Toward Unified Cloud Service Discovery for Enhanced Service Identification

Abdullah Alfazi[1], Quan Z. Sheng[2], Ali Babar[1]
Wenjie Ruan[1] and Yongrui Qin[3]

[1] School of Computer Science
The University of Adelaide, Adelaide SA 5005, Australia
{abdullah.alfazi, Ali.babar, wenjie.ruan}@adelaide.edu.au
[2] Department of Computing
Macquarie University, Sydney, Australia
michael.sheng@mq.edu.au
[3] School of Computing and Engineering
University of Huddersfield, United Kingdom
y.qin2@hud.ac.uk

Abstract. Nowadays cloud services are being increasingly used by professionals. A wide variety of cloud services are being introduced every day, and each of which is designed to serve a set of specific purposes. Currently, there is no cloud service specific search engine or a comprehensive directory that is available online. Therefore, cloud service customers mainly select cloud services based on the word of mouth, which is of low accuracy and lacks expressiveness. In this paper, we propose a comprehensive cloud service search engine to enable users to perform personalized search based on certain criteria including their own intention of use, cost and the features provided. Specifically, our cloud service search engine focuses on: 1) extracting and identifying cloud services automatically from the Web; 2) building a unified model to represent the cloud service features; and 3) prototyping a search engine for online cloud services. To this end, we propose a novel Service Detection and Tracking (SDT) model for modeling Cloud services. Then based on the SDT model, a cloud service search engine (CSSE) is implemented for helping effectively discover cloud services, relevant service features and service costs that are provided by the cloud service providers.

Keywords: Service discovery, cloud service, classification, service identification

1 Introduction

Cloud computing has been growing rapidly in the past few years. It is a relatively new computing paradigm that has the capability to deliver several services on demand. In cloud computing, users are able to share a large pool of computing resources over the Internet with modest cost. Regardless of the computing

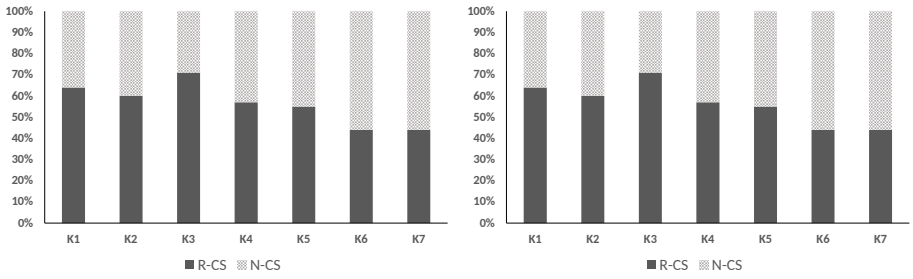
resources' quantity, location, and time, users can access the desirable computing resources [2]. In the area of cloud services, many research efforts have been conducted to handle security [10], privacy [9], and trust management [7], but cloud service discovery is still encountering challenges in terms of finding appropriate services to cloud users on the World Wide Web. With cloud computing, service discovery faces new challenges on the Internet due to a number of reasons. Firstly, cloud services offer different service functions, e.g., processing data, building business logics, and supporting infrastructure capabilities. Secondly, recent research has found that only 2% cloud service providers publish their services following the Web Services Description Language (WSDL). And the rest providers publish their services without considering any standards to describe their services and resources [8]. This has made cloud service discovery very challenging. Compared with Web services discovery, Web services in general use standard languages, such as the Web Services Description Language (WSDL), Unified Service Description Language (USDL), to expose their interfaces. Thirdly, the variety of Service Level Agreements (SLAs) between cloud service users and cloud service providers increases the difficulty of discovering cloud services.

Discovering service is widely considered an essential problem in many research areas such as ubiquitous computing, mobile networks, peer-to-peer (P2P) services, and service oriented computing [6, 14]. In the past decade, service discovery has been a very active research area, particularly in Web services [13]. However, for cloud services, challenges need to be reconsidered and solutions for effective cloud service discovery are very limited.

To find a cloud service on the World Wide Web, a potential user normally relies on a general-purpose search engine to find a suitable cloud service. However, using such kind of search engine for this purpose is a tedious task because the search results provide large quantity of irrelevant cloud service search results, including news, blogs, journal papers, wiki, and articles, etc. This is because the term "Cloud" is a very general and widely used terminology. Therefore, the difficulty of discovering cloud services on the World Wide Web arises as a big and challenging issue. For example, we can easily see that cloud is one of the most important and popular terminologies in websites about meteorology. Moreover, many websites talking about cloud service are not necessary the provider of any cloud service. Some businesses also have nothing to do with cloud computing but they may use cloud in their names or service descriptions (e.g., cloud9carwash[1]). Fig 1 shows the first 100 search results from a current general-purpose search engine (Google or Bing) using different Keywords, such as cloud service, cloud storage, cloud service provider, cloud hosting, cloud software, cloud platform or cloud infrastructure to search for cloud services. Furthermore, general search engines are very weak in providing details about cloud service features (e.g., cloud service type, process limitation, storage maximums, memory capacity, and so on). Considering all these limitations of general-purpose search engines, in this

[1] http://www.cloud9carwash.com/

work, we aim to design a cloud service search Engine to alleviate the aforementioned issues.



(a) Google (b) Bing

Fig. 1. The first 100 searching results from two most popular search engines (Google and Bing) using different keywords, such as cloud service, cloud storage, cloud service provider, cloud hosting, cloud software, cloud platform and/or cloud infrastructure to search cloud services

In this paper, we focus on the design and implementation of a cloud service search engine (CSSE). CSSE helps distinguish between cloud services and other services available on the Internet. Furthermore, CSSE provides more details of a service and its features which can support cloud service search users on how to identify an appropriate cloud service towards their needs. The two main components of CSSE include i) a cloud service identifier and ii) a cloud service feature extractor. This identifier helps identify cloud services during the process of cloud service discovery and the process of determining a cloud service features. Moreover, the cloud service identifier can automatically identify cloud service by utilizing a classification method. Then, the feature extractor determines/extracts a cloud service's features using a cluster method and a novel approach, called Service Tracking and Detection (SDT), to detect and track other services. Finally, we can extract cloud service's features that can be used to facilitate the searching process. In a nutshell, the contributions of our work are as follows:

- We design and develop a cloud service search engine to provide highly accurate cloud service search results and provide useful details about cloud services' features, which can facilitate cloud service selection from search users.
- The cloud service identifier is built by utilizing cloud service features extracted from real cloud service providers.
- Inspired by the Topic Detection and Tracking [1], we propose a novel Service Detection and Tracking (SDT) methodology for the detection of cloud services.

- We build a unified model to expose the cloud service's features to a cloud service search user to ease the process of searching and comparison among a large amount of cloud services.
- We conduct extensive experiments to validate our proposed approach. The results demonstrate the applicability of our approach and its capability of effectively identifying and extracting cloud services' features from the World Wide Web.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents our search engine (CSSE) architecture, including details of cloud service identification and extraction of cloud service features. Section 4 provides an implementation and our experimental study of our CSSE search engine. Finally Section 5 offers concluding remarks.

2 Related Work

Nowadays, the most popular approach for discovering cloud service refer ontology. Youseff et al. [16] classify cloud computing based on its components, consisting of five layers: the applications, the software environment, the software infrastructure, the software kernel, and the software hardware. Each layer contains one or more services depending on the level of abstraction. Further, each layer relies on computing concepts to measure limitations and strengths. Another attempt involves building an ontology based on the cloud business ontology model. Kang and Sim [4] propose a cloud service discovery system that uses an ontology-based approach to discover cloud services close to users' requirements. However, cloud service providers still need to register at the discovery system in order to publish their cloud services. Furthermore, their work relies on software agents to perform reasoning tasks (e.g., similarity reasoning, equivalent reasoning and numerical reasoning). Yoo et al. [15] select a cloud service that best meets a user's requirements by using a cloud ontology based on resource services. The authors use the similarity computing degree of virtual cloud service physical resources to determine the best cloud service for users. Dastjerdi et al. [3] propose an approach that uses ontology-based discovery for QoS-aware deployment of appliances on IaaS providers. This approach support end user to meet their needs from range of IaaS providers based on QoS preferences. However, the ontology design only find the suited IaaS providers for end users. Furthermore, the ontology does not support PaaS and SaaS providers.

Ma et al. [5] propose ontology-based resource management of cloud providers. This cloud computing ontology defined the concepts that described their relations. However, the ontology has to meet cloud service requirement and has been conducted in simulated environment. Rodrguez-Garca et al.[12, 11] exploit an automatic general ICT domain that can be used to discover the cloud services best matching user needs. The authors use semantic annotation in order to improve the cloud service discovery results. From cloud service descriptions, semantic content can be extracted by using the annotation platform. Then, the

semantic content can be used by the semantic search engine to assist users in finding those services that meet with their requirements and expectations.

In summary, these works did not consider the problem of how to use the cloud service web content to automatically identify cloud service. Moreover, they did not provide adequate details about features of cloud services on the Internet. To alleviate these issues, in this paper we aim to design and implement a cloud service search engine to provide highly accurate cloud service search results and provide useful details about cloud services' features, which can facilitate cloud service selection from search users.

3 Overview Cloud Service Search Engine

In this section, we first introduce our cloud service search engine (CSSE), then spotlight on describing our approach on cloud service identification and building cloud service profile.

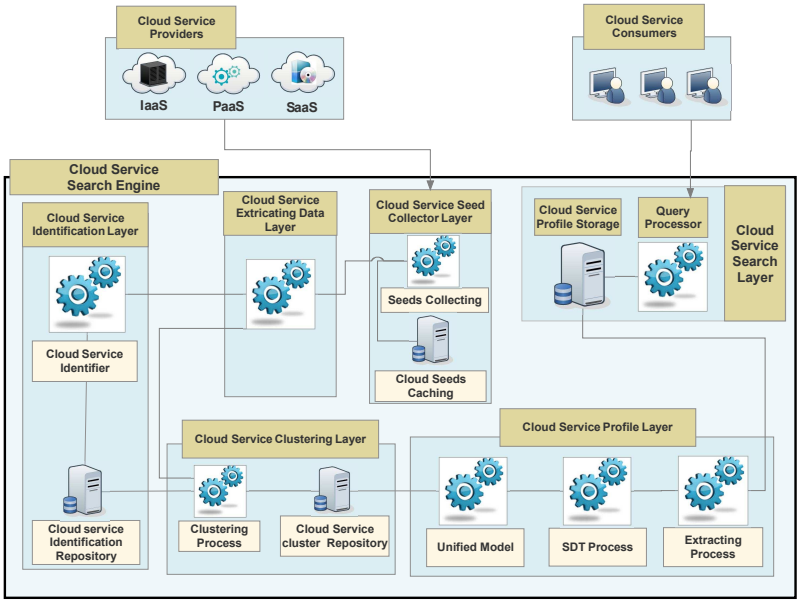


Fig. 2. Cloud Service Directory

3.1 CSSE Architecture

Fig 2 depicts the main components of the cloud service search engine (CSSE), which consists of six major layers: namely (1) *Cloud Services Seeds Collection Layer*, (2) *Cloud Services Extracting Data Layer*, (3) *Cloud Services Identification Layer*, (4) *Cloud Services Cluster Layer*, (5) *Cloud Services profile Layer* and (6) *Search Engine User Layer*.

Cloud Services Collector Layer: This layer is responsible for collecting possible cloud service seeds (i.e., the cloud services' URLs) in real environments. We initially collect cloud service seeds using two approaches. Firstly, we develop the cloud service source collector module that is able to collect cloud services automatically by crawling Web portals and indexes on search engines, such as Google, Bing, and Baidu. Secondly, we develop the cloud service seed inquiry based module that has the capability to inquire a cloud service and determine whether this cloud service has been cached in cloud service seeds repository. This inquiry can be done by both cloud service customers and cloud service providers. Furthermore, if the cloud service users inquire about a seed registered in the system, the system can return the inquiry result directly. Otherwise, the seed will be sent to the Cloud Services Extracting Data Layer for obtaining the essential details that can determine if the seed provides the cloud service.

Cloud Services Extracting Data Layer: This layer is responsible for extracting essential content in the cloud service source such as description, keywords, text content and hyper links. The cloud services' content lead to support of building automation cloud service identifier. The cloud services' content can be achieved automatically by filtering the cloud service source (i.e, cloud service source html homepage to text). Then the cloud services' content is sent to Cloud Services Identification Layer while cloud service hyper links sources can be sent to Cloud Service Cluster Layer if the source pass the Cloud Services Identification Layer.

Cloud Services Identification Layer: This layer is responsible for identifying cloud service provider. The Cloud Services identifier contains the process of identifying features to determine whether a given source is cloud service or not. This processing relies on classification method to realize the identification. The identification can be updated automatically after identifying a new cloud service provider to enhance identifying knowledge. Furthermore, the identifier only focuses on cloud computing which does not include cloud mobile computing. However, our cloud service directory can avoid the lack of success in identifying a cloud service source. Because it allows cloud service provider to register their services in our system, these cloud service sources can be added and can be recognized by the identification as a new cloud service provider.

Cloud Services Clustering Layer: This layer is responsible for clustering cloud service providers. the cloud service clustering is able to collect to the most similar cloud service into clusters based on clustering method. The clusters are built depend on two features which are cloud services' text and cloud services' hyper-links. Firstly, using cloud services' text can lead to finding the most similar cloud services into one cluster. Secondly, cloud service hyper-links can lead to detecting the exact services the cluster provide. This clustering approach is able to decrease the distance during detecting the service and tracking.

Cloud Services Cloud Profile Layer: This layer is responsible for generating cloud service profile based on the following processes:

1. *Modelling*: this process is responsible for building cloud service model. the model is built based on the service features. In addition, we observe several cloud services to identify the service features such as type, price, capacity. Moreover, this model is used to find the service in a cluster by investigating about the features. The process begins by selecting a cloud service provider and determine the service features that provide. Then, we build JSON model for this service. The JSON model includes many details about the cloud service which are the cloud service features. More details can be shown in Section 3.4.
2. *Detecting and Tracking*: this process is responsible for detecting and tracking other cloud services based on service feature model. This processing can search for the service inside the cluster by taking the JSON model which is built for the service and track this model. The process of detecting and tracking can investigate the whole cloud service websites to find the service. After we find the service we directly build the cloud service model for the cloud service.
3. *Extracting and Storing*: this process is responsible for extracting and storing the cloud service JSON model. The details of cloud services can be received from Detecting and Tracking process. Then, we can store this details in JSON model to support in building a cloud service search engine. We update this process weakly to discover any different in the cloud service features.

Search Engine User Layer: This layer provides a Web interface for users to search cloud services. A user can simply specify a searching keyword for finding cloud services. She can also specify other constraints (e.g., categories like IaaS) to narrow down the searching scope. Our system will contact the cloud service service profile repository. If it is found in the repository, the detailed information (e.g., access link, features, description) of satisfied cloud services will be returned to the user.

3.2 Cloud Service Identification

In this section we demonstrate our approach to identify real cloud service providers. The proposed approach is a task that uses both information retrieval and machine learning techniques to identify cloud services. The approach of identifying task consists of number of steps. Firstly, we aim to build documents corpus which is cloud service sources. Therefore we assemble a large collection of cloud services providers' homepage and other homepage are highly related to cloud service but they are not real cloud services. This documents corpus is generated from the cloud services' homepage $S = \{s_1, ..., s_n\}$. Moreover, it exploits 5882 real cloud service and 5000 not cloud services to build classification method. Then, a document matrix is built to include the documents and is vectorized

each document using the following weighting function $t = tf/(tf - td)$ where tf denotes the term frequency, $tf - td$ the the other total terms appear in the documents $s = \{t_1, ..., t_n\}$. Since we achieve highly dimension of terms, we try to reduce the dimensionality of the terms by using Latent Semantic Analysis (LSA) which is implemented using randomized Singular Value Decomposition (SVD) to build the document matrix. After we have built the document matrix we consider each term that uses weighted function as features. Finally, we utilize the cloud service text features to apply the classification method. We use k-Nearest Neighbor Classification which uses a Given data matrix of cloud services' terms $T = \{t_1, ..., t_n\}$ with K classifier and t vector $s \in S$. This classifier can find the K nearest class to cloud service vector s. The classification method can be helpful in distinguish between cloud service providers and non cloud services.

3.3 Cloud Service Clustering

Since using cloud service text features can be used to support identifying cloud service, identifying the cloud service features is a totally different task and need to extract new features and using different techniques. Therefore, we use clustering approach to find the service features by adding the cloud services' web page hyper-links as features. The cloud services hyper-links features are used to support in building cloud service profile features. In addition, we consider each hyper-link occurring in cloud serviced home page as features to build the clusters. This clustering can assign each cloud service provider as nearest as possible to ease discovering the service type and its features. The process of clustering consists of two methods. In the beginning, the clusters are built based on the term vectors that are weighted by term frequency.. After we cluster the cloud services' terms. Then, we use the cloud service hyper-link $s = \{h_1, ..., h_n\}$ as features weighted by *Term Frequency* that shows in cloud services home page. Then, we apply K-Means which gives a number of clusters K that desirables and improved iteratively the Euclidean distance between each data point and the centroid nearest to it in our experiments. This processing is able to decrease the distance during applying Service Detection and Tracking process because it shows that the similar type of cloud service more frequently appear in the same cluster.

3.4 Cloud Service Profile

In this section we demonstrate the process of generating cloud service profile. The cloud service profile process will detect a service and tracking cloud services. The process consists of three phases, including Cloud Service Modelling, Service Detection and Tracking and Cloud Service Extracting and Storing. Algorithm 1 describes the cloud service profile processes.

Cloud Service Modelling: Cloud service modelling is embedded in cloud service features. This cloud service features have been achieved by observing several

Algorithm 1: Cloud Service Profile Algorithm

Input: A set of cloud service sources belong to a cluster $\mathbb{C} = \{S_1, S_2, \ldots\}$, HF vectors $\overline{s_1}, \overline{s_2}, \ldots, \overline{s_{|\mathbb{C}|}}$ for each cloud service source in $\mathbb{C}$, the number of nearest cloud service is K, the target cloud service source S_t

Output: $\mathbb{NS}$: a subset which contains K neerest cloud service sources of S_t from $\mathbb{C}$

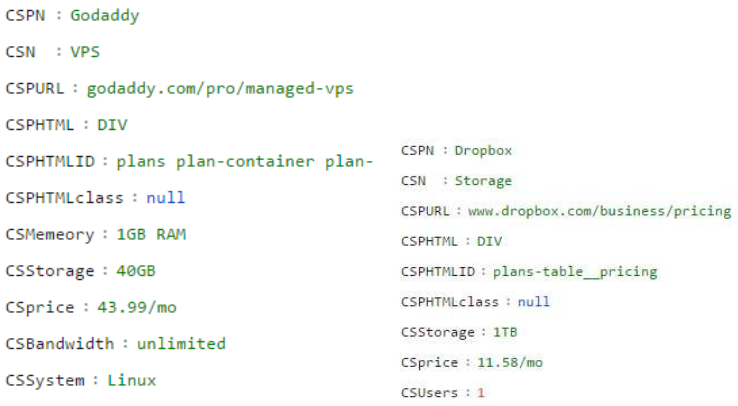
```
NS ← ∅;
while C is not empty do
    NS ← ∅;
    Add S_t to sub set cluster NS;
    Remove S_t from C;
    for each S_i ∈ C do
        Compute cosine(S_t, S_i) based on HF vectors s_St and s_j for cloud service
        sources S_t and S_j;
        Add S_i to subset NS;
        Sort NS based on similarity score
        Pick up top K cloud service sources in NS
    end for
end while
```

Constant Features	**Variable Features**
Cloud Providers	Cloud Service CPU Name
Cloud Service Name	Cloud Service CPU Capacity
Cloud Service URL	Cloud Service Memory Capacity
Cloud Service HTML Tag	Cloud Service Storage Capacity
Cloud Service HTML ID Tag	Cloud Service Storage type
Cloud Service HTML Class Tag	Cloud Service Price

Table 1. Cloud Service features

of cloud services in real environments. In addition, the cloud services features are different from service to service. For example, VPS features are totally different from storage features but they might share some common features see Fig 3. However, we add some of constant features for all cloud service types that are provided by cloud service providers. Table 1 shows constant features and variable features.

Service Detection and Tracking: In the *Topic Detection and Tracking* (TDT) a topic is defined as a seminal event or activity, along with all directly related events and activities [1]. In order to replace the cloud service instead of topic, a cloud services is defined as a set of utilities that can provide several services. In addition, this services can be shared with several of providers. Therefore, the Service Detection and Tracking is a novel approach has been built to discover a service then track the same type of service over punch of cloud service providers. The service detection and tracking is able to compare the features of



(a) Cloud Service VPS Godaddy (b) Cloud Service Storage Dropbox

Fig. 3. Cloud Service Modelling

different services. For example, if we pick up a service from a cluster and we detect this service is VPS then we will track this service inside the cluster based on cosine similarity score of hyper-links features because we assume this is VPS cluster. However, the service tracking is not easy task because each cloud service providers describe its service under different hyper-links. However, we observe many cloud services describe their service features under hyper-links such as plan, price, features or the name of service. Therefore, when we track a service we target to investigate those four hyper-links if they are available on cloud services' homepage, otherwise we can manually discover the cloud service.

Cloud Service Extracting and Storing: The aim of this processing is to extract cloud service features from cloud services inside the cluster. Therefore, we search for this cloud service features $s = \{f_1, ..., f_n\}$ inside the cluster then we start collect the cloud services' features. Our searching method uses initial point to search inside cloud service page which can be the plan, price, features or the name of service. This initial can be determined depend on the features that we already made. For example, if we search about VPS, we can investigate about VPS features such as memory capacity, storage capacity, CPU type, price and operating system. In addition, if we find these features we can add it to JSON model otherwise it become null.

4 Evaluation

4.1 Dataset: CSCE 2013

We use real-world cloud service sources metadata of the 5,883 valid cloud services and 5,000 valid non-cloud services in our evaluation. This data was chosen

because it has been verified by our previous research work [8]. This dataset can easily provide labeling for our classification model. In our work, these Web sources of cloud services and non-cloud services have been processed in two phases. In the first phase, cloud service sources were fetched as HTML file. Then, cloud service hyper-links were extracted. In the second phase, cloud service sources were filtered to obtain only the cloud services sources text. In our experiments, we removed the HTML tags and non-English cloud services in the cloud services sources. To eliminate and filter the HTML tags, we used the HTML Parser[2]. Next, we eliminated non-English cloud services metadata using a language detection library[3]. Finally, the text features of the cloud service and non-cloud service sources only contain English language cloud services. Moreover, we discarded any cloud service sources with less than 30 terms to enhance our identification process. Then, we filtered the cloud services sources metadata and finally obtained 4,397 cloud services sources and dismissed 845 cloud services source text descriptions. Our final dataset comprises a total of 4,397 cloud services sources and 4,030 non-cloud services which has been divided into i) 75% training data to build cloud service features model and ii) 25% to test cloud service features. By using cloud services features, we ran the experiments to generate cloud service identifying model effectively. Then, we conducted the second experiment to show our model can effectively predict and detect cloud service.

4.2 Identifying Cloud Services

Based on our classification model, we ran the system to generate our cloud services identification by using the K-Nearest Neighbour. Firstly, we built cloud service corpus that contain 3,297 real cloud services and another non-cloud service corpus that includes 3,023 non-cloud services. Then, we achieved 987,457 terms from cloud service document matrix with high sparsity. However, after we removed the term sparsity the cloud service document matrix is still high dimension. Therefore, we used Latent Semantic Analysis (LSA) to reduce the dimension cloud document matrix and built our term features based on LSA concept. We used different number of K concepts and we ran our classification method. In the beginning, we ran the experiment with small number of K concepts and we gradually increase the K concepts. Fig 4 show the accuracy of term features using various k-nearest neighbor. In addition, it obtained high accuracy if we increase the k in k-nearest neighbor which achieved 86% with 6-nearest neighbor and increase the K concepts. In addition, we can see that the 500 concepts are obtained high accuracy with 6-nearest neighbor. However, we can notice that increasing the K of concepts can lead to decrease the identifying process because increasing of terms drives our identification to increase the noise data.

We also conducted an experiment to identify cloud services for proving the precision and recall of our proposed cloud service identifier. In this experiment

[2] http://htmlparser.sourceforge.net
[3] https://code.google.com/p/language

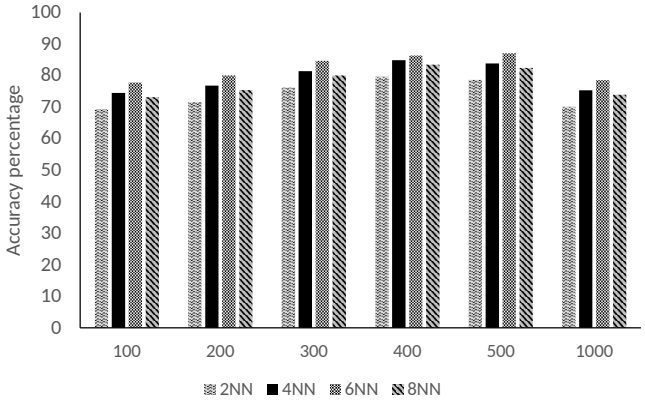


Fig. 4. Term features classification accuracy (%) using k-nearest neighbor classification, with LSA dimension reduction cloud service.

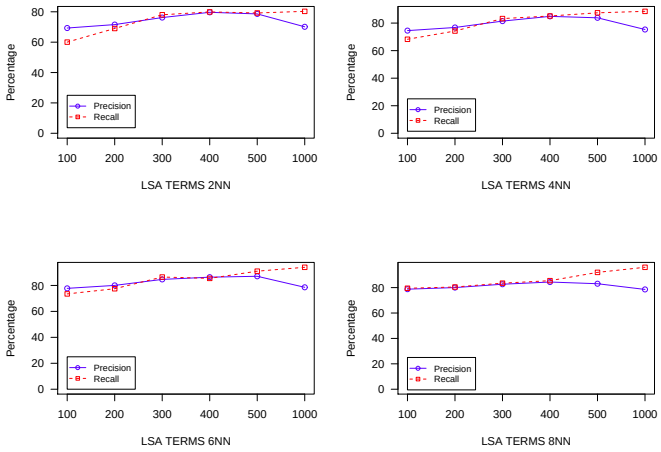


Fig. 5. The precision and recall using k-nearest neighbor classification, with LSA dimension reduction cloud service

(see Fig 5 for the results), We used LSA terms that extracted from cloud service corpus then we run k-nearest neighbor model to determine the precision and recall. Moreover, the precision represents the percentage rate of distinguishing between cloud service or non-cloud service, whilst recall represents the percentage of cloud services identified. At the beginning, the experiment is started with 100 LSA terms then we increase the number respectively. We can see high relation between the number of LSA terms and precision and recall. we find that the increasing the number of LSA terms can reduce out model precision but it lead to increase the recall percent. Further more, we find that increasing the the K number can lead to increase the recall sharply.

4.3 Cloud Service Profile

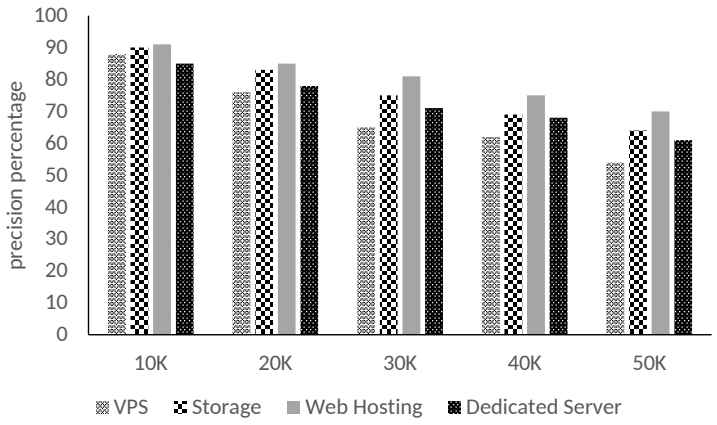


Fig. 6. Cloud Service Profile

In order to determine the service, we ran our cloud profile algorithm to detect and track the service. This algorithm can search for a specific service over punch of cloud service providers. To ensure the accuracy of the results, we have labeled the cloud services sources manually before we ran our algorithm. Then, we compared our algorithm with manual processing. Fig. 6 shows the precision result of cloud profile algorithm. We can see that if we increase the number of cloud service sources we increase the difficulty of detecting the service features because the precision is decrease. However, we find that service which are popular in real environments such as cloud VPS and Web Hosting can be found precisely by our algorithm. Furthermore, we observed that some cloud service providers show their service features under alternative hyper-links such as plan, features and price.

Fig 7 shows that where mostly we can find that the cloud service features inside the cloud services. From Fig 7 we found the 36% of cloud service describe

their services under the service name hyper-links while 21% describe under the features hyper-links. More over, 18% of cloud service providers described their services under price hyper-link while only 9% of cloud services used plan hyper-links to describe their services. However, 18% of cloud service providers describe their services under specific name for the service which leads to difficulty in finding the service and its features.

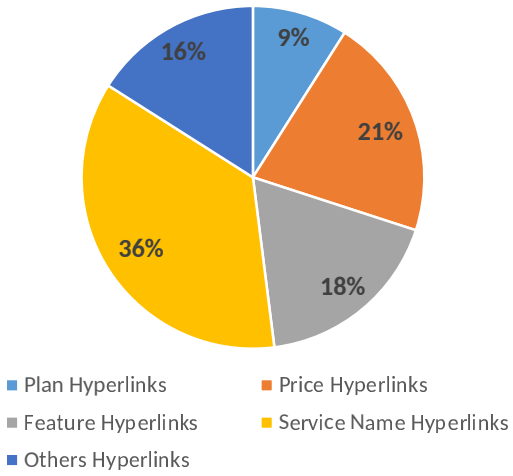


Fig. 7. Cloud Service Profile Features

5 Conclusions

With the growing adoption of cloud computing, efficiently finding relevant cloud services for customers is becoming a critical research issue. Unique characteristics of cloud services such as lack of standardization, diverse and dynamic services at different levels, make cloud services discovery a very challenging task. In this paper, we have conducted a comprehensive analysis of the cloud services currently available on the Web. We have developed a cloud service search engine that collects, extracts, and identifies cloud services. We have also provided details about cloud service features. Based on the cloud service identifier information, we have provided in-depth statistical analysis including the accuracy on cloud providers under current search engine and the relationship between a cloud service and its Web page profile. These results offer an overall view on the current status of cloud services in the World Wide Web. The most intriguing finding is the fact that cloud service web pages play a significant role in discovering cloud services and applications. Furthermore, we have proposed a novel approach to extract the service features based on our Service Detection and Tracking model which significantly increases the accuracy of identifying the service features.

References

1. Allan, J.: Topic detection and tracking: event-based information organization, vol. 12. Springer Science & Business Media (2012)
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al.: A View of Cloud Computing. Communications of the ACM 53(4), 50–58 (2010)
3. Dastjerdi, A.V., Tabatabaei, S.G.H., Buyya, R.: An Effective Architecture for Automated Appliance Management System Applying Ontology-based Cloud Discovery. In: Proc. of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010). pp. 104–112 (2010)
4. Kang, J., Sim, K.M.: Cloudle: An Ontology-Enhanced Cloud Service Search Engine. In: Proc. of the 2010 International Conference on Web Information Systems Engineering (WISE 2010). pp. 416–427 (2010)
5. Ma, Y.B., Jang, S.H., Lee, J.S.: Ontology-based Resource Management for Cloud Computing. In: Intelligent Information and Database Systems, pp. 343–352. Springer (2011)
6. Meshkova, E., Riihijärvi, J., Petrova, M., Mähönen, P.: A Survey on Resource Discovery Mechanisms, Peer-to-Peer and Service Discovery Frameworks. Computer Networks 52(11), 2097–2128 (2008)
7. Noor, T.H., Sheng, Q.Z.: Trust as a Service: A Framework for Trust Management in Cloud Environments. In: Proc. of Web Information System Engineering (WISE 2011), pp. 314–321. Springer Berlin Heidelberg (2011)
8. Noor, T.H., Sheng, Q.Z., Ngu, A.H., Dustdar, S.: Analysis of Web-Scale Cloud Services. IEEE Internet Computing 18(4), 55–61 (2014)
9. Pearson, S., Benameur, A.: Privacy, Security and Trust Issues Arising from Cloud Computing. In: Proc. of IEEE Second International Conference onCloud Computing Technology and Science (CloudCom 2010). pp. 693–702 (2010)
10. Ren, K., Wang, C., Wang, Q.: Security Challenges for the Public Cloud. Internet Computing, IEEE 16(1), 69–73 (2012)
11. Rodríguez-García, M.A., Valencia-García, R., García-Sánchez, F.: Creating a Semantically-enhanced Cloud Services Environment Through Ontology Evolution. Future Gener. Comput. Syst. 32, 295–306 (Mar 2014)
12. Rodríguez-García, M.Á., Valencia-García, R., García-Sánchez, F., Samper-Zapater, J.J.: Ontology-based annotation and retrieval of services in the cloud. Knowledge-Based Systems 56, 15–25 (2014)
13. Segev, A., Sheng, Q.: Bootstrapping Ontologies for Web Services. IEEE Transactions on Services Computing 5(1), 33–44 (2012)
14. Wei, Y., Blake, M.B.: Service-Oriented Computing and Cloud Computing: Challenges and Opportunities. IEEE Internet Computing 14(6), 72–75 (2010)
15. Yoo, H., Hur, C., Kim, S., Kim, Y.: An Ontology-Based Resource Selection Service on Science Cloud. In: Grid and Distributed Computing, vol. 63, pp. 221–228 (2009)
16. Youseff, L., Butrico, M., Da Silva, D.: Toward a Unified Ontology of Cloud Computing. In: Proc. of 2008 Grid Computing Environments Workshop (GCE 2008). pp. 1–10 (2008)