplace. They provide a primary way for enterprises to interact with their customers. Service centers receive a large number of service requests from customers and partners. IT consulting services are also in high demand as companies are under pressure to maintain a technical advantage in today's hypercompetitive market. Accurate and timely delivery of pertinent information to assist in service request prevention and resolution is critical for providing the highest levels of service to customers. This information can be serviced in the form of a curated knowledge repository and can be used to infuse the service request with knowledge on how to solve the issue. In addition, the knowledge can be coded into business rules that can be used in the form of automated event processing to proactively fix or even prevent issues in other customer networks with the same devices/software images, thereby avoiding service requests all together.

New technology has enabled the generation of more information in the hands of customers, as never before has a customer had so much information about a company's products and services. Based on a recent study, most of their service requests can be answered using enterprise's information sources. For example, over 70% of customers and partners find answers on Cisco's Web [28]. However, most of the explicit knowledge assets of today's organizations consist of unstructured textual information in electronic form. In most cases, these data may be located in different sources such as service request repositories, enterprise websites, and social networks of Subject Matter Experts (SMEs). Moreover, the data may be stored in heterogeneous formats, in most cases unstructured, including text, command line interface (CLI) output, and Web pages. The large amount of heterogeneous data sources complicate request resolution causing lengthy diagnosis time. Furthermore, educated and demanding consumers using new communication channels (such as online chatting) call for higher quality of services [46]. Unless relevant data is displayed promptly and efficiently, service center engineers tend to ignore the provided reference information and spend their time working on the actual service request [39].

1.1 Problem

Let us first see how service center engineers deal with service requests from customers. After receiving a service request, the engineers need to understand the issue or problem before they can deliver an accurate answer. For questions they are familiar with, they may quickly give suggestions based on their experiences. For questions new to them, they may search the questions from different sources (such as the Internet, the social network sites, related service requests, Enterprise's white papers or technical reports), read the relevant documents to understand their meaning and then organize and summarize the answer. The information exchanged between the customer

- *Juan Li and Nazia Zaman are with Department of Computer Science at North Dakota State University, Fargo, ND 58108 USA (e-mail: j.li@ndsu.edu, nazia.zaman@ndsu.edu).*
- *Ammar Rayes and Ernesto Custodio are with Cisco Systems, San Jose, California. E-mail: rayes@cisco.com, ecustodi@cisco.com.*

xxxx-xxxx/0x$xx.00 © 200x IEEE Published by the IEEE Computer Society

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

and the service center is documented using a service request system. This system is used to capture the case history as well as to implement the case handling workflow. Engineers often use informal methods to research inquiries and deliver answers to customers (manuals, binders, sticky notes, case histories, etc.) [30]. However, these methods are not optimal because they do not promote automation, knowledge sharing, and knowledge reuse or expedient resolutions [45].

Consulting services operate in more proactive situations. They are expected to provide customers the expertise to build, improve and scale their IT environment. They are also expected to find and prevent problems before they occur. These demands face scenarios similar to those experienced by service center engineers since consulting engineers tend to perform repetitive tasks within a customer's network or across their customer base. These tasks, such as performing network assessments and optimizations are often done manually.

1.2 Existing Approaches

Nowadays modern service centers have been working on improving resolution efficiency by building knowledgebase solutions. Knowledge management sharply reduces the need for escalation within and beyond a service center [47]. Often (over 70% of the time) a service request being asked to a service center, has usually been asked before, and most likely will be asked again. Therefore, most service centers try to capture answers to previously posed requests and build structured knowledge from this experience [38, 51]. Upon receiving a service request, the system will match the service request with similar cases which have been resolved before. This kind of knowledge contributed by skilled engineers and based upon actual experience, can be presented in the form of a knowledge repository or infused into the actual service request for faster access and to facilitate efficient service request response.

Knowledge management systems built upon service center engineers' previous experience on answering similar service requests or customer service engagements will facilitate efficient responses to customer inquiries and resolutions. However, it has a few shortcomings. First, service requests which have not been posed before cannot benefit from this system. Second, up-to-date information from other sources such as those discussed in social network sites cannot be quickly integrated to the knowledgebase to serve customers. Based on [44], service center staff cannot keep pace with the complexity of requests, and existing tools or skills cannot keep up with customer expectations. Request resolution rates have dropped for consecutive years, leaving customers with just a three-in-four chance of having their issue resolved.

1.3 Our Approach and Contributions

To address the aforementioned problems, we propose an online knowledge mining system, which can help users locate the most up-to-date and relevant information related to service requests or customer engagement, even if the users' requests are new to the system. To get up-to-date information related to particular topics, we turn to the richest sources in the world – the Internet and the enterprise's intranet. We implement a semantics-expanded search engine, which can search information based on the semantics rather than syntax. To remove the massive amount of noise returned from the search engine and shape the information into a powerful representation, we propose and implement a semantics-enhanced multi-level classification mechanism. The proposed classifier can classify information to a structured format that can be easily understood and absorbed by service center engineers or customers. The structured information is called Intellectual Capital, or IC for short. IC can be used in the form of business rules that can be used by a production rule system to facilitate inference and reuse.

The proposed IC mining model offers better categorization of service request resolution data along with improved specification and matching techniques. The proposed work integrates rich semantics, advanced search with data mining and machine learning technologies. The goal of this work is to realize a usable, intelligent, and effective framework for IC mining. In particular, the contributions of this paper are summarized as follows:

1. We propose an online search and classification model to mine IC. This approach overcomes the existing problems of knowledge discovery in service centers, namely (1) cold start, i.e., unable to solve the never-seen-before problems, and (2) difficult to integrate up-to-date new information.
2. We design algorithms to utilize the enterprise's ontology to guide search and data analysis leading to better performance.

1.4 Paper Organization

The rest of the paper is organized as follows. Section 2 gives an overview of the system architecture. Section 3 describes the details of our methodologies. Section 4 presents a use case of the proposed mechanism. In Section 5, we evaluate the proposed methods and show the effectiveness of this model with a comprehensive set of experiments. Related work and concluding remarks are provided in Sections 6 and 7, respectively.

2 System Architecture

2.1 Overview

Fig. 1 shows the architecture of the proposed IC mining system. It consists of two major components, namely federated IC search engine and ontology-enhanced classifier. A predefined enterprise ontology is used in both components to improve the system performance. The working process of the system is as follows: To create IC of a particular topic, the service center engineer needs to input set of keywords to the integrated search engine to search for a particular problem. Keywords can be expanded with semantically-related concepts to disambiguate and refine the query. Relevant documents and Web pages retrieved by the search engine will go through the classifier's preprocessor to make the data machine learning ready. Preprocessed data then will be fed to the classifier to get classified.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

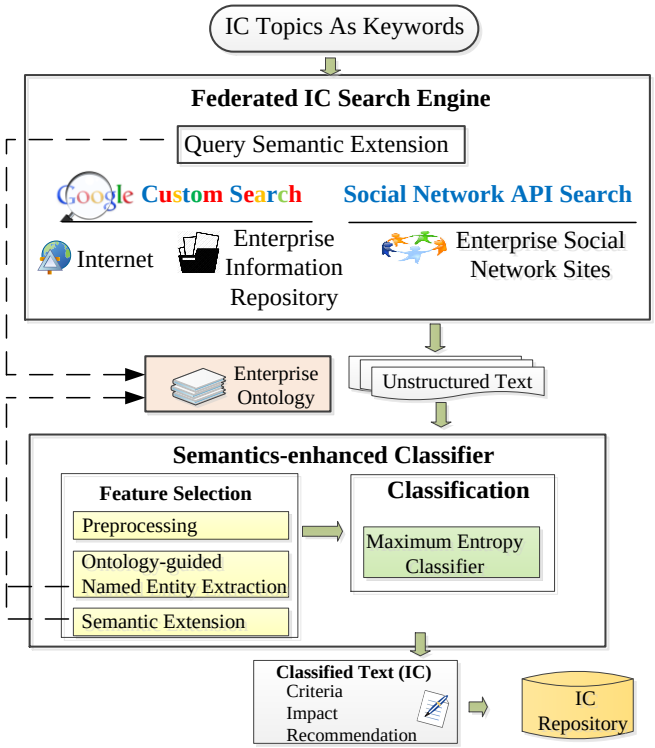


Fig. 1. IC mining system architecture

To improve the performance of the classifier, again, enterprise ontology is applied to guide the classification process. The classified results (IC, text fragments) will be provided to the engineers for verification and summary. Finally, the engineers verify/edit the final IC and store it in the IC repository. The detailed workflow of the system is explained in Section 4.

We leverage the complementary strengths of humans and computer software to solve the knowledge discovery and management task more efficiently. By providing a friendly interface for user feedback in areas such as problem definition, query refinement, results evaluation and verification, we can continuously capture engineers' knowledge and incorporate that into the searching and learning system to optimize the result.

2.2 Enterprise Ontology

We utilize semantic web technologies, in particular ontology, to add domain knowledge to unstructured information to enable more accurate and intelligent knowledge management. Defined as "specifications of a shared conceptualization of a particular domain", ontology [31] provides a shared and common understanding of a domain that can be communicated among people and across application systems, and thus facilitate knowledge sharing and reuse. Besides providing formal, machine-executable meaning on concepts, ontology supports inference mechanisms that can be used to enhance semantic matchmaking. Moreover, ontology provides an adequate grounding for the representation of coarse- to fine-grained entities and is able to deal with the subtleties of different text. With the assistance of ontology, our mining tool can automatically infer relationships between important concepts thus enabling accurate knowledge extraction and organization.

Enterprises in different domains should use their own domain ontologies that represent important concepts and relationships in their business, such as products, technologies, and customers. The proposed IC mining mechanism does not depend on any particular ontology. As an example, in our project we have exploited an ontology provided by Cisco, which contains Cisco's core background knowledge including Cisco products, Cisco networking solutions, and Cisco technologies represented in RDF/OWL [32]. The ontology was designed by Cisco experts. It includes 9901 entities in which 1476 classes and 8425 instances. Fig. 2 shows part of the high-level ontology.

The main semantic relationships we have utilized include: (1) the hierarchical relationship between classes. For example, from the root to the leaf branch of the ontology tree we have '*Cisco Products*'->'*Switches*'->'*LAN Switches*'>'*Cisco Catalyst 5000 Series Switches*'. These specialization/generalization relationships correspond to what we know as *IS-A* relationship resulting in a hierarchical arrangement of concepts. We should note that one class may have multiple super classes. (2) The *type* relationship between a particular class and its corresponding instances. For example, '*Cisco Analog Station Gateway*' has type '*Voice Gateways*' and '*RIP v2*' has type *'IP Routing*'. (3) Important object properties which specify the high level properties between concepts. For example, in our ontology, '*Product has network solutions*' and '*Product has technology*' are two such kind of properties.

2.3 Semantics-Assisted Federated Search Interface

Instead of letting users search multiple sources to get relevant information, we have implemented a single federated search interface for different information sources, such as the Internet, the enterprise's Intranet, and the enterprise's social network sites to get IC-relevant webpages and documents. We use Google's custom search API and the enterprise's social network API to facilitate the search. As we know the users of the search engine would be engineers of the service center, we can roughly predict the kinds of things they might search for based on the industry, objective and relevant technology. For example, a service center that supports a Network Operations Center (NOC) would likely be interested on how to configure different routing protocols on various router models. Therefore, we can anticipate and expand their queries to help users refine, specify, and disambiguate their queries, thus finding results that are the most relevant.

The query refinement and expansion are performed based on the enterprise ontology. For this purpose, we have proposed a Semantic Entity Recognition (SER) and Semantic Entity Expansion (SEE) algorithm to locate and expand important entities appeared in the query. The detailed algorithm description is presented in Section 3.1. Fig. 3 illustrates an example about how a query is expanded with all of its semantic meanings and related concepts, and how users can use the expansion to refine and disambiguate their queries. The details of this example is illustrated in Section 4.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

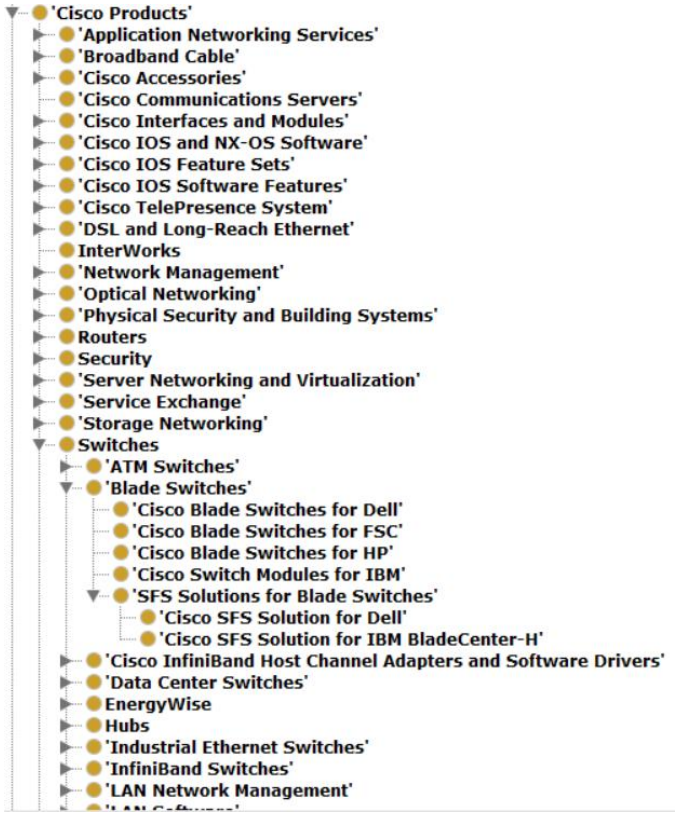


Fig. 2. Part of the domain ontology used in this work

2.4 Semantics-enhanced Multi-level Classifier

The search results returned by the federated search engine include web pages and documents. Although the results are relevant to the query, they contain a massive amount of irrelevant and noisy information. Users, normally, would have to read these web pages and documents to understand the content, and then manually find the answers of the query by themselves. This process is very time and energy consuming. Therefore, it is not enough to find and direct users to the best place containing the desired information. The specific part of the information contained in the source must also be extracted. Moreover, the extracted information needs to be shaped into a powerful and compacted representation that can be easily understood and absorbed by the service center engineers. For this purpose, we have to understand the information needs of the service center engineers.

In our project, we surveyed service engineers and network consulting engineers. To understand the problem and provide solutions to service requests, engineers normally need to know the following information related to the problem: the *problem/criteria*, the *impact*, and the *recommendation*. *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. The *recommendation* is the suggestions to solve the problem. These three types of information form the IC content in our project. For different enterprises, the IC format can be slightly different. For example, some service requests need *troubleshooting* information.

Based on the above observation, we model our IC mining problem to a multi-level classification problem. In particular, we have a set of training documents $D=\{D_1, \ldots, D_n\}$, such that each document is a webpage identified by an URL. For a particular document D, it is composed of a set of paragraphs $D=\{P_1 \ldots P_m\}$. For each paragraph Pi in a document D, a class value is drawn from the set of different discrete values. For a given test document instance for which the paragraphs' class labels are unknown, the training model is used to predict the class labels for this instance. At the first level of classification, we identify noise such as white paper directories, table of contents, and social network signatures. In this level, a class value is drawn from a set with two values "*noise*" and "*non-noise*". After noise is identified and removed, at the second level of classification, we distinguish IC-relevant paragraphs from IC-irrelevant paragraphs by classifying paragraphs as either "*IC-relevant*" or "*IC-irrelevant*". At the final level, IC-relevant paragraphs are further classified as "*criteria*", "*impact*", and "*recommendation*". We present the detailed classification process in Section 3.2.

3 METHODOLOGY

In this section, we present the major methods/algorithms we proposed to support the functionality of the IC mining system.

3.1 Semantic Entity Recognition and Extension

3.1.1 Semantic Entity Recognition

To figure out the semantic meaning of a query or a paragraph of text, we need to identify the important entities appeared in the query or text, and then expand them with their semantics meaning. As query is a special kind of text, the algorithm used in extracting semantic entities from a paragraph of text can be applied to extracting entities from queries with minor changes. Therefore, in this section we present semantic entity recognition from paragraphs of text.

Pre-processing is performed on the text before extracting entities. First the text documents are cleaned from any unnecessary information such as the surplus "clutter" (boilerplate, templates) around the main textual content of a web page boilerplate content extractor [52]. And then data is segmented into sentences. We have used the Punkt sentence segmenter proposed by Kiss et al [1] to segment sentences. Afterwards, a tokenizer is used to divide text into a sequence of tokens, which roughly correspond to "words". We adopted Penn Treebank Tokenization [2] to perform this. Some of the most common short function words, such as "the", "a", "is", "which", are useless in text analysis. We remove these words from the text to reduce the data size. Stemming, the process of reducing a word to its root or simpler form by removing inflectional endings, is also performed. To avoid over stemming errors (for example, for some named entities, we do not want to stem them and leave them into some meaningless state) we adopt the Improved Porters Algorithm [34] to refine the existing stemming algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

The enterprise's domain ontology is used to direct the entity extraction process. For service centers, domain knowledge is often codified in the form of specialized lexicons and ontologies. Text contents from the Internet, on the other hand, are in the form of unstructured notes. Using domain knowledge to locate semantic entities recorded in such form presents a challenging problem. This problem cannot be solved by traditional Named Entity Recognition (NER) approaches. Traditional NER works to identify all textual references of named entities – noun phrases referring to specific individuals like persons, organizations, location and so on. NER systems can utilize different techniques to identify and extract entities, such as grammar based entity detection [3-4], statistical named entity recognizer [5] and gazetteer based entity recognizer [6]. Most of the existing NER approaches do not consider the usage of the ontology as a reference. Since entities in the ontology are represented as strings, the ontology-guided named entity extraction problem can be modeled as a string matching problem, as proposed in [35]. More challenging is that the form of a named entity in free text can vary substantially from its ontology version. For example, the semantic entity 'Cisco Catalyst 5000 Series Switches' in the ontology might be referred to as 'Catalyst 5000' or 'Catalyst 5000 Switches' or even '5000 Series' in the text. Therefore, this problem is to extract semantic entities from text which is informal or otherwise prone to noise and errors.

There have been multiple research works proposed on proximate string matching [35, 48, 49, 50]. The most popular and effective approaches are based on comparing the edit distance between strings. Edit distance between two strings is the number of deletions, insertions, or substitutions required to transform source string into target string. Edit distance can effectively capture typographic errors, words with alternative spellings, and does not rely on the separation of word boundaries [35]. Therefore, edit distance with its variants (such as Levenshtein distance, Damerau-Levenshtein distance, Jaro-Winkler distance) have been wildly used in string matching and comparing. A serious problem of existing approaches using edit distance is the significant computational cost, especially for matching long strings.

To address this problem, we propose a simplified entity extraction algorithm based on the special usage of enterprise service center terminologies. The basic strategy is to maximally reduce search space and search complexity. We first choose possible candidate strings to do matching. Then we decompose the candidate string to single words, and match the words with the given ontology. Afterwards, another round of scan is performed on matched words to combine them back to phrases. A modified edit distance-based algorithm is devised to measure the relevance of the phrase to ontology concept with a single word as a unit for allowable edit operations. This approach avoids applying expensive edit distance matching on long strings, thus dramatically reduces the performance cost and simplify the complexity.

In order to efficiently search ontology content, we created an inverted index over the ontology, which associates each semantic entity (class, and instance) with a set of terms of lexical representations. Specifically, the keywords associated to each ontology entity are extracted from the standard ontology meta property *rdfs:label*. We remove some common keywords before indexing. An example of the generated inverted index is shown in Table 1, where each keyword is associated to one or several ontological entities. The ontological entities are uniquely identified by the identifier of the ontology. These indices are used to identify potential ontological entities that can be associated to a particular keyword.

TABLE 1
KEYWORD-ONTOLOGY INDEX

Keyword	Ontological entities
Access	E2, E8
BBSM	E20
…	
Catalyst	E34, E35, E36…

In this algorithm, the longest multi-word expressions that appear in the text are mapped to the most specific concepts in the ontology. Firstly, we apply the part-of-speech (POS) procedure [53] to tag the documents. POS tagging is used as the basis for extracting higher-level structure – phrases. As ontology concepts are typically symbolized in text within noun phrases, we only consider noun phrases for potential ontological entities. In this way, we dramatically reduce the problem space.

Secondly, for tokens (corresponds to terms/words) appeared in noun phrases, we use the inverted index to search the semantic entities associated to the tokens. If there's a hit, we will tag the word with the entity ID. We should note that for one word, it may belong to multiple ontology concepts. In such case, we tag the word with IDs of all associated semantic concepts. For example, word 'Catalyst' appears in multiple concepts of the ontology such as 'Catalyst 5000 Switch Series' or 'Catalyst 6000 Switch Series'.

Thirdly, after we have finished the keyword-entity matching and tagging phase, we try to identify potential semantic entities in the document. This is done by scanning the tags of the terms in the same noun phrase: if multiple words in the noun phrase point to the same semantic entity, they should be considered as belonging to the same entity. The rationale of this approach is based on the observation that some words tend to be omitted and the orders of the words may be switched in phrases used in the informal documents. For example, noun phrase 'Catalyst 5000' in the document would be mapped to the semantic entity 'Cisco Catalyst 5000 Series Switches' in the ontology using this strategy.

Lastly, we apply an edit distance-based algorithm to further verify if the extracted entities are really defined in the ontology. In this algorithm, we treat each word as a character in the original edit distance-based algorithms (for instance, Jaro-Winkler distance). In other words, a single word is a unit for allowable edit operations. Then we employ this algorithm to evaluate the similarity of the extracted entity with the semantic entity defined in the ontology. If their distance is less than a pre-defined threshold,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

Algorithm1 The semantic entity recognition (SER) algorithm

```
Apply POS on the document
for each noun-phrase in the document do
   for each token in the noun-phrase do
      search token in the inverted index of keyword-ontology
      if a hit is found then
         tag the matched token with the ontology entity's ID i
   end for
   if multiple tokens have the same ID i then
      merge them as a single entity
   calculate the adapted edit-distance between the extracted
   entity and the ontology entity identified by ID i
   if distance<= threshold then
     the extracted entity is a legal semantic entity
end for
```

the extracted entity is verified to be a legal semantic entity. Through these four steps, semantic entities are recognized. Algorithm 1 lists the detailed procedure.

3.1.2 Semantic Entity Extension

With semantic entities being identified, we attempt to further strengthen content representation through the use of conceptual abstraction; that is to augment the semantic representation of a document beyond a set of plain words. As it is possible that an extracted semantic entity may correspond to multiple semantic entries in the ontology, we need to choose the most appropriate concept according to the document context using a disambiguation function. The basic idea of the disambiguation function is to compare the semantic similarity between the context information of the extracted entity with each of the related semantic entity defined in the ontology and choose the most similar one. The semantic similarity is computed based on the semantic distance of the ontology graph. The details of the computation can be found from our previous work [54].

We expand the identified entities with semantically relevant entities based on the most important relationships defined in the domain ontology. There are multiple semantic relationships that can be used. In our current implementation, we used two major relationships of an ontology: the 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 aforementioned two relationships, we create two other inverse index tables, in which each keyword is associated with its corresponding super classes (or *type* class for individual objects). Examples of these indexing structures are shown in Table 2 and Table 3.

For every ontological entity in the text, we will expand that entity with all of its ancestor concepts up to a maximal distance. Note that the distance parameter needs to be chosen carefully as climbing up the taxonomy too far is likely to obfuscating the concept representation [24].

With the semantic expansion implemented, we can enrich text with its semantic context. The enhancement of ontology may capture the semantics of the text and overcome the shortcomings of data analysis in the syntax level. For example, one document presents how to configure a router. While another document is about configuring a switch. Literally the words used in these two documents maybe quite different. However, if we expand entities of "router" and "switch" with their semantic ancestor entity "Network Device", we can see that they are related.

TABLE 2
SUBCLASS-SUPERCLASS INDEX

Ontology Entity	Super Class
E1	E3
E2	E4
…	
E10	E1, E7

TABLE 3
INSTANCE-CLASS INDEX

Instance	Type Class
E12	E1, E6
E18	E3
…	
E29	E31

3.2 Semantics-enhanced Multi-level Classification

In this section, we propose an ontology-enhanced classification approach.

3.2.1 Ontology-guided Feature Selection

Feature selection is very important for text classification due to the high dimensionality of text features and the existence of large amount of irrelevant features. A wide variety of methods have been proposed to determine the most important features of classification. So far, most existing text classification systems have adopted the Bag-of-Words (BoW) model known from information retrieval. In the BoW model single words or word stems are used as features and word frequencies or more elaborated weighting schemes, such as TFIDF, are used as feature values [15]. The BoW model, however is restricted to detecting patterns in the used vocabulary only, while conceptual relationships and domain knowledge remain ignored. For example, multi-word entity, "catalyst 5000" is chunked into pieces with possibly very different meanings like "catalyst" and "5000". Using the BoW model, "catalyst 5000" and "switch" will be treated as totally different things although semantically they are closely related.

To address the aforementioned problem, we propose to identify ontological entities from text and then include the identified ontological entities and ontological relations as features. The semantic entity extraction and extension algorithm presented in Section 3.1 can be used for this purpose. In this way, semantic meaning of the feature will be preserved and classification would be more accurate. In particular, features that are considered for classification's purpose include: the expanded semantic entities, the top n most frequent words, the type of the website where the document comes from, inclusion of query keywords, bag-of-hint words, length of a paragraph, and relative location

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

of a paragraph. For words and semantic entities, we use their presence, as opposed to a count. That is if the feature is present, the value is 1, but if the feature is absent, then the value is 0. Documents from different website should be distinguished. For example, social network discussions should be processed differently from whitepaper documents. A document, especially a whitepaper document may include multiple topics/sub-topics. IC-relevant problem may account for only a minor part of the document. Therefore, whether a paragraph contains query keywords should also be considered as a feature related to identify if a paragraph is IC-related. Also, words such as "problem", "note", "recommend", "suggest" may hint that the corresponding sentence is related to an IC category.

3.2.2 Maximum Entropy Classifier

We choose the maximum entropy as our implementation of the classifier, which has proven to produce effective text classification results [33]. The Max Entropy classifier is a probabilistic classifier which belongs to the class of exponential models [43]. Unlike the Naive Bayes classifier, the Max Entropy does not assume that the features are conditionally independent of each other. This is particularly true in our case where our features are obviously not independent. The main idea behind maximum entropy principle is that unknown model generating the sample data should be the model that is most uniform and satisfy all constrains from training data.

Maximum entropy distribution is of the exponential form. The model is represented by the following:

$$P(c|d)=\frac{1}{Z(d)}\exp\left(\sum_i \lambda_i \, f_i(d,c)\right)$$

Z(d) is the normalization function which is computed as:

$$Z(d)=\sum_c \exp\left(\sum_i \lambda_i \, f_i(d,c)\right)$$

In this formula, c is the class, d is the document, and λ is a weight vector. The weight vectors decide the significance of a feature in classification. A higher weight means that the feature is a stronger indicator for the class. The weight vector is found by numerical optimization of the lambdas so as to maximize the conditional probability. It can be estimated by an iterative way using algorithms such as Generalized Iterative Scaling (GIS) [25], Improved Iterative Scaling (IIS) [26], or LBFGS Algorithm [27]. We use the Stanford Classifier to perform MaxEnt classification. To train the weight, we have used conjugate gradient ascent and added smoothing (L2 regularization) [36].

3.2.3 Multi-level Classification for Biased Data

In a document containing IC, there is a huge disproportion in the number of IC-relevant paragraphs and IC-irrelevant paragraphs: on average the irrelevant information accounts for ten times as much as IC information. As most classifiers generally perform poorly on imbalanced data-sets because they are designed to minimize the global error rate [40], the classifier tends to classify almost all instances as IC-irrelevant which is the majority class in our case. Here resides the main problem for imbalanced data-sets, because the minority classes, the IC, are the most important ones that are spread in a large group of majority classes. To address the issue of biased data, we propose a multi-level hybrid-sampling classification mechanism.

Fig. 3. Screenshot of IC search engine with semantic query extension

At the first level of classification, we identify and remove noise information. Then at the second level of classification, we distinguish IC-relevant information from IC-irrelevant information. At this step, we apply two effective techniques, over-sampling [41] and under-sampling [42], to solve the class imbalance problem. In particular, we oversample the minority IC-relevant class samples and under sample the majority IC-irrelevant samples until the IC-relevant samples are not less than the IC-irrelevant samples. As illustrated in our experiments, this multi-level hybrid-sampling approach improves the classification performance by increasing the precision and recall. After IC-relevant data have been identified, we can further classify them to different IC categories.

4 Use Case

Based on the proposed methodology, we have implemented a pilot system – IC mining toolkit. The toolkit includes an IC search engine, an auto (and manual) IC Extractor (the classifier), and an Ontology manager. Figures 3, 4, and 5 are screenshots of these system components. The toolkit is being used and tested in the Cisco service center.

To use the system, engineers need to identify IC problems through feeding a set of keywords to the IC search engine. For example, in Fig. 3 the keywords used are "cisco 7200". The query can be semantically expanded with all related concepts and relationships. This would help the users to narrow down the IC problem and refine the query. For example, after clicking the "Semantically Extend Query" button, the pop-up window (at the right side of Fig. 3) lists all of the concepts and relationships related to "cisco 7200".

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

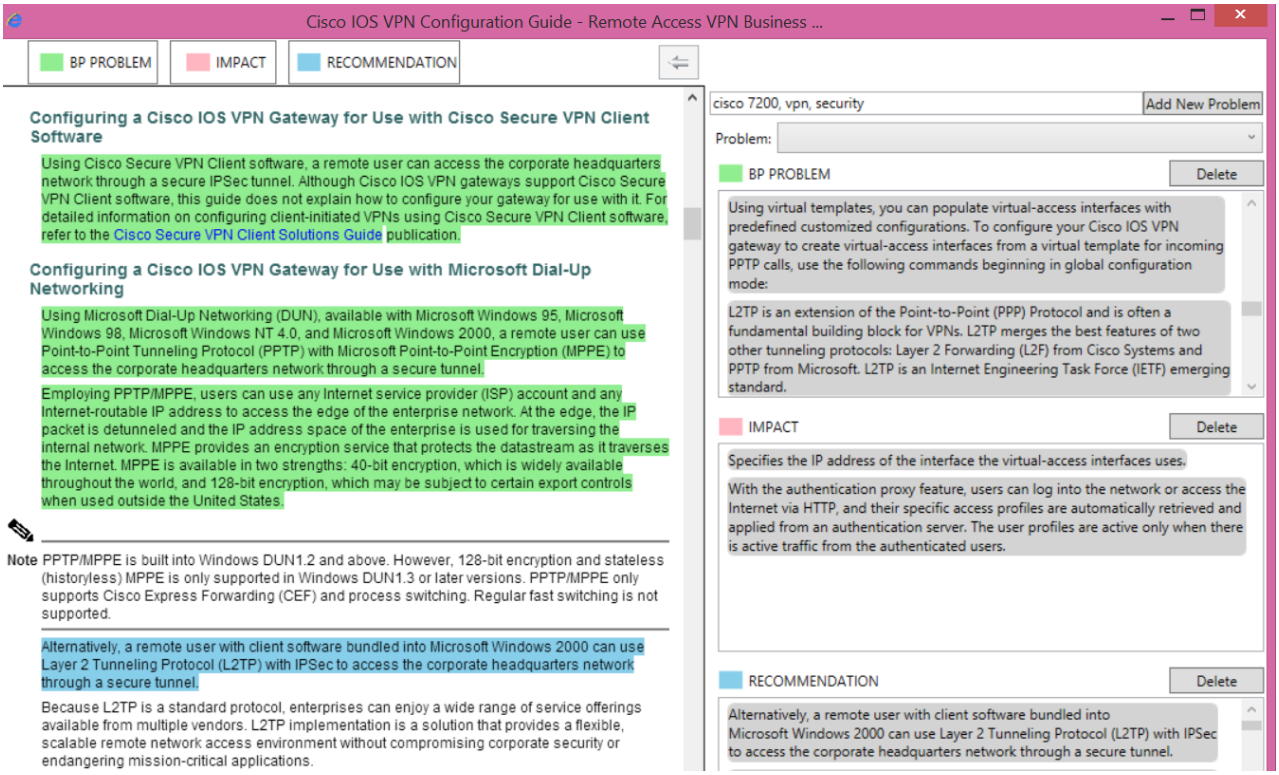


Fig. 4. Screenshot of IC extraction interface

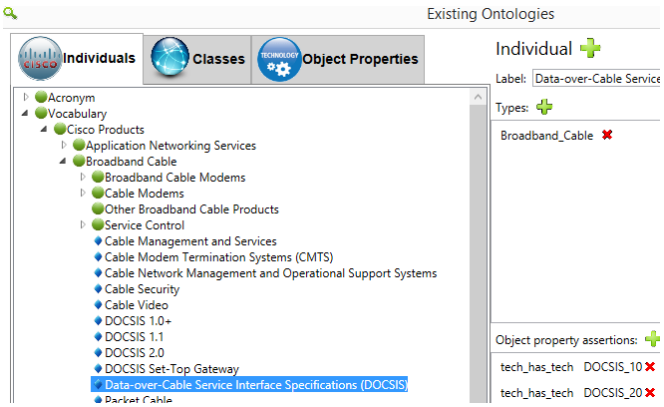


Fig. 5. Screenshot of ontology management interface

The user can use the check box to select the most appropriate concepts/relationships to expand the query. After the user clicks the "Search" button, the federated search engine searches the query from multiple data sources (in our case, including the Internet, the Cisco Intranet and Social Network sites). The user can manually choose search results for next step knowledge mining, otherwise the system will automatically pick the top k results to extract IC.

Fig. 4 is the screenshot of IC extractor (classifier). The IC extractor shows the user how IC is extracted from the original text. As shown in Fig. 4, different categories of IC are highlighted with different colors in the left panel. The extracted IC is displayed in the right panel. We provide users a friendly interface to correct/revise the extracted IC. The user can add IC by selecting and highlighting the appropriate text from the left panel. The selection will be automatically displayed in the right panel. Also, the user can remove an extracted IC.

We also provide users an interface to manage (add, remove, and edit) their enterprise ontology. As shown in Fig. 5, users can manage class, properties, and individuals through this interface. The ontology is saved in OWL/RDF format.

5 Evaluation

In the first set of the experiments, we evaluated the performance of the Semantic Entity Recognition (SER) algorithm. The semantic entities extracted were compared against manually pre-detected entities in each document. The algorithm were evaluated using recall and precision measures. The results are illustrated in Table 4. The definition of recall and precisions are defined as follows:

$$recall = \frac{|\, relevantEntities \cap retrievedEntities \,|}{|\, relevantEntities \,|}$$

$$precision = \frac{|\, relevantEntities \cap retrievedEntities \,|}{|\, retrievedEntities \,|}$$

TABLE 4
Performance of the SER Algorithm

	1-Word	2-Word	3-Word	4-Word
Precision	98.1%	100%	100%	100%
Recall	96.3%	94.5%	94.4%	88.9%

As describe in Section 3.1.1, it is possible that an extracted entity may correspond to multiple semantic entities defined in the ontology. In the entity extraction phase, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

do not determine which semantic entity among the multiple candidates in the ontology is most relevant to the extracted entity (we will determine that in the entity extension phase). Therefore, as long as a semantic entity defined in the ontology is extracted from the source text, we count it as a "relevantEntity". From Table 4, we can see that the SER parser achieves almost perfect precision and good recall for various lengths of semantic entities. Because all of the noun phrases retrieved are matched and further verified with the enterprise ontology, the parser gets perfect precision. The recall rate is mainly impacted by the POS tagger used.

The proposed SER algorithm is a simplified algorithm; while the simplification does not noticeably impair the precision of the system thanks to the special usage of semantic entities in the knowledge sources. The form of a named entity in our knowledge sources can vary from its ontology version. However, the major variations include world-level insertion, deletion, substitution, and permutation. All of these variations can be effectively captured by the proposed algorithm.

In the next set of experiments, we evaluated the performance of the classifier. The data set includes 83 configuration Best Practice use cases that have been manually tagged by human experts. The original documents are returned by the IC search engine. After pre-processing, human experts read the documents and tag each paragraph of the documents as one of *unrelated*, *criteria*, *impact*, and *recommendation*. The category of *criteria* is termed "BP problem" in the toolkit.

Feature candidates include 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. The classifier model then uses these features to determine the class label for a particular paragraph. Each data set has two tab-separated columns where first column indicates the class label of a paragraph and the last column is a comma-separated feature set for that paragraph. Our model also maintains a property file containing several parameters to tune the model: adjust different variables of the classifier (such as regularization, convergence tolerance for parameter optimization, smoothing method) for performance optimization.

After choosing an initial feature set, a productive method for refining the features is called error analysis. At first, a development set containing the corpus data is selected to create the model. This development set is then further divided into the initial training set and the development test set; the former set is used to build the model and the later one is used for error analysis. We split the development set with a random partition of 80% data in initial training set and the rest 20% is the development test set.

To evaluate the effectiveness of the proposed system, we investigated how this toolkit improves the productivity of the Best Practice IC extraction for service request resolution. We compare the average time used for extracting IC related to a particular topic manually and with the assistance of our IC mining toolkit. In the manual IC mining process, engineers input the IC topic in terms of a set of keywords to search engines and Cisco internal search tools. Then the engineers read and understand the documents or webpages extracted by the search engines. Next they record (copy/paste) related information regarding "criteria", "impact", and "recommendation" of the problem to a document. And finally they can summarize the IC based on the recorded information. With the assistance of the IC mining tool, the engineers simply input the keywords to the toolkit and relevant information categorized as "criteria", "impact", and "recommendation" will be automatically returned to them. For incorrect or incomplete knowledge, the engineers can use our tool to highlight the documents/web pages to correct the returned IC.

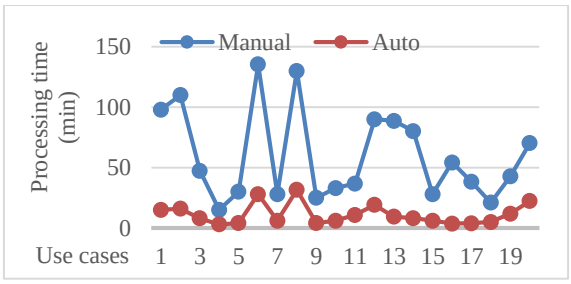


Fig. 6 Comparison of processing time of auto and manual IC extraction

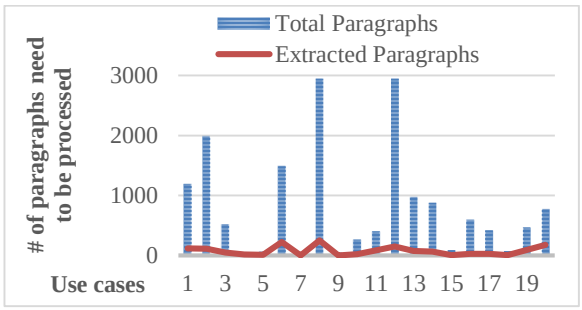


Fig. 7 Comparison work load of auto and manual IC extraction

Fig. 6 plots the processing time in these two scenarios to solve the same set of problems. We can see that our tool dramatically reduce the response time. We must note that we have already included the time used to correct the inaccurate IC returned by the toolkit. Fig. 7 compares the information load of manual and automatic IC mining. The information load is computed as the number of paragraphs users have to read or considered to solve a problem. Again, if the paragraphs returned by the tool is not right, we would add all paragraphs the engineers manually processed.

We conducted experiments to evaluate the performance of the IC mining classifier using different feature sets. The feature set includes: frequent words (W), extended semantic entities (S), inclusion of query keywords (K), type of website (T), length of paragraph (L), and relative location of the paragraph (R). Our goal is to determine whether incorporating semantically extended entities and other features help improve the performance. For measuring the performance, we use macro-averaged F1, accuracy, precision, recall, which are defined as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2015.2474861, IEEE Transactions on Services Computing

TABLE 5
FPERFORMANCE MEASUREMENTS FOR DIFFERENT FEATURE SETS - 10 FOLDS CROSS VALIDATION

Feature Set	Micro-avg. F1	Macro-avg. F1	Accuracy	Precision	Recall
W	0.74	0.61	0.76	0.66	0.58
W+S	0.8	0.71	0.8	0.73	0.68
W+S+K	0.81	0.72	0.81	0.73	0.7
W+S+K+T+L+R	0.82	0.73	0.82	0.74	0.72

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

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

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

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

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

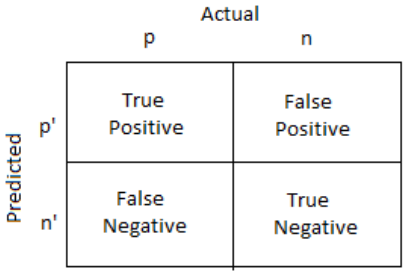


Fig. 8. Confusion Matrix.

All these measurements are being calculated using the confusion matrix in Fig. 8 based on two possible outcomes: positive (p) and negative (n). To conduct the experiment, we take an average of 10 runs by shuffling the dataset. We have used 10 folds cross validation by subdividing the original dataset. For all these folds we got similar results which indicates the stability of the score. 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 improves performance and the combined feature set performs best as compared to others.

Fig. 9 shows how under-sampling improves the classification performance on imbalanced data. The figure illustrates the performance under different ratio of IC-relevant samples and IC-irrelevant samples. As can be seen from the figure, the first set of data (labeled as "none", which means no under-sampling has been applied) has very high accuracy but low precision and recall. This is because that the classifier try to minimize the global error and classify more instances as IC-irrelevant which is the majority class in our case. We then under-sampled the majority IC-irrelevant paragraphs. Although the accuracy decreases, the

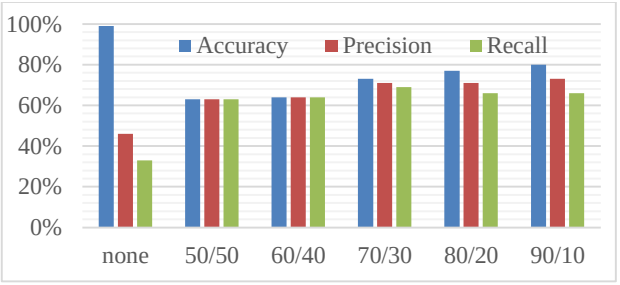


Fig. 9. Classification Performance with Different Under-sampling rate

precision and recall increases as we under-sample the irrelevant data. The precision and recall rate are more important than the accuracy in our project, as false positive (classifying irrelevant data to relevant) is acceptable for us, but false negative (classifying relevant data as irrelevant) is not.

6 RELATED WORK

Enterprises have realized the importance of using their knowledge assets to provide better customer service. As Rasooli et al. pointed out, there are different sets of factors involved in the process of knowledge management of call centers and service centers, including knowledge acquisition, knowledge generation, knowledge distribution, knowledge adaptation and knowledge utilization [7]. Based on these factors, using a case study approach, an abstract high-level knowledge management model for call centers was proposed [7]. In [8], the authors described a SVM-based call-type classification and acoustic modeling for speech recognition in the context of a telephone-based call center corpus. In [9], researchers present a knowledge discovery framework using a service oriented architecture (SOA) designed with the primary goal to ease usage for non-knowledge discovery experts. They gave high-level guidelines on the development and application of the proposed framework. In their work of knowledge extraction and reuse within service centers [10, 11], the researchers proposed a text analytics system which includes hierarchical classifier and a recommender. The classifier would classify service center requests to well-defined categories, namely what, why, and how. The recommender will recommend previously solved solutions to similar requests. Although these knowledge management systems have been proposed and designed, none of them can effectively address the challenges we mentioned previously, such as dealing with very dynamic social media and never-seen service requests. In addition, most existing researches do

not consider domain knowledge of the enterprise.

Researchers have proposed and developed ontology-based knowledge management systems to enable efficient information sharing and reusing, as ontology can effectively handle the information heterogeneity issues in different data sources. For example, in [14] Fernández et al. proposed an innovative, comprehensive semantic search model by utilizing ontologies. This model extends the classic information retrieval model, addresses the challenges of the massive and heterogeneous Web environment, and integrates the benefits of both keyword and semantic-based search. Similarly, in [15] G. Rong et al., developed an ontology-based information retrieval system to manage and retrieve non-metallic pipe knowledge of oilfield. A. Uszok et al. [13] proposed to use a global ontology to manage knowledge in coalition environment domain. In [17], an ontology-based knowledge management approach for E-Learning systems was presented. The approach also integrated data quality component. The authors of [18] describes an ontology-based method for forest knowledge management. As surveyed in [16], ontology has been widely used for knowledge discovery and sharing in bio-informatics and medical informatics.

Several approaches have been proposed to classify or annotate documents based on pre-defined categories, domain knowledge or ontology. [19] presented an approach that uses Yahoo!-Categories as a concept hierarchy in order to classify documents using an n-gram classifier. In [20] the authors have presented a search system that uses linguistic ontologies (in particular Multiwordnet [37]) to classify search results online in order to disambiguate result sets with respect to given search terms. Their classification approach is simply computing the cosine similarity between the search results and the multi-senses returned from the linguist ontology of the keyword. A similar approach was also used by Cheng et al. [21]. In their work, they utilized WordNet to classify search results. In [22], the authors expand words in a document with the words' synonyms defined in WordNet, and they claimed this expansion would improve classification accuracy. The authors did not address the issue of how to expand words with multiple semantic meanings. Suganya et al. proposed a two-level representation model to represent text data, one is for representing syntactic information using *tf-idf* value and the other is for representing semantic information using Wikipedia [23]. Three SVM-NN classifiers are applied on the syntactic level, the semantic level, and the combined result of the two previous classifier respectively. All of these existing classification approaches work on the single lexicon level, without considering multi-word semantic entity, noise, and even mistake information. Our approach effectively solved these problems.

7 Conclusions

To help enterprise customer centers resolve service requests and expedite the time to resolve cases using online data, we propose an efficient knowledge management module to transform the mountain of data into reusable knowledge or Intellectual Capital (IC). In this module, the service resolution problem was modeled as an online search plus classification problem. In particular, data will be collected from the enterprise data repositories and the Internet using a custom search engine. Then the search results will be pre-processed and classified to extract IC. A novel classifier was presented, which utilized the enterprise domain ontology to direct the classification process. Experimental results show that the ontology guided-classifier dramatically improve the system performance. This model offers better categorization of service request resolution data along with improved specification and matching techniques. A pilot system based on the proposed strategy has been used in real enterprise service centers. It effectively improves the service engineer's performance and increases the amount of reusable knowledge.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The authors also wish to thank Jinkle Jose, Ramesh Kumar Kocherla, Shameer Ali Madappally, Shiva Kumar Mariyanna from Cisco. This work was supported by a grant from Cisco' IP Based Smart Services program.

References

[1] Kiss, Tibor, and Jan Strunk. "Unsupervised multilingual sentence boundary detection." Computational Linguistics 32, no. 4 (2006): 485-525.

[2] https://catalog.ldc.upenn.edu/LDC99T42

[3] Chiticariu, Laura, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. "Domain adaptation of rule-based annotators for named-entity recognition tasks." In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 1002-1012. Association for Computational Linguistics, 2010.

[4] Mikheev, Andrei, Marc Moens, and Claire Grover. "Named entity recognition without gazetteers." In Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, pp. 1-8. Association for Computational Linguistics, 1999.

[5] Chieu, Hai Leong, and Hwee Tou Ng. "Named entity recognition: a maximum entropy approach using global information." In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pp. 1-7. Association for Computational Linguistics, 2002.

[6] Tjong Kim Sang, Erik F., and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition." In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, pp. 142-147. Association for Computational Linguistics, 2003.

[7] Rasooli, Pooya, and Amir Albadvi. "Knowledge Management in Call Centres." Electronic Journal of Knowledge Management 5, no. 3 (2007): 323-332.

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

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

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

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

[12] Abburu, Sunitha, and G. Suresh Babu. "A Framework for Ontology Based Knowledge Management." International Journal of Soft Computing and Engineering (IJSCE) Volume-3, Issue-3, 2013.

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

[14] Fernández, Miriam, Iván Cantador, Vanesa López, David Vallet, Pablo Castells, and Enrico Motta. "Semantically enhanced Information Retrieval: an ontology-based approach." Web Semantics: Science, Services and Agents on the World Wide Web 9, no. 4 (2011): 434-452.

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

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

[17] Sangodiah, Anbuselvan, and Lim Ean Heng. "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, pp. 105-108. IEEE, 2012.

[18] Fan, Jing, Xiuying Liu, Ying Shen and Tianyang Dong, —Ontology-based Knowledge Management for Forest Channell, In Proc. 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012), IEEE, pp. 1523-1527, 2012.

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

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

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

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

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

[24] Bloehdorn, Stephan, and Andreas Hotho. "Boosting for text classification with semantic features." Lecture Notes in Computer Science 3932 (2006): 149.

[25] Darroch, John N., and Douglas Ratcliff. "Generalized iterative scaling for log-linear models." The annals of mathematical statistics (1972): 1470-1480.

[26] Berger, Adam. "The improved iterative scaling algorithm: A gentle introduction." Unpublished manuscript (1997).

[27] Malouf, Robert. "A comparison of algorithms for maximum entropy parameter estimation." In proceedings of the 6th conference on Natural language learning-Volume 20, pp. 1-7. Association for Computational Linguistics, 2002.

[28] IP Based Smart Services: http://www.cisco.com/web/about/ac50/ac207/crc_new/university/RFP/rfp12074.html

[29] Feldman, Susan. "The high cost of not finding information. KMWorld Magazine." (2004).

[30] Rasooli, Pooya, "Knowledge management in call centers." Master Thesis 2005.

[31] Fensel, Dieter. Ontologies. Springer Berlin Heidelberg, 2001.

[32] Introduction to OWL: http://www.w3schools.com/web-services/ws_rdf_owl.asp

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

[34] Willett, Peter. "The Porter stemming algorithm: then and now." Program: electronic library and information systems 40.3 (2006): 219-223.

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

[36] Ng, Andrew Y. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance." Proceedings of the twenty-first international conference on Machine learning. ACM, 2004.

[37] Multiwordnet: http://multiwordnet.fbk.eu/english/home.php

[38] Rasooli, Pooya, and Amir Albadvi. "Knowledge Management in Call Centres."Electronic Journal of Knowledge Management 5.3 (2007): 323-332.

[39] Service Creation and Enhancement: http://www.cisco.com/web/about/ac50/ac207/crc_new/university/RFP/rfp07015.html

[40] Fernández, Alberto, Salvador García, María José del Jesus, and Francisco Herrera. "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 (2008): 2378-2398.

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

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

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

[44] 2013-2014 Global Contact Centre Benchmarking Report. http://www.dimensiondata.com/Global/Global-Micro-sites/CCBenchmarking

[45] Kim, Youngsoo, Ramayya Krishnan, and Linda Argote. "The learning curve of IT knowledge workers in a computing call center." Information Systems Research 23, no. 3-part-2 (2012): 887-902.

[46] Malisuwan, Settapong, Navneet Madan, Wassana Kaewphanuekrungsi, and Napaporn Petchinda. "Adoption of New Information Economics for Informational System Development in Modern Day Call Center." International Journal of Trade, Economics & Finance 5, no. 1 (2014).

[47] [47] Ashu, Roy, Business Value of Contact Center Knowledge Management: A Strategic Perspective, eGain Communications, 2012.

[48] Hall, Patrick AV, and Geoff R. Dowling. "Approximate string matching." ACM computing surveys (CSUR) 12, no. 4 (1980): 381-402.

[49] Shaikh, Muniba, Nasrullah Memon, and Uffe Kock Wiil. "Extended approximate string matching algorithms to detect name aliases." In Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on, pp. 216-219. IEEE, 2011.

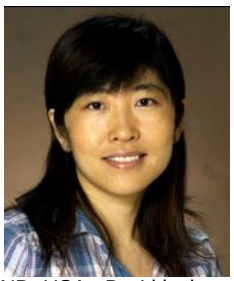
[50] Boytsov, Leonid. "Indexing methods for approximate dictionary searching: Comparative analysis." Journal of Experimental Algorithmics (JEA) 16 (2011): 1-1.

[51] Heitz, Christoph, Geoffrey Ryder, and Kevin Ross. "Knowledge Management in Call Centers: How Routing Rules Influence Expertise and Service Quality." In MSOM Conference Proceedings, pp. 1-7.

[52] boilerplate content extractor: http://code.google.com/p/boilerpipe/

[53] Stanford Log-linear Part-Of-Speech Taggerhttp://nlp.stanford.edu/software/tagger.shtml

[54] Li, Qingrui, Juan Li, Hui Wang, and Ashok Ginjala. "Semantics-enhanced privacy recommendation for social networking sites." In Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on, pp. 226-233. IEEE, 2011.



Juan Li received the B.S. degree in Computer Science from the Beijing Jiaotong University, Beijing, China, in 1997, the M.S. degree in Computer Science from the Chinese Academy of Sciences, Beijing, China, in 2001, and the Ph.D. degree in Computer Science from the University of British Columbia, Vancouver, Canada, in 2008. Currently she is an Associate Professor in the Computer Science Department at the North Dakota State University (NDSU), Fargo, ND, USA. Dr. Li is the author of one book and more than 70 articles. Dr. Li's major research interest lies in distributed systems, intelligent systems, social networking, and semantic web technologies.

Nazia Zanman received a B.S. degree from the University of Dhaka, Dhaka, Bangladesh, in 2007, and a M.S. degree from the same university, in 2009. Currently, she is a PhD student of Computer Science Department at the North Dakota State University, Fargo, ND, USA. Her current research focuses on intelligent systems, social networking, and natural language processing.

Ammar Rayes is a Distinguished Engineer at Cisco Systems and the Founding President of The International Society of Service Innovation Professionals, www.issip.org. He is currently chairing Cisco Services Research Program. His re-search areas include: Smart Services, Internet of Everything (IoE), Machine-to-Machine, Smart Analytics and IP strategy. He has authored / co-authored over a hundred papers and patents on advances in communications-related technologies, including a book on Network Modeling and Simulation and another one on ATM switching and network de-sign. He is an Editor-in-Chief for "Advances of Internet of Things" Journal and served as an Associate Editor of ACM "Transactions on Internet Technology" and on the Journal of Wireless Communications and Mobile Computing. He received his BS and MS Degrees in EE from the University of Illinois at Urbana and his Doctor of Science degree in EE from Washington University in St. Louis, Missouri, where he received the Outstanding Graduate Student Award in Telecommunications.

Ernesto Custodio received a bachelor's degree from the State University of New York at Buffalo, in 1995, and a MBA from the H. Wayne Huizenga School of Business and Entrepreneurship in 2013. Currently he is a Ph.D. student of Information Systems at Nova Southeastern University's Graduate School of Computer and Information Sciences. Ernesto is a Senior Software Development Manager at Cisco working on the next generation of knowledge management systems.