Soft Computing-based Methods for Semantic Service Retrieval

Supannada Chotipant

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY
UNIVERSITY OF TECHNOLOGY SYDNEY

A thesis submitted in fulfilment of the requirements for
the degree of Doctor of Philosophy

July 2017

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature: ....

Date: ....

ACKNOWLEGEMENT

First of all, I would like to express my sincere gratitude to my principle supervisor, Dr Farookh Khadeer Hussain. My priceless journey started when he accepted me to be his PhD student in August 2013. He has given me not only excellent guidance in my research, but also great support and encouragement. This really helps me to finish my research and learn how to be a good researcher.

Additionally, I would like to thank my parents who always stand beside me and walk through all bad situations with me. I am who I am because of them, so it is undoubtedly that this success also belongs to my parents. I cannot be here without their endless care, support, and encouragement. Likewise, I would like to thank my family, Prof. Tony Moon, and my friends who always help me to get over all the bad situations.

Finally, I dedicate this thesis to Dr Farookh Khadeer Hussain, who is a key person of this work, my parents, and my family.

ABSTRACT

Nowadays, a large number of business services have been advertised to customers via online channels. To access the published services, the customers typically search for the services by using search engines. Consequently, in order to meet the customers' desires, many researchers have focused on improving performance of the retrieval process. In the recent past, semantic technologies have played an important role in service retrieval and service querying. A service retrieval system consists of two main processes; service annotation and service querying. Annotating services semantically enables machines to understand the purpose of services, while semantic service querying helps machines to expand user queries by considering meanings of query terms, and retrieve services which are relevant to the queries. Because of dealing with semantics of services and queries, both processes can further assist in intelligent and precise service retrieval, selection and composition. In terms of semantic service annotation, a key issue is the manual nature of service annotation. Manual service annotation requires not just large amount of time, but updating the annotation is infrequent and, hence, annotation of the service description changes may be out-of-date. Although some researchers have studied semantic service annotation, they have focused only on Web services, not business service information. Moreover, their approaches are semi-automated, so service providers are still required to select appropriate service annotations. Similar to semantic service annotation, existing literature in semantic service querying has focused on processing Web pages or Web services, not business service information. In addition, because of issues of ubiquity, heterogeneity, and ambiguity of services, the use of soft computing methods offers an interesting solution for handling complex tasks in service retrieval. Unfortunately, based on the literature review, no soft-computing based methods have been used for semantic service annotation or semantic service querying. In this research, intelligent soft-computing driven methods are developed to improve the performance of a semantic retrieval system for business services. The research includes three main parts, namely, intelligent methods for semantically annotating services, querying service concepts, and retrieving services based on relevant concepts. Furthermore, a prototype of a service

retrieval system is built to validate the developed intelligent methods. The research proposes three semantic-based methods; ECBR, Vector-based and Classification-based, for accomplishing each research part. The experimental results present that the Classification-based method, which is based on soft-computing techniques, performs well in the service annotation and outperforms both the ECBR and the Vector-based methods in the service querying and service retrieval.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PUBLICATIONS

1. Chotipant, S., Hussain, F.K., Dong, H. & Hussain, O.K. 2014, 'A Fuzzy VSM-Based Approach for Semantic Service Retrieval', *Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part III*, Springer International Publishing, Cham, pp. 682-9.

2. Chotipant, S., Hussain, F.K., Dong, H. & Hussain, O.K. 2015, 'A Neural Network Based Approach for Semantic Service Annotation', *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part II*, Springer International Publishing, Cham, pp. 292-300.

3. Chotipant, S., Hussain, F.K. & Hussain, O.K. 2015, 'An automated and fuzzy approach for semantically annotating services', *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pp. 1-7.

4. Chotipant S, Hussain FK, Hussain OK. 'SERNOTATE: An automated approach for business service description annotation for efficient service retrieval and composition', *Concurrency Computat: Pract Exper. 2017*:e4189.

Chapter 1 Introduction

1.1 Introduction

In this chapter, an overview of the importance of semantic service retrieval is presented in Section 1.2 . Service annotation and service querying are considered and applied to accomplish the service retrieval task. In Section 1.3 , the issues that are related to the semantic service retrieval, including the semantic service annotation, semantic query expansion, and semantic service retrieval, are discussed. Based on the research issues, the proposed objectives of the thesis are presented in Section 1.4 . The scope and conditions of this thesis are defined in Section 1.5 . Then, Section 1.6 presents the contributions of the thesis, which are divided into scientific contributions and social contributions. To complete the proposed semantic service retrieval methodology, various semantic-based tasks are required. Consequently, the outline of the thesis (which is organized into nine chapters) is presented. The summary of each chapter is described in Section 1.7 . Finally, the conclusion of this chapter is presented in Section 1.8 .

1.2 Importance of the service retrieval

World Wide Web (WWW) has played an important role for worldwide interactive communications among people via the Internet. The developments of the Internet and WWW have widely assisted in much improvement in many areas, such as business, education, government, and industry. In terms of the business area, the web-based environment enables businesses to reduce costs of product advertisements, easily and rapidly communicate with customers and partnerships, and globally publish their products. Consequently, the traditional business network environment, which is close, rigid and centralized or distributed based collaboration, has been moved to the digital network environment called Digital Ecosystems which is an open, flexible, domain clustering and demand-driven based collaboration (Boley & Chang 2007; Chang & West 2006).

The Digital Ecosystems are the virtual economic ecosystems which are comprised of two major elements; namely species and fundamental services and technologies. Species in the Digital Ecosystems are divided into three types; namely biological species like humans, economic species like business organizations and digital species like computers, software, and applications. Consequently, services in the Digital Ecosystems are comprised of human-based services, organization-based services, and application-based services. The main idea of the Digital Ecosystems is that species are able to link, communicate and collaborate with each other based on supporting services and technologies in order to reach their business goals with the optimal solution.

To support the collaboration in the Digital Ecosystems, the species should be able to find their applicable species to work with. That is, customers who want to use services can search for appropriate service providers; meanwhile, business organizations can search for their suitable business partners or business alliances. The challenge of this work is how to deal with the different features of the services as follows (Dong et al., 2012):

- *Ubiquitous*: A vast number of business services have been published in various service registries which are geographically dispersed over the Internet. Consequently, researchers need to consider how to correctly and quickly retrieve services from various distributed registries.
- *Heterogeneous*: There is no standard format for modeling and representing business services in the Digital Ecosystems. As a result, it is complex to search and combine desired services from various formatted services.
- *Ambiguous*: Content of services is described in natural language. As a result, the services are ambiguous because the terms used in the services have different senses and meanings. To precisely retrieve services, semantics of services also need to be considered.

Consequently, the development of an effective service retrieval methodology is a complex task and is not easy to deal with. As a result, the research topic about service retrieval in the Digital Ecosystems is interesting and still needs to be explored.

1.3 Issues related to semantic service retrieval

Nowadays, many businesses provide their own service information to customers via online channels, such as websites, mobile applications, and social media. Normally, if the customers desire to access the business service information on the Web, they need to search the services by web search engines, such as Google, Yahoo or Bing. Although these search engines work properly for web document retrieval, they are not designed for searching service information particularly (Dong, Hussain & Chang 2011). Consequently, a large amount of irrelevant information is frequently retrieved to customers. Furthermore, current web search engines are unsuitable for searching and combining services based on customers' goals. For example, a traveler in Thailand wants to search services in all service categories about traveling in Australia such as air ticket, car rental, and accommodation services. In this case, the traveler needs to search for each service category separately (air ticket; car rental; hotel accommodation separately) and combines related services from all service categories by himself. This leads a framework that searches and combines business services to be required.

Semantic technologies have played an important role in data and resource discovery (Eftychiou, Vrusias & Antonopoulos 2012; Liu et al. 2007), indexing (Lai & Moulin 2013), querying (Awad, Polyvyanyy & Weske 2008; Wlodarczyk et al. 2012) and integration (Jin et al. 2008; Wang, Huang, et al. 2009) in various research areas. The purpose of using semantic technologies is to get meanings of data and resources. This assists users and machines to understand the content; as a result, they are able to properly retrieve, share and combine resources.

The focus of this thesis is on the semantic-based methodology for business service retrieval. That is, the methodology is able to semantically retrieve services that are relevant to a given query. The main idea of this thesis is to represent the semantics of services and queries by using service ontology. Then, the methodology semantically retrieves relevant services by matching semantics between services and queries. To accomplish this task, the semantic service retrieval task is partitioned into three principal topics, namely, semantic service annotation, semantic service querying, and semantic service retrieval. The issues that are related to those topics are described in Section 1.3.1 , Section 1.3.2 , and Section 1.3.3 respectively.

1.3.1 Issues related to semantic service annotation

Semantic service annotation enables users and machines to understand the purpose of services and can further assist in intelligent and precise service retrieval, selection and composition. Currently, meanings of services are manually described by the service providers. Although manual service annotation reflects accurate meaning, it requires a large amount of time on the service providers. Moreover, the service providers seldom update the annotation. Consequently, the annotation may get out-of-date. For example, a large number of travel agent services which have been provided in the public always update the information about package tours and promotions according to traveling seasons. As a result, manually updating the service annotations is time-consuming and unsuitable. Because of these limitations, there is a need for automated annotation approaches for services. The annotation approaches have to be triggered, whenever the services have evolved overtime. In order to semantically annotate the business services, the issues of ubiquity, heterogeneity and ambiguity of services need to be addressed. As a result, semantically annotating the services not only requires knowledge that represents the meanings or concepts in a specific domain, but also needs to deal with numerous services, diverse formats and ambiguity of service information.

Unfortunately, soft-computing techniques such as neural network and machine learning, which are well suited to such a complex annotation process (Zadeh 1994), have not been applied into the annotation process. Additionally, existing research about semantic service annotation also focuses on only Web services, not business services. The differences between web services and business services are grounded in three issues; service objective, service format, and service size, which are described in details in Chapter 3. Because of those issues, semantic annotation for business services and semantic annotation for web services are disparate.

Furthermore, most of published papers about web service annotation (Bo & Zhiyuan 2013; Canturk & Senkul 2011; Duo, Juan-Zi & Bin 2005; Lerman, Plangprasopchok

& Knoblock 2006; Liu & Shao 2010; Meyer & Weske 2006; Mokarizadeh, Kungas & Matskin 2011; Patil et al. 2004; Soon Ae & Warner 2008; Stavropoulos, Vrakas & Vlahavas 2013; Wang et al. 2010) propose semi-automated approaches which still require providers to select appropriate annotations. Consequently, those are unsuitable for the real world in which services are increasingly changed.

1.3.2 Issues related to semantic service querying

Not only the semantic service annotation, but the semantic service querying also assists in intelligent service retrieval, selection, and composition. Semantic service querying enables machines to understand the meaning of a user's query. This leads the system to retrieve service information that is exactly relevant to the user's requirement. In this thesis, the semantic service querying is comprised of two tasks; query expansion and service querying. The query expansion generates a proper query from a user query in order to get more concepts that relate to the user intention and retrieve more services. The service querying aims to get meanings or concepts; which relate to the service domain, of the expanded query. Thus, the purpose of the semantic service querying is to semantically return service concepts that are relevant to the user query.

In terms of the query expansion, the user queries are normally defined by using human language; unstructured text format. Consequently, the service querying undoubtedly faces the ambiguity and heterogeneity issues. The user queries may be comprised of ambiguous terms, which have several meanings. That is, concepts whose descriptions contain those terms but the content is irrelevant to the user intention can be retrieved. On the other hand, concepts that refer to the user requirement but do not contain query terms are ignored. To create proper queries, many researchers have proposed approaches for reformulating and expanding the queries.

Semantic query expansion is applied for semantically enlarging a query in order to increase matching resources; service concepts for this thesis. Many researchers have focused on expanding queries by using synonyms of query terms. In this case, the service querying system is able to retrieve more concepts from all senses that relate to the query terms; however, this also leads the system to retrieve more irrelevant concepts from unrelated senses of words. In addition, domain-specific ontology is also applied for the query expansion. Instead of using the synonyms, the proposed approaches use terms appearing in the ontology to enlarge the query. Those approaches are suitable for searching information in the specific areas in which technical terms, which are not in the lexical dictionary, are generally used. In this case, the performance of the querying system is increased if relevant concepts are described by technical terms. However, some service providers who publish and describe services use un-technical terms to make unfamiliar users understand what the services are. In this case, the retrieval system that uses the domain ontology based expansion approach cannot retrieve some relevant services.

In terms of the service querying, the expanded query is used for getting concepts of the original user query. Overall, the semantic service querying works in a similar way to the semantic service annotation. The semantic service annotation aims to get meanings or concepts of services, whereas the semantic service querying aims to understand concepts of queries. Similar to the semantic service annotation, soft-computing techniques have not been applied to the querying process. Additionally, existing research about semantic service querying also focuses on only Web services, not business services.

1.3.3 Issues related to semantic service retrieval

The research on semantic service retrieval focuses on how to semantically retrieve relevant services based on user queries. The effective semantic service retrieval approaches not only precisely return relevant services to the users, but also assist machines to select appropriate services and combine related services in order to achieve users' goals.

Similar to semantic service annotation, the majority of the existing approaches for semantic service retrieval focus on web services (de Castilho & Gurevych 2011; Madkour et al. 2012; Toch et al. 2007; Zhao et al. 2012), not online business services. Only Dong et al (2011) propose a semantic-based retrieval approach for online service advertising. Although the approach semantically retrieves services based on relevant ontological service concepts, the researchers expand queries by using only synonyms from WordNet. As previously mentioned, this can make the retrieval approach retrieve more irrelevant services and be unable to retrieve services that are described in technical terms. Consequently, the issue of the ambiguity of user queries needs to be solved. Regarding this issue, soft computing techniques seem to be an effective solution for semantic retrieval because they are able to handle imprecision, uncertainty, and approximate situations. Unfortunately, based on the literature review, the use of soft computing based methods for semantically retrieving services has not been proposed.

In this thesis, the focus is on soft-computing based approaches for automated business service annotation, querying, and retrieval. To validate the proposed approaches, we develop a prototype of service retrieval system in transport service ontology. Several issues that may improve the performance of service retrieval are identified in our work and solutions developed for them.

1.4 Objective of the thesis

Based on the issues related to the semantic service retrieval, the objectives of this thesis are presented as follows:

1. To develop intelligent soft-computing based methods for automatic semantic service annotation.
2. To develop semantic based methods for service query expansion.
3. To develop intelligent soft-computing based methods for semantic service querying.
4. To develop intelligent soft-computing based methods for semantic service retrieval.
5. To validate the above developed methods by building a prototype of service retrieval.

1.5 Scope of the thesis

This thesis proposes a methodology that is able to semantically retrieve services from a given query. Another two methodologies, semantic service annotation and semantic service querying, are also proposed and applied in order to accomplish the service retrieval. The purpose of the service annotation methodology is to annotate services to relevant concepts; whereas, the service querying methodology aims to query concepts that relate to the query.

Typically, the service retrieval task relates to three primary tasks; service crawling, service annotation and service querying. It should be noted here that this thesis focuses on only service annotation and service querying tasks, and the service crawling methodology is not discussed here. The thesis applies a dataset of services which was crawled from Yellow Pages by using a focused crawler provided by Dong et al (2011) because their proposed crawler is designed for crawling business service advertisements and also performs well.

In addition, services in the real world are categorized into various types, such as IT-enabled services, e-services, web services, and cloud services. It should be noted that this thesis proposes a methodology for retrieving only business services which are advertised on the Web.

1.6 Contributions of the thesis

The focus of the thesis is about an intelligent methodology for semantic annotation, semantic querying and semantic retrieval for online service advertising. In this section, the research contributions which are categorized into 2 groups - scientific contributions and social contributions are presented.

1.6.1 Scientific Contributions

1. This thesis reviews existing literature in the area of semantic service technologies, namely, semantic service annotation, semantic service crawling, semantic service querying, and semantic service retrieval for online services.
2. This research focuses on soft-computing based methods for automated semantic annotation for online service information. Initial results and experimentation demonstrate the superiority of Neural Network (NN) based soft-computing methods over other annotation approaches.

3. This research applies both synonyms and related ontological terms to expand a service query. The experimental results demonstrate that the hybrid-based query expansion methods assist in the querying performance improvement over the usage of only synonyms or ontological terms.
4. This research focuses on soft-computing based approaches for semantic querying for online service information. The experimental results show that the Classification-based approaches, especially the NN-based approaches, outperform other approaches.
5. This research focuses on a fuzzy-based approach for semantic retrieval for online service information. The experimental results demonstrate that the fuzzy NN-based retrieval approach performs much better than the non-fuzzy NN-based approach, whereas overall results of the other fuzzy-based approaches are similar to the results of the non-fuzzy based approaches.
6. This research compares the performance of various soft-computing approaches for semantic service retrieval technologies.

1.6.2 Social Contributions

1. From the perspective of service customers, being able to better understand the purpose of services and the meaning of queries, the customer can more accurately retrieve the services that are relevant to their query intentions/objectives.
2. From the perspective of service providers, because of automated service annotation, the companies can advertise their service information more effectively and do not need to worry about service annotation. The service providers do not need to define the categories of the services because the companies' service advertisement will be automatically annotated to related service concepts by the proposed methods. This increases the flexibility to advertise their service information.

3. From the perspective of the Digital Ecosystems, the proposed semantic service methodologies enable users and machines to understand the purpose of services provided in the Digital Ecosystems, and retrieve services which are relevant to user intentions. That is, the proposed methodologies are able to select suitable services and providers for the purpose of use and collaboration in the Digital Ecosystems.

1.7 Plan of the thesis

The focus of this thesis is on the semantic-based methodology for business service retrieval. To accomplish this goal, the thesis is organized into nine chapters. In this section, a brief summary of each chapter is provided as follows.

Chapter 2: Chapter 2 provides a review of existing literature in the areas that relate to semantic service retrieval methodology; semantic service annotation, semantic service querying, and semantic service retrieval. The research in semantic service annotation aims to understand the purpose of a service, whereas the research in semantic service querying is about reformulating a query by using query expansion methods. The research in semantic service retrieval aims to retrieve services that are relevant to the query.

Chapter 3: Chapter 3 defines the research problem of the semantic service retrieval system. Terminologies about the problem are firstly presented. The defined problem is divided into five research issues. The research questions and the research objectives are defined based on those issues. To solve the defined research issues, the existing scientific research methods are discussed and the selected method is described in detail.

Chapter 4: Chapter 4 presents an overview of the solution of the research questions that were addressed in Chapter 3. The definitions about the semantic service retrieval; such as service, service ontology, and service knowledge base, are firstly described. The solution overview of the main research questions is presented and the solution overviews of each research question in Chapter 3 are then separately described.

Chapter 5: Chapter 5 proposes the semantic service annotation methodology in order to automatically annotate a service to relevant service concepts. The proposed methodology consists of three modules; term extraction, service-concept matching, and service-concept connection modules. In addition, three types of semantic service annotation approaches; ECBR, Vector-based and Classification-based, are proposed. To annotate a service, those approaches follow the same modules, but the relevance value between the service and a concept is calculated by different methods. The details of each service annotation approach and its experimental results are discussed in this chapter.

Chapter 6: Chapter 6 proposes the semantic service querying methodology in order to retrieve service concepts that are relevant to a query. The proposed methodology consists of three modules; term extraction, querying expansion and querying modules. Regarding the querying expansion module, the purpose of this module is to enlarge a query based on synonyms and related ontological terms of each query term. Consequently, WordNet-based and ontology-based methods are proposed for the query expansion. Regarding the querying modules, three types of semantic service querying approaches; ECBR, Vector-based and Classification-based, are proposed. The details of each service querying approach and its experimental results are discussed in this chapter.

Chapter 7: Chapter 7 proposes the semantic service retrieval methodology in order to retrieve services that are relevant to a query. The proposed methodology consists of three modules; service annotation, service querying, and service retrieval module. The service retrieval module receives service annotation and service querying outputs from the proposed approaches in Chapter 5 and Chapter 6 respectively. To retrieve services, non-fuzzy based and fuzzy based approaches are proposed. The non-fuzzy based approach retrieves services that are annotated to queried concepts, whereas the fuzzy based approach applies fuzzy logic to calculate the relevance score between a service and the query. The details of the proposed service retrieval approaches and their experimental results are presented and discussed in the chapter.

Chapter 8: Chapter 8 presents the implementation of the semantic service retrieval methodology. Two prototypes; service annotation and service retrieval, are developed. The service annotation prototype is used for annotating service to relevant ontological service concepts. In contrast, the service retrieval prototype combines the service querying and service retrieval together. Given a query, the prototype returns relevant concepts and services to the user. Furthermore, tools and libraries that are used for developing the prototypes are also presented in this chapter.

Chapter 9: Chapter 9 concludes the thesis and summarizes the experimental results of the proposed service annotation, service querying, and service retrieval approaches. Eventually, the future work of the thesis is presented.

1.8 Conclusion

This chapter presents the introduction of this thesis which focuses on the semantic service retrieval. The importance of the semantic service retrieval is to enable species in the Digital Ecosystems to semantically find proper service providers to work or communicate with; hence this supports the species collaboration in the Digital Ecosystems. To develop the semantic service retrieval methodology, there are three primary tasks that are focused on in the thesis; namely service annotation, service querying, and service retrieval. The main issue of those tasks is that the use of soft-computing based methods for those tasks has not been proposed. Consequently, the focus of this thesis is on the soft-computing based methodologies for semantic service annotation, querying and retrieval respectively.

Chapter 2 Literature Review

2.1 Introduction

This chapter presents an overview of existing literature about semantic service retrieval. The content of this section is organized into three main parts; including 1) semantic service annotation 2) semantic service crawling 3) semantic query expansion and 4) semantic service retrieval. Existing research on the semantic service annotation relates to how to provide the meaning or the purpose of services on the Web. The research work on the semantic service crawling is about how to semantically gather service information provided on the Web. The literature in the area of the semantic query expansion focus on how to prepare an appropriate query for querying services; meanwhile, the semantic service retrieval is about how to retrieve relevant services based on a query. Existing literature that relates to those research topics is described in Section 2.2 – Section 2.5 . Then, the existing approaches of each research topic are evaluated in Section 2.6 .

2.2 Semantic Service Annotation

Nowadays, a huge number of business services have been published on the Internet because companies desire to reduce costs and easily access their customers. Annotating services semantically enables machines to understand the purpose of services and can further assist in intelligent and precise service retrieval, selection and composition. Currently, meanings are manually ascribed to most services by service providers. Although this makes the results more acceptable, it is time-consuming and there are difficulties in dealing with online tasks. The automation of the process of service annotation is therefore desirable.

This section presents existing literature in the area of the semantic service annotation which explains how to give the proper meanings or purposes of the e-services. The literature are divided into two categories; semantic service annotation for web services and semantic service annotation for other online services, which are described in Section 2.2.1 and Section 2.2.2 respectively. Moreover, each category is also subdivided into two sub-categorizes; namely, non-ontology based and ontology-based service annotation approaches. An overview of the literature review of the semantic service annotation is presented in Figure 2.1.

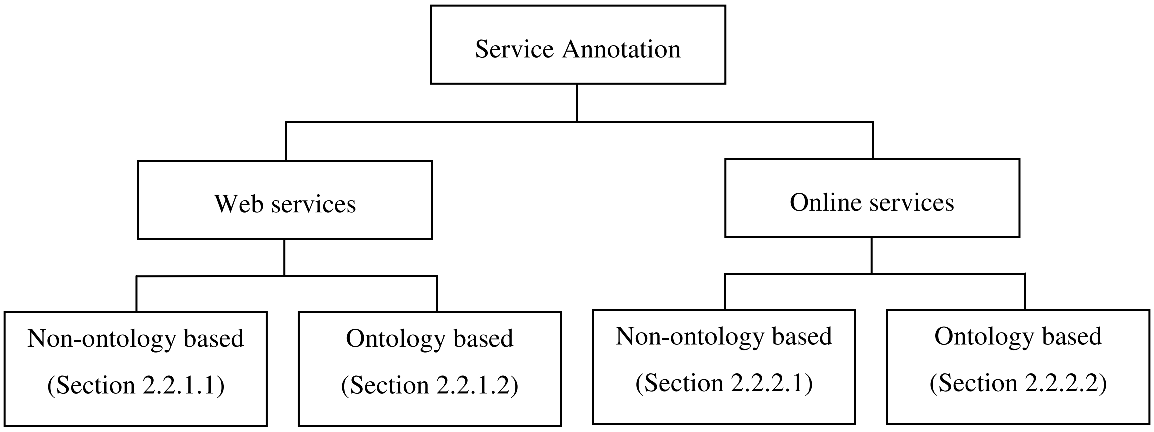


Figure 2.1 Overview of the literature review of the semantic service annotation

2.2.1 Semantic Service Annotation for Web Services

There exists many publications about Web service annotation (Bo & Zhiyuan 2013; Canturk & Senkul 2011; Duo, Juan-Zi & Bin 2005; Lerman, Plangprasopchok & Knoblock 2006; Liu & Shao 2010; Meyer & Weske 2006; Mokarizadeh, Kungas & Matskin 2011; Patil et al. 2004; Soon Ae & Warner 2008; Stavropoulos, Vrakas & Vlahavas 2013; Wang et al. 2010). All of them focus on semi-automated approaches for annotating services. That is, given services, the mentioned approaches suggest annotations to the service providers. Then, the providers have to select the suitable annotations by themselves. In this section, the web service annotation is further categorized into non-ontology based and ontology-based annotation.

2.2.1.1 Non-Ontology based service annotation

Stavropoulos et al. (2013) develop a graphic tool, Iridescent, for manually creating and editing the web service descriptions. To annotate web services, the tool uses a string-based approach, Common Words, to match active ontology concepts to elements of the services, and then suggests them to web service creators.

Likewise Stavropoulos et al. (2013), Meyer and Weske (2006) also present a term-based technique for annotating web services. It aims to propose a light-weight community-based approach for semantic service annotation. Instead of focusing on service preconditions and effects, this approach asks a user to categorize services with keywords. Inspired by folksonomy annotation technique, the users can tag documents and then share them in a system. In this work, the authors present a service folksonomy, a hypergraph whose vertices are a set of actors A, a set of tags T and a set of available services called service landscape *S*. Edges of the service folksonomy link an actor (service user) "a" who categorizes a service "s" with tag "t". The strength of this proposed approach is that it is simple and there is no need to use service preconditions and effects. However, the service annotations may be redundant because of freedom tagging.

2.2.1.2 Ontology-based service annotation

In contrast, the other publications are ontology-based annotation approaches (Bo & Zhiyuan 2013; Canturk & Senkul 2011; Duo, Juan-Zi & Bin 2005; Lerman, Plangprasopchok & Knoblock 2006; Liu & Shao 2010; Mokarizadeh, Kungas & Matskin 2011; Patil et al. 2004; Soon Ae & Warner 2008; Wang et al. 2010).

Patil et al. (2004) propose MWSAF (METEOR-S Web Service Annotation Framework) which is a semi-automatic annotation framework for Web services. Given a Web service document (WSDL file), the system suggests a list of matched ontologies to a user for adding semantic information to that Web service. The main idea of this proposed semantic annotation method is to map a Web service, which is in a form of XML schema, to ontology, which is in a form of DAML, RDF-S or OWL. Firstly, the authors convert both Web service and ontology to SchemaGraph. Secondly, they attempt to calculate Match Score (MS) to capture the similarity between Web service and ontology. The MS is divided into two parts, namely ElemMatch and SchemaMatch. The former presents the linguistic similarity between names of service and ontology while the latter reflects the structural similarity between two concepts. Thirdly, the authors find the best mapping ontologies by considering the MS and the number of sub-concepts of the two concepts. Then, the system suggests those ontologies to a user. In this case, a user can accept or reject the proposed mapping ontologies.

Inspired by METEOR-S, Duo et al. (2005) propose a Web service annotation approach by using OWL ontology. Same as the METEOR-S, the proposed approach applies both terminology similarity and structural similarity to discover relevant ontologies. Main process of this approach is divided into three steps; 1) WSDL which is in the format of XML schema is directly translated into a temporary OWL-based ontology, 2) The ontology in the previous step is mapped to existing ontologies, and 3) The mapping result is applied for creating the semantics of the web service in the form of OWL-S. A difference between this approach and METEOR-S is that the proposed approach does not need to convert the services to a new standard like SchemaGraph, but it focuses on OWL-S description which normally is used in semantic web services.

In addition, there exist some publications in which the authors apply the idea of mapping web services to service ontology (Bo & Zhiyuan 2013; Canturk & Senkul 2011; Hui et al. 2010; Jiang & Luo 2013; Wang et al. 2010). Then, they propose different ontology alignment methods for matching the converted service ontology to relevant domain ontology.

Wang et al. (2010) focus on how to generate the service network. They propose a services classification method in order to allocate an input service into one or more than one related domain ontologies. Then, they attempt to annotate services by using the retrieved domains. In the process of service network construction, the input services are put into the actual layer, while their annotations are put into the abstract layer. Regarding the process of service annotation, concept matching between parameter concepts in input services and ontology concepts in selected domains is the main process of service annotation. In order to calculate MatchDegree (the degree of concept matching), a text similar degree, and type similar degree are defined. While the former applies natural language processing techniques such as WordNet and bipartite graphs to find the similarity between parameter name and ontology concept name, the latter calculates the similarity between parameter type and ontology property type. Then, the ontology concept with the highest MatchDegree value is selected to be the service annotation.

Jiang & Luo (2013) present a semi-automatic semantic annotation framework for web services and propose a novel Web service matching algorithm which includes ontology in Web service description. For annotating web service documents, the framework converts a Web service description in a form of WSDL document to OWL-S file by using WSDL2OWL-S tool. The XML vocabularies in WSDL depend on the domain ontology which is prepared previously. Then, annotated service documents (OWL-S files) are registered into service repository. For discovering a service, an ontology-based semantic matching algorithm is proposed. In this paper, the authors focus on matching the input parameters and output parameters between a registered service and a service request. The main idea of this method is to find the similarity between concepts in registered and requested services which are defined in the OWL-S form. The authors divide the concept similarity into 2 sub-similarities; namely, semantic similarity distance and semantic overlap. The former calculates the semantic distance by using a shortest linked path between two concepts while the latter calculates the semantic overlap by considering the number of common parents that the two concepts have. Then, the framework combines the concept similarity of input and output parameters in order to discover the best-matched service. In this case, the output parameters are more significant than input parameters.

Canturk and Senkul (2011) propose a lexicon-based alignment technique for semantically annotating web services. The approach aims to find the relatedness between the web services and the ontology. The main idea of the approach is to generate an ontology of a web service and apply an ontology alignment technique to map the service ontology to the domain-specific ontology. To process the ontology alignment, not only matching keywords of the service and the ontology are used, but different senses of words are also considered. With using WordNet, synonyms of service keywords are included in the service ontology by using level-sense synsets. To compare ontological keywords, the researchers calculate their similarity by considering levels of the keywords in the sense comparison tree.

Apart from the ontology alignment methods, some researchers address the web service annotation problem as a service classification problem (Bruno et al. 2005; Heß, Johnston & Kushmerick 2004; Lerman, Plangprasopchok & Knoblock 2006; Oldham et al. 2005).

Oldham et al. (2005) Inspired by MWSAF; the Meteor-S web service annotation framework (Patil et al. 2004), this paper attempts to improve the performance of

MWSAF by using Naive Bayesian classification. Both original MWSAF and the proposed approach in this paper aim to annotate web service descriptions (WSDL) to the relevant domain ontology. However, the researchers state that a disadvantage of the original MWSAF is that it is time-consuming, because it needs to compute a relevance score for every node in the service ontology and domain ontology. To solve this problem, the researchers apply the Naive Bayesian technique to classify WSDL descriptions into a related domain. Word frequencies of method names and argument names in WSDL are used for training the classifier. Given a WSDL, a domain ontology with the highest probability from the trained classifier will be returned. The experiments demonstrate that the proposed approach performs faster and better than the original MWSAF.

Lerman et al. (2006) attempt to automatically annotate the input and output of a web service to semantic types. Task of this paper is divided into two main sub-tasks; including 1) input data type recognition and 2) output data type recognition. The researchers propose a classification method, called metadata-based classification method, which exploits terms in WSDL of a service for identifying the input data type. In contrast, a content-based classification method, which invokes the service and uses the output content, is proposed for identifying the output data type. To classify the input and output types into the domain ontology model, the Logistic Regression algorithm is applied. The experiments show that both input and output classifiers perform correctly. However, this approach focuses on only the input parameter from a single domain. The relatedness between input parameters in several domains also needs to be considered.

Heß et al. (2004) introduce ASSAM which is a semi-automated tool for semantically annotating Web services. This tool suggests ontological concepts that are relevant to each WSDL element and enables a user to generate the OWL-S based semantic service metadata. ASSAM comprises two main tasks; namely WSDL annotation and data aggregation tasks. To annotate the WSDLs, the researchers propose an iterative relational classification algorithm. An advantage of the proposed approach is to combine results from several classifiers; such as intrinsic and extrinsic classifiers. This leads to the improvement of the performance of Web service annotation and it is able to deal with the incomplete feature situation. To combine data from Web services, a schema mapping algorithm called OATS is proposed. It focuses on matching instance data by using several distance metrics.

Bruno et al. (2005) aim to annotate web services and generate relationships among annotated services. Given a WSDL document, textual service description is extracted and represented by tf-idf metric. Then, the approach uses a trained Support Vector Machine (SVM) to classify the services into a specific domain. This annotation process can be invoked when a service provider publishes a service or a user queries services. Then, Formal Concept Analysis (FCA) is applied for creating a concept lattice which consists of key concepts of the services and their relationships. Those concept relationships are able to present hidden semantics among services. Moreover, this technique also assists in concept lattice refinement. That is, new key concepts can be added into the lattice when the service providers register new web services.

In contrast, Liu and Shao (2010) propose a framework for web service annotation and discovery. The researchers point out the semantic service annotation is an important key that assists in the semantic service discovery. Regarding the service annotation, the main idea of this paper is to annotate every parameter of the web service to a concept in the domain-specific ontology. The researchers apply SAWS algorithm for calculating the similarity value between every service parameter and every ontological concept. This paper uses a binary sequence to represent a service and a user request. Regarding the service discovery, the matching score between the request and the service is computed by using the intersection operation between two binary sequences. The service with the maximum matching score will be discovered. A major advantage of this proposed framework is to separately store the service annotation into the external database because of the purpose of reusability.

Furthermore, the heuristic-based technique is also applied in web service annotation (Mokarizadeh, Kungas & Matskin 2011). The proposed method is divided into 2 sub-methods, namely an ontology learning method and a semi-automatic semantic annotation method. Given a corpus of Web services in the form of WSDL and XSD documents, reference ontology of that service is generated by the former method. Then, given a Web service, the latter method is used to generate a suitable service annotation from generated ontology. In this paper, the authors apply Bag-of-Words model into an ontology learning method in order to extract a set of terms from a corpus of Web service documents. Then, they use a set of matching rules to retrieve the relationships between terms. Moreover, a synset which is a set of terms that have equivalent meanings is created from the instances of reference ontology. Regarding annotating a service, its element which is matched with a term in the synset is annotated by the entity reference of that synset.

2.2.2 Semantic Service Annotation for Other Online Services

Although most published papers are focused on web service annotation, some researchers have proposed annotation methods for the other services (Soon Ae & Warner 2008; Vehviläinen, Hyvönen & Alm 2006). Similar to the web service annotation, the literature of annotating other online services are divided into non-ontology based and ontology-based service annotation.

2.2.2.1 Non-Ontology based service annotation

Vehviläinen et al. (2006) apply semantic technologies for developing a tool for library help desk services, which is able to automatically answer questions from the users. The task in this paper is divided into two main sub-tasks; question-answer (QA) annotation and answer matching. The researchers propose a semi-automated method for annotating QA pairs with ontological concepts. That is, the method suggests concepts that are relevant to QA pairs to a librarian. Then, they manually select proper concepts and annotate them. In this paper, concepts are extracted from the input question by using natural language processing techniques. Moreover, this paper uses tf-idf and semantic cluster methods to rank the relevant concepts. To retrieve answers, the CBR-based method is applied.

2.2.2.2 Ontology-based service annotation

Apart from help desk service, Deep Web resources are annotated in order to enhance the efficiency of Deep Web search engine (Soon Ae & Warner 2008).
This paper proposes an automated semantic based approach for annotating Deep Web services (DWS), which are dynamic data accessed from Web and Web

services. The researchers state that the DWS annotations are able to present the content of the data source and this assists in effective DWS searching. To annotate DWSs, Deep Web data sources are sampled and clustered into different coverage dimensions. Then, the semantic projection function is applied for mapping the input semantic instances to the output semantic instances. The approach uses an ontology to convert the mapping function into RDF relationships format which is stored as the DWS annotation. Furthermore, the approach focuses on not only the service information like WSDL in Web services but also considers the DWS content extent which consists of the information about the frequency distribution of the content.

2.3 Semantic Service Crawling

Apart from the semantics of the services, this study also focuses on service crawlers which are software agents that are able to browse the Web, find the services, and index them for the purpose of service retrieval. Because of the specific features of service advertising information, namely ubiquity and heterogeneity, the retrieved service information is still not accurately relevant to service consumers' desires (Dong, Hussain & Chang 2012). For this reason, semantically crawling services is another essential issue that may improve the performance of service advertising discovery.

Focused crawling is another research topic which has been widely studied for a decade. A focused crawler is a software agent that crawls Web pages based on specific topics. This leads them to overcome the issue of time-consuming associated with non-focused crawling. In this case, a focused crawler is required to download Web information relating to service advertising. There exist many publications about focused crawlers, which can be divided into two categories; non-ontology based focused crawlers and ontology-based focused crawlers. An overview of the literature review of the semantic service annotation is presented in Figure 2.2.

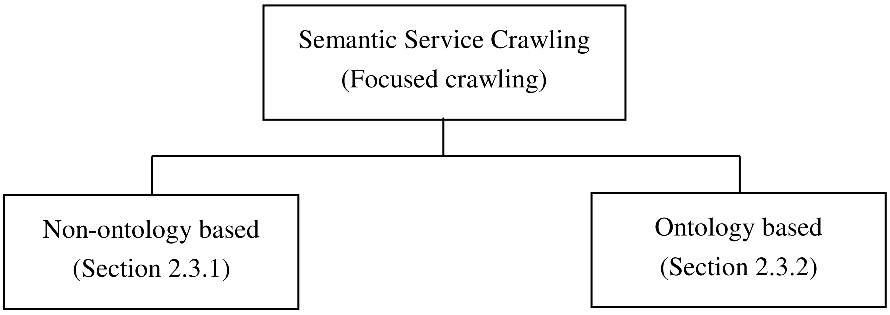


Figure 2.2 Overview of the literature review of the semantic service crawling

2.3.1 Non-ontology based focused crawling

Many researchers have proposed non-ontology based methods in order to develop web crawlers and information retrieval models. Some of them focused on text-based approaches. Ji et al. (2009) proposed a vertical search engine for online barter information named ExSearch. The researchers developed a focused crawler in order to collect the barter information from free-text Web pages by using text mining and information extraction based techniques. Moreover, Osuna-Ontiveros et al. (2011) applied a natural language processing based technique, to model web pages. The main idea of this paper is to find the topics of each web page. The researchers used CBC algorithm to cluster web pages and generate topics. Then, web pages are represented by a vector of topics. From the experiments, the results of this method outperformed those of traditional search engine. However, it took much time in the clustering process. Similar to this approach, Du et al. (2013) attempted to extract concepts or topics from web pages, index web pages by using topics, and then their system will retrieve web pages based on crawling topics. However, Du et al. applied formal concept analysis to extract concepts and create a concept context graph (CCG), which was used for ranking web pages. Furthermore, the performance of proposed graph depended on word units. That is, if key terms of web pages were close to the crawling concepts, the result would be better.

In contrast, some researchers applied link-based techniques in order to crawl and retrieve online information. Shchekotykhin et al. (2010) proposed a novel focused crawling method called xCrawl. The main purpose of this method was to improve recall value. The crawler would find the good hub pages that link to a lot of relevant documents. As a result, it was able to quickly locate the desired documents. However, the limitation of this approach was that the semantics of the document were not concerned.

Apart from text-based and link-based approaches, Almpanidis and Kotropoulos (2005) and Almpanidis et al. (2005) applied a hybrid approach; a combination of content-based and link-based techniques, to classify web pages into topics. Then, it retrieved web documents based on crawling topics, which was similar to some previously mentioned approaches.

Likewise, Xia et al. (2007) attempted to extract features which were able to properly represent chemical documents. The scholars applied various techniques, such as Latent Semantic Indexing (LSI), Mutual Information (MI), Naive Bayes (NB), and Support Vector Machine (SVM), to calculate the content relevance score. From the experiment, the combined LSI and SVM approach provided the best performance. Similar to Xia et al. (2007), Bernard et al. (2011) used LSI to extract textual features from the content of the Internet and find Online Financial Transaction (OFT) service. Moreover, they applied the logistic regression to decide whether the web page was OFT service. From the experiment, it showed that its results were better than those of manual searching.

2.3.2 Ontology-based focused crawling

Similar to non-ontology based focused crawling, ontology-based focused crawling approaches are categorized into main groups; text-based, structure-based, hybrid, and other specific approaches. Regarding text-based approaches, Dong et al. (2008, 2011) presented an ontology-based focused crawler for transport services. In this paper, the domain of knowledge was represented by service ontology, which consisted of service concepts. The aim of this paper is to browse the transport service metadata and index them by using service concepts. They applied an extended case-based reasoning (ECBR) algorithm to calculate the service-concept similarity. If the similarity is high, the service metadata is logically linked to a concept. In addition, the researchers also provided the measures, such as precision, recall, harvest rate, and fallout rate, for evaluating the crawling performance. The experiments showed that the proposed crawler performed well in all indicators when a threshold value was set to 0.8.

In contrast, Zheng et al. (2008) applied an artificial neuron network (ANN) for crawling web pages based on domain ontology. The aim of this paper was to find the optimal concept weights that maintain harvest rate during the crawling process. The researchers applied ANN to learn whether a document is related to the topic. They assumed that a topic contained relevant concepts in the domain ontology. While the input of ANN is term frequencies of concepts which relate to the topic, the output is the relevance score between a web page and the topic. The experiment showed that the performance of crawling method depended on how good the ontology was.

Like Zheng et al. (2008), Huang et al. (2009) calculated the relevance measure between the content of web pages and the topics which were defined in the form of ontology. The more the value of relevance score is, the more the content is closely related to the crawling topics. Unfortunately, the proposed method used only word frequency for relevance calculation. That is, the issue of the ambiguity of words was not concerned. In contrast, Fang et al. (2007) considered both semantic-based and link-based techniques in order to crawl Deep Web sources. Similar to previous work, relevance measure played an important role for mapping the content of Deep Web sources to existing ontology. Their experimental results presented that the proposed approach outperformed the keyword-based crawling method.

In addition, there have been some researchers that applied the fuzzy-based technique for ontology-based crawling. Jalilian and Khotanlou (2011) proposed a new method to calculate the similarity between topic and web document. Given a topic, related concepts in the domain ontology are weighted and retrieved by considering the distance between concepts and a topic and topic term frequency of web pages. By comparing with breadth-first and best-first search techniques, the proposed approach gave better results. In contrast, Su et al. (2005) focused on how to dynamically adapt the domain ontology. They applied the reinforcement learning to evolve the ontology by adjusting the weights of concepts. In order to compute the document-concept relevance, the proposed method focused on two factors, namely the frequency of class in a document and the weight of class showing the relevancy between class c and topic t. From the experiment, this approach outperformed typical word based learning or ontology-based approach.

Some hybrid approaches for ontology-based focused crawling have been published for a few years. Dong et al. (2012) proposed an ontology-learning based focused crawler for service advertising information. The aim of this work was to crawl the service advertising metadata based on the domain ontology. Similar to the other works, the researchers attempted to compute the relatedness between the service metadata and ontology concepts. The scholars applied text-based model (TCM) and probabilistic-based model (PCM) in order to match service metadata and concepts. After that, they used SVM to classify the relatedness of service metadata and concepts into two classes; relevant and non-relevant class. In addition, the same researchers have published papers that focused on the issue of ontology learning as well (Dong & Hussain 2013, 2014). While Dong and Hussain (2013) proposed a semi-supervised framework for ontology learning, Dong and Hussain (2014) proposed an unsupervised framework.

2.4 Semantic Query Expansion for Services

Apart from the semantic service annotation and service crawling, query expansion or query modification is an important technique for improving the performance of a semantic search engine. The usage of only terms in the query may not be enough to retrieve all relevant information. Consequently, query expansion methods are applied in order to increase the accuracy of information retrieval.

Mangold (2007) mentioned that query modification techniques are divided into 3 groups, namely manual, query rewriting, and graph-based techniques. Regarding the manual modification technique, users are able to add/remove related terms into/from the domain ontology. In contrast, regarding query rewriting techniques, user's query is automatically repaired by adding more query terms, removing non-related query terms, or replacing the query by related terms. For graph-based modification techniques, the documents are required to link to the ontology. Both concepts and documents are nodes of a graph. The results of this technique are not the modified queries. Given a query, the related nodes in the graph are retrieved. Then, the graph is traversed from those related nodes to find relevant documents.

Bhogal et al. (2007) also review the existing query expansion approaches which they categorize into two main groups; namely relevance feedback based approaches and knowledge-based approaches. The main idea in this review is to find the meaning of context of queries. From the review, the context is obtained from relevance feedback, term co-occurrence, and knowledge model. Regarding the relevance feedback-based approaches, the context is extracted from a document collection. Given an initial user query, a system returns documents which match the query. After that, the user has to select relevant documents from the previous result list. Relevant terms are extracted from the relevant documents and then added to the initial query. In contrast, regarding knowledge-based methods for query expansion, the context is extracted from term co-occurrence and ontology. In this case, the specific terms are considered from the relatedness to concepts in the ontology. Then, the terms, which are closely related to a concept of the user query, are selected to add to the initial query. In the review, use of term co-occurrence, long-span collocates, generative lexicon, and a probabilistic model is provided in order to calculate the relatedness.

Although much research work in the area of the query expansion for information retrieval has been published, the query expansion for online services has been discussed in some publications and still needs more research. This section presents existing literature in the area of the semantic query expansion for online services, which are divided into three categories; lexical-dictionary based, service knowledge-based, and hybrid-based query expansion approaches. The lexical-dictionary based approaches expand queries by using terms in a dictionary, while the service knowledge-based approaches expand queries by using terms in an existing service knowledge, for example, service information and service ontology. The hybrid-based approach expands queries by using both dictionary and service knowledge. An overview of the literature review of the semantic query expansion for services is presented in Figure 2.3.

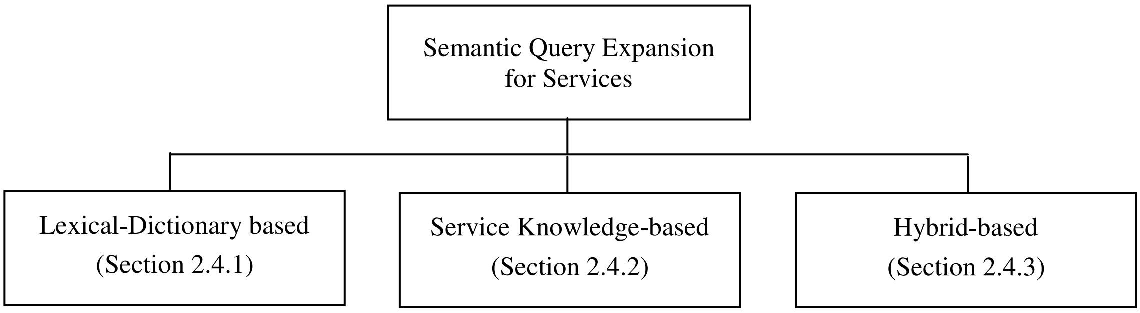


Figure 2.3 Overview of the literature review of the semantic query expansion for services

2.4.1 Lexical-Dictionary based query expansion

Mekhzoumi et al. (2011) propose a semantic-based approach for Web service discovery. This approach is divided into three sub-tasks; user query definition, query reformulation, and query matching. That is, a user gives a query based on provided query definition pattern. The query reformulation then semantically enlarges the query. Finally, the query matching calculates similarity value between the expanded query and each OWL-S web service. This paper proposes the query structure which consists of service name, input parameter, output parameter, and text description. The proposed approach applies WordNet to semantically enrich the query. The researchers introduce three expansion methods; 1) using morphological family, 2) using synonyms of the best-known word sense and 3) using synonyms of all word senses. To retrieve web services, the approach selects services based on similarities between the query and service profiles. This approach uses a syntactic method; Jaro-Winkler method, for service name matching and also applies a semantic method; Jiang-Conrath method, for calculating the similarities of input, output and text description parameters. The empirical results show that the approach with synonyms of the best-known word sense expansion gave the best performance.

Crasso et al. (2011) propose an approach for web service discovery, called WSQBE+. The main idea of this approach is to automatically find relevant service categories which assist publishers and searchers in web service publishing and web service discovery respectively. The proposed approach is able to suggest ranked categories that relate a new service to the service publisher; meanwhile, it discovers web services based on service categories. The WSQBE+ is developed from WSQBE, which creates queries from web service functionalities. A distinct difference from existing service discovery approaches is to retrieve web services by using service source code as a query. Service components are divided into implemented and non-implemented components. Implemented components act as inputs, while non-implemented components act as queries or examples. The output of the WSQBE+ is a web service that matches the query. To enhance the performance of the WSQBE, the WSQBE+ adds the text-mining based query expansion method into the approach. In this paper, the query is expanded by using attribute names, operation names and comments in the service source code. This content is split into terms and text-mining is then applied for removing stop-words and stemming. Expanded query terms are represented by using a vector and TF-IDF is applied for weighing query terms in the vector. Because of a large number of services, Rocchio's classification algorithm is used for reducing service space. Centroid, an average of services in each service category, is computed. To retrieve services, the query vector is matched to the centroid of each category by using cosine similarity.

2.4.2 Service Knowledge-based query expansion

Ohura et al. (2002) mine weblogs of the Japanese Yellow Pages, called iTOWNPAGE, and also propose a query expansion approach in order to improve the effectiveness of service category querying. This paper applies K-means clustering algorithm for grouping user access logs. Those clusters are applied for query expansion in the next step. Based on the empirical results, the researchers state that each cluster may contain categories from both sibling and non-sibling categories. That is, a user query may refer to several contexts and categories. Consequently, in this paper, the researchers divide the proposed query expansion into two problems; 1) Intra-Category Recommendation, and 2) Inter-Category Recommendation. Given a user request, Intra-Category Recommendation is able to recommend categories which are in the same category hierarchy of the user input. In contrast, Inter-Category Recommendation is able to suggest categories that relate to web access log, although they are non-sibling categories of the user input. The experiments show that this approach enables users to retrieve hidden relevant service categories.

Wang et al. (2009) propose efficient query expansion solutions for discovering service advertisements. The major advantage of this approach is to deal with the issues of time complexity and I/O cost. The approach consists of three major tasks; namely 1) advertisement indexing, 2) advertisement clustering and 3) advertisement retrieval. In this work, service advertisements are represented by using a set of keywords, called bid phrases. The relatedness between bid phrases and existing advertisements are represented by using a bipartite graph. Then, the researchers apply an agglomerative iterative clustering algorithm and use Jaccard similarity on that graph in order to cluster bid phrases. To index the service advertisements, the approach introduces a block-based index. That is, clustered bid phrases from the previous step are linked to non-fixed index blocks, and this assists in I/O transaction reduction. The size of each index block depends on the clustering results. Given a query, bid phrases in a relevant cluster are used to expand that query and retrieve more relevant advertisements. To retrieve relevant advertisements, a spreading activation algorithm is used to calculate the relevance scores between the user query and bid phrases and select top-k relevant advertisements. To evaluate the performance of the proposed approach, the researchers implement a system named AdSearch. The empirical results demonstrate that it performs well with less execution I/O.

Zhang et al. (2011) propose a context-sensitive querying method for retrieving web services. The researchers point out that typical web service search systems retrieve services by considering service content such as service document, service name, operation name, and parameters. However, the short length of the service content may decrease the performances of those retrieval approaches. Consequently, this paper focuses on an application-oriented query instead of a content-oriented query. Query terms in their work are from not only the service content but also the application scenario which is defined by BPMN. The service context which shows the relatedness between services and application requirement needs to be focused on. The researchers use a bipartite graph to represent the service context. Services and applications are classified into topics, and bipartite graphs of those services and application are generated under each related topic. This paper also proposes a query expansion method for the application-oriented queries. Service and application descriptions are extracted into terms. Unnecessary terms are filtered out. Then, based on the bipartite graph, the correlation degree between service terms and application terms are calculated by using probabilistic reasoning. Based on the topic, terms with high correlation degree are used for expanding the query.

2.4.3 Hybrid-based query expansion

Ma et al. (2013) propose a discovery approach for Web services, called the Lexical and Semantic Service Search (LS^3). The proposed approach is divided into two main tasks; 1) query expansion task and 2) service ranking part. Regarding the query expansion, the researchers expand a query by using both lexical-based and semantic based similarities. Regarding the lexical expansion, given a query, verbs and nouns are extracted from operation names and web service elements respectively. To expand those terms, WordNet, a lexical ontology, is applied for retrieving synonyms, hypernyms and hyponyms of those terms based on a defined threshold. Moreover, the approach also uses the domain-specific ontology to expand the query with the same process in the lexical expansion. Each term from the lexical method is mapped to the domain ontology to find relevant concepts. Then, the query is expanded by using terms in concepts that relate to the concepts in the previous step. That is, those conceptual terms act as synonyms, hypernyms, and hyponyms in the domain ontology.

It should be noted that Mekhzoumi et al. (2011), Wang et al. (2009) and Crasso et al. (2011) focus on both the semantic query expansion and semantic service retrieval. That is, they also focused on the comparative analysis of semantic service retrieval.

2.5 Semantic Service Retrieval

Traditional search engines use word occurrences to retrieve documents that are relevant to users' desire. Although they are widely used for this process, they still have limitations. Because of focusing on only frequency of words in documents, they may not retrieve other relevant documents that contain different words. Many researchers have broadly studied about retrieving documents by focusing on their semantics in order to improve the performance of the traditional search engines.

Semantic search approaches enable the user to retrieve semantically related documents. The focus of this thesis is on semantic retrieval approaches for business service advertising, not text documents on the Web. That is, the thesis attempts to develop a specific search engine for service information. As mentioned in section 2.1 and 2.2, the semantic service annotation and semantic service crawling are discussed for adding semantics to the services. These are the main components of a semantic service retrieval system.

In this section, the literature are divided into two categories; non-fuzzy based and fuzzy-based semantic service retrieval approaches, which are described in Section 2.5.1 and Section 2.5.2 respectively. An overview of the literature review of the semantic service retrieval is presented in Figure 2.4.

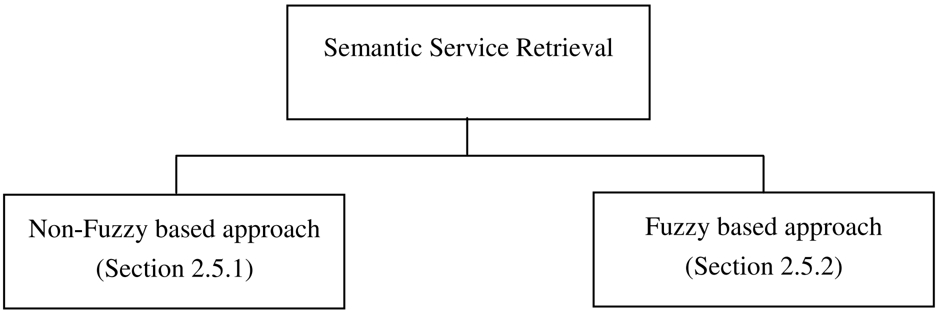


Figure 2.4 Overview of the literature review of the semantic service retrieval

2.5.1 Non-fuzzy based service retrieval

In this section, the focus is on the literature about semantic search approaches. There are a number of academic publications about semantic search approaches. Mangold (2007) surveys the semantic search approaches for retrieving text documents. He presents 7 criteria; architecture, coupling, transparency, user context, query modification, ontology structure and ontology technology, to discuss 22 semantic search systems. All search engines in this paper are ontology-based search approaches. Some of them apply user context, the users' desired information, to modify user query. Furthermore, Dong et al. (2008a) also carry out a survey about semantic search technology. Besides the semantic search engines which are mentioned in the previous survey, they also focus on various semantic search methods; graph-based, distributed hash table (DHT)-based, description logic (DL)-

based, and DAML+OIL-based approaches. In conclusion, regarding the survey, they state that there is an issue about persons' subjective perceptions. Dong et al. (2008a) conclude that we cannot design a conceptual structure that perfectly matches with users' desires. Another interesting issue is a lack of validation process. That is because several semantic search methods just focus on the conceptual model with little or no focus on validation of these methods (Bhagwat & Polyzotis 2005; Dichev & Dicheva 2006; Jin, Ning & Chen 2006; Patel et al. 2003).

In our research, the focus is on the semantic methods for service retrieval. Some scholars have studied this area. Toch et al. (2007) proposed a semantic approach for searching Web services, called OPOSSUM (Object-PrOcess-SemanticS Unified Matching). There are three main points which are provided in this approach, namely (a) defining a data model for querying Web services, (b) developing query matching techniques to improve the efficiency of recall, and (c) developing indexing techniques to reduce the processing time. Firstly, the authors defined a semantic operation and a service network. While an operation comprises input parameters, output parameters, concept labeling function and the certainty value of concept labeling, a service network is a graph of ontology-based operations which are linked by dependencies with a certainty value. Secondly, given a query, keywords of the query are mapped to concepts that have a high matching score. Then, a formal query is generated from those related concepts. After that, the matching certainty is calculated according to the correspondence between concepts of a transformed query and a service. Then, operations which are ranked by the matching certainty value are retrieved. Moreover, this approach also supports complex queries including conjunctive and disjunctive operators. Thirdly, concept index and service network index are proposed. While the former is a hash-based index that maps concepts to operations, the latter is a graph-based index that clusters the operations. This leads the number of edges between operations to be decreased. Consequently, the search space and the response time are reduced.

Dong et al. (2011) develop a semantic search engine for business service advertisements. Given a query, the engine retrieves relevant services based on the specific service ontology. Services are represented by service metadata which mainly comprised of the service provider, service name, and service description; meanwhile concepts of the service ontology consist of concept name and concept descriptions. An actual concept, a concept at the bottom layer, can link to many service metadata if the service metadata relates to that concept. The main idea of the retrieval process is to find concepts that are relevant to a query and then retrieve services that are linked to those concepts. The extended case-based reasoning (ECBR) approach is proposed to calculate the relevance score between the query and descriptions of the concept. The relevance score is high if the number of matching terms between the query and concept descriptions is high.

Akkiraju et al. (2006) propose a new semantic-based approach for web service discovery and web service composition. The semantic matching method is used to discover web services, while the AI planning algorithm is used to compose the discovered services. Web services are firstly annotated to related ontological concepts by using WSDL-S. Then, the services are indexed by using related terms and concepts. The main contribution of this work is to apply both domain-independent ontology and domain-specific ontology to match the relatedness between concepts and obtain the related terms for service indexing. Using the domain-independent ontology helps the proposed method to find synonyms of the service descriptions, while using the domain-specific ontology can find terms that relate to the domain ontology. Those terms from both sources (domain-independent and domain-specific ontology) are used for calculating the similarity score between concepts. Then, the winner-takes-all scheme is used for finding an appropriate index. To discover services, a backward searching algorithm is used for discovering relevant web services. Finally, the discovered services, the service indexes, and a metric planner are used for composing the services.

Iqbal et al. (2008) propose a semantic-based approach for web service discovery. The main contribution of this work is to use SAWSDL, SPARQL, and ebXML in order to make the service discovery system more flexible. Moreover, this work attempts to retrieve services based on four semantic factors; namely, functional semantics, data semantics, non-functional semantics, and behavioral semantics. Firstly, the proposed approach annotates services, whose descriptions are in the

WSDL format, by using SAWSDL (Semantic annotation for WSDL and XML schema). However, the SAWSDL can be applied for only input and output of a given WSDL. In this work, the researchers also use OWL-S and SPARQL to annotate pre-conditions and post-conditions of web services. In addition, this work defines user goals and service results by using SPARQL. To discover web services, the SPARQL-formatted goal and SPARQL-formatted service results are matched.

Paliwal et al. (2007) propose a new approach for semantically discovering web services. The main idea of this work is to use a statistical method like the Latent semantic indexing (LSI) with the ontology-based semantic mapping. Given a service request (a service query), the approach firstly expand the query by using ontological terms. The approach attempts to match keywords in the given request to relevant ontological concepts. Then, terms in those relevant concepts are used for service request expansion. A vector which is used to represent the service request is created based on the expanded request. LSI considers term associations and is used to reduce the dimensions of the representing vector. To discover web services, the cosine similarity between request vector and web service vector is calculated and used.

2.5.2 Fuzzy-based service retrieval

In addition, there have been some publications about fuzzy based approaches for semantic service retrieval (de Castilho & Gurevych 2011; Madkour et al. 2012; Zhai, Cao & Chen 2008; Zhao et al. 2012). The overviews of those approaches are described as follows.

Zhai et al. (2008) present a fuzzy-based semantic retrieval for traffic accident information. They also present a fuzzy ontology framework which consists of 3 components, namely, concepts, concept properties, and property values. They focus on the relationships, such as order relation, equivalence relation, inclusion relations, and complement relation between fuzzy concepts to semantically expand the queries. Unfortunately, this paper presents the conceptual model and does not validate the proposed model. Furthermore, the researchers do not mention how to extract the traffic accident information from web pages.

Zhao et al. (2012) propose a Bloom filter based approach for fuzzy matching Web services and users' requests. Given a request, both request and service are represented by Bloom filter vector. Given a service, the researchers assume that a service contains a set of *n* atomic services, which is represented in the form of vector. The values of each element in the vector are defined by using a hash function. In this paper, the hashing value will be 1, if an atomic service belongs to service set *S*. Then, the researchers apply B-Coverage to calculate the similarity between a vector of the request and one of the services. A service will be returned if its similarity is higher than the coverage limitation. The experiment shows that this method is accurate over 95%.

Madkour et al. (2012) present a novel fuzzy-based approach for Web service discovery and selection. The researchers focus on dealing with context information, which is fuzzy and uncertain. The context information in this paper is divided into 2 categories, namely functional context and non-functional context, which are represented by fuzzy domain ontology and graph respectively. Firstly, the functional context is used to discover services by inferring from semantic rules. After that, the discovered services are ranked by considering syntactic similarity, linguistic similarity, structural similarity, and semantic QoS similarity. Unfortunately, like Zhai, Cao & Chen (2008), this paper provides only a conceptual model and skips the experimention part.

De Castilho and Gurevych (2011) propose a fuzzy-based model for semantically retrieving Web services, which is called the Knowledge-Based Fuzzy Set Model (KB-FSM). The main idea of this paper is to combine bag-of-words represented by VSM and semantic relations in domain ontology in order to calculate the similarity between a natural language query and Web service description. The experiment shows that the result of the proposed model is better than the other knowledge-based approaches.

2.6 Critical Evaluation of Existing Approaches: An Integrative View

This section presents the critical evaluation of existing semantic-based approaches that are provided in the previous sections. We separate the complete service discovery task into four sub-tasks, including service annotation, service crawling, query expansion for services, and service retrieval. The literature evaluation for each sub-task for service discovery is individually discussed in Section 2.6.1 to Section 2.6.4

2.6.1 Semantic Service Annotation

The semantic service annotation is an important part of the service discovery. It aims to convey the meanings of services to the retrieval system. This helps the system to understand the service purposes and also discover the services more accurately. Based on the literature in Section 2.2 , the existing semantic-based approaches for service annotation are summarized and compared in Table 2.1.

Table 2.1. Comparative analysis of semantic annotation methods

Existing research work	Content type	Annotation model	Soft-computing approach
Patil et al. (2004)	Web services	Semi-automated	No (Ontology alignment)
Zhang et al. (2005)	Web services	Semi-automated	No (Ontology alignment)
Oldham et al. (2005)	Web services	Semi-automated	Yes / Machine learning (Naive Bayes)
Lerman et al. (2006)	Web services	Semi-automated	Yes / Machine learning (Naive Bayes)
Meyer and Weske (2006)	Web services	Semi-automated	No (Term based approach)
Vehviläinen et al. (2006)	Library help desk services	Semi-automated	No (Prefix trie)
Soon Ae and Warner (2008)	Deep Web resources	Semi-automated	No (Ontology alignment)
Hui et al. (2010)	Web services	Semi-automated	No (Ontology alignment)
Liu & Shao (2010)	Web services	Semi-automated	No (Logical reasoning)

Existing research work	Content type	Annotation model	Soft-computing approach
Canturk and Senkul (2011)	Web services	Semi-automated	No (Ontology alignment)
Heß et al. (2004)	Web services	Semi-automated	Yes / Machine learning (Iterative relational classification)
Bruno et al. (2005)	Web services	Automated	Yes / Machine learning (Support Vector Machine)
Mokarizadeh et al. (2011)	Web services	Semi-automated	No (Heuristic-based technique)
Bo and Zhiyuan (2013)	Web services	Semi-automated	No (Ontology alignment)
Stavropoulos et al. (2013)	Web services	Semi-automated	No (Term based approach)

Shortcomings of the existing literature in semantic service annotation

Based on a thorough review of existing literature, in the area of semantic service annotation, the issues are identified as follows:

1. The current research work lacks developing approaches for service annotation of non-Web services. Although several semantic service annotation methods have been proposed for a decade, most researchers have focused on only the annotations for Web services. Unfortunately, a lot of service information, which is not in the form of WSDL, has been provided on the Web and is currently available (Dong, Hussain & Chang 2009). Hence, there is a need for semantic annotation approaches for online service information are needed in order to improve the efficiency of service crawling and service retrieval.

2. The current research work lacks developing approaches for automated service annotation. Most of the existing work on semantic service annotation, with the exception of Bruno et al. (2005), is the semi-automated approach, which still needs the service providers to select proper annotations. Although the selected annotations can reflect the semantics of services from the perspectives of the providers, the semi-automated approaches are not suited for annotating the very large amount of online service information which has been increasingly provided and changed.

3. Regarding the issue of ambiguity and heterogeneity of online services, the soft computing techniques such as Fuzzy Logic (FL) or Neural Network (NN) are well suited to the complex annotation process. Unfortunately, from the literature review, those techniques have not been applied into the semantic annotation process.

To address the above mentioned shortcomings, an automated soft-computing based method is developed to solve the semantic annotation problem for business service information.

2.6.2 Semantic Service Crawling

Apart from the semantic service annotation, the semantic service crawling is another important part of the service discovery. The research work in this area aims to semantically gather service information on the Web and collect and index it in the service knowledge base. Semantic crawling methods assist the service retrieval systems to collect more suitable services and this leads the system to more precisely retrieve services. Based on the literature in Section 2.3 , the existing semantic-based approaches for service crawling are summarized and compared in Table 2.2.

Table 2.2. Comparative analysis of semantic crawling methods

Existing research work	Content type	Crawling model	Soft-computing approach
Ji et al. (2009)	Online barter information	Non-ontology based	No (Text-based approach)
Osuna-Ontiveros et al. (2011)	Web pages	Non-ontology based	No (Text-based approach)
Du et al. (2013)	Web pages	Non-ontology based	No (Text-based approach)
Shchekotykhin et al. (2010)	Web pages	Non-ontology based	No (Link-based approach)
Almpanidis et al. (2005)	Web pages	Non-ontology based	No (Hybrid approach)
Xia et al. (2007)	Chemical documents	Non-ontology based	No (Textual feature extraction)
Bernard et al. (2011)	Online Financial Transaction (OFT) service	Non-ontology based	No (Textual feature extraction)
Dong et al. (2008, 2011)	Transport services	Ontology-based	No (Text-based approach)

Existing research work	Content type	Crawling model	Soft-computing approach
Zheng et al. (2008)	Web pages	Ontology-based	Yes (Artificial Neural Network)
Wei et al. (2009)	Web pages	Ontology-based	No (Text-based approach)
Fang et al. (2007)	Deep web sources	Ontology-based	No (Hybrid approach)
Jalilian and Khotanlou (2011)	Web pages	Ontology-based	Yes (Fuzzy based approach)
Su et al. (2005)	Web pages	Ontology-based	No (Text-based approach)
Dong et al. (2012)	Service advertising	Ontology-based	No (Hybrid approach)
Dong and Hussain (2013)	Service advertising	Ontology-based	No (Hybrid approach)
Dong and Hussain (2014)	Service advertising	Ontology-based	No (Hybrid approach)

Shortcomings of the existing literature in semantic crawling

Based on a thorough review of existing literature, in the area of semantic crawling, the issues are identified as follows:

1. Use of Neural Network (NN) for semantic crawling has been proven to achieve better results (Zheng, Kang & Kim 2008). Additionally, the use of Fuzzy based method has improved the precision value for semantic crawling (Jalilian & Khotanlou 2011). However, those approaches are developed for crawling Web pages, not service information. Moreover, the use of either other soft computing methods such as Genetic Algorithm (GA) for semantic crawling, or the combination of soft computing methods such as Neuro-Fuzzy System (NFS) has not been investigated.

2. Most of the existing work ignores issues of heterogeneity and ambiguity of service information. Because of various service providers, service information is provided by using different perspectives and patterns. Furthermore, the service information is typically described in natural language. Consequently, the content of services is ambiguous. Fortunately, the use of soft computing methods is able to solve real

world problems with approximate solutions (Zadeh 1994). Regarding the semantic crawling process, its results may be precise and useful if the soft computing methods are applied in the process. Although some research work (Dong & Hussain 2013, 2014; Jalilian & Khotanlou 2011) has focused on either heterogeneity or ambiguity, no research work seriously focuses on both issues.

In order to address the above mentioned shortcomings in this thesis, we use the focused crawling approach which is proposed by Dong et al. (2012) to crawl transport services provided on Yellow Pages.

2.6.3 Semantic Query Expansion for Services

Semantic query expansion is another important research area that assists in the intelligent service retrieval system development. The research work in this area aims to reformulate or enlarge a query in order to increase more relevant services, even though those services do not contain the query terms. Based on the literature in Section 2.4 , the existing semantic-based approaches for query expansion are summarized and compared in Table 2.3.

Table 2.3 Comparative analysis of semantic query expansion methods for online services

Existing papers	Content type	Query expansion method
Ohura et al. (2002)	Weblogs	K-means clustering
Mekhzoumi et al. (2011)	Web services	WordNet-based method
Wang et al. (2009)	Service advertisements	Agglomerative iterative clustering
Zhang et al. (2011)	Web services	Probabilistic reasoning
Crasso et al. (2011)	Web services	Text mining
Ma et al. (2013)	Web services	WordNet-based and Ontology-based

Shortcomings of the existing literature in semantic query expansion

Based on a thorough review of existing literature, in the area of semantic query expansion, the issues are identified as follows.

1. The current research work lacks developing approaches for semantic query expansion of business services. Most existing work focuses on expanding the queries for web service discovery.

2. The current research work lacks developing approaches for effective query expansion of business services. Although some researchers have proposed the query expansion methods for business service domain, only service knowledge-based information is used for enlarging the query. However, lexical-based terms, like synonyms, also should be applied to improve the performance of query expansion.

In order to address the above mentioned shortcomings in our work, hybrid based methods, which apply both lexical-based and service knowledge-based information, are developed to solve the query expansion problem for business service information.

2.6.4 Semantic Service Retrieval

Semantic service retrieval semantically retrieves services that are relevant to a user query. Based on the literature in Section 2.5 , the existing semantic-based approaches for service retrieval are summarized and compared in Table 2.4.

Table 2.4. Comparative analysis of semantic retrieval methods

Existing papers	Content type	Query expansion	Soft-computing based approach
Toch et al. (2007)	Web services	Yes (Ontology-based approach - Specific)	No (Graph based approach)
Dong et al. (2011)	Service advertisements	No	No (Text based approach)
De Castilho and Gurevych (2011)	Web services	No	No (Structure-based approach)
Madkour et al. (2012)	Web services	No	Yes (Fuzzy based approach)
Zhai et al. (2008)	Traffic accident information	Yes (Ontology-based approach - Specific)	Yes (Fuzzy based approach)
Zhao et al. (2012)	Web services	No	No (Bloom filter method)

Existing papers	Content type	Query expansion	Soft-computing based approach
Mekhzoumi et al. (2011)	Web services	Yes (WordNet-based method)	No (Jaro-Winkler & Jiang-contrath method)
Wang et al. (2009)	Service advertisements	Yes (Agglomerative iterative clustering)	No (Spreading activation algorithm)
Crasso et al. (2011)	Web services	Yes (Text mining)	No (Cosine similarity)
Akkiraju et al. (2006)	Web services	No	No
Iqbal et al. (2008)	Web services	No	No (SPARQL matching)
Paliwal et al. (2007)	Web services	Yes (Ontology-based approach - Specific)	No (Cosine similarity)

Shortcomings of the existing literature in semantic querying

Based on a thorough review of existing literature, in the area of semantic service retrieval, the issues are identified as follows:

1. The current research work lacks developing approaches for service retrieval of business services. Although many semantic service retrieval methods have been proposed for a decade, most publications have focused on only web service retrieval. However, semantic retrieval approaches for business services or service advertisements are also currently required in order to support the business collaborations in the Digital Ecosystems.

2. The current research work lacks developing soft-computing based approaches for business service retrieval. Although some researchers have proposed the fuzzy-based retrieval approaches, they focus on web service and traffic information domain. Furthermore, one of the proposed fuzzy-based approaches uses only ontological terms to expand the user queries, while another approach does not consider query expansion. Based on the issue of the ambiguity of queries, effective query expansion methods are still required to increase the performance of service retrieval.

In order to address the above mentioned shortcomings in this thesis, soft-computing based methods are developed to solve the semantic retrieval problem for business service information.

2.7 Conclusion

This chapter presents the review of the existing academic literature about semantically retrieving online services. Based on typical service retrieval systems, the literature is separated into four main parts; semantic service annotation, semantic service crawling, semantic service query expansion and semantic service retrieval. The literature in the area of the semantic service annotation present how to give the meanings or purposes of online services, while the approaches for semantic service crawling explain how to semantically collect and keep online services in the system knowledge base. To retrieve the services, the approaches for semantic service querying and semantic service retrieval are required. The literature about service querying focuses on how to properly define and reform user queries, which assist in retrieval performance improvement. This thesis aims to semantically retrieve online business services; however, most of the existing literature focuses on semantic-based approaches for web services. Although some publications mention semantics in online business services, few semantic-based approaches are studied and the performance of the business service retrieval still needs to be improved. Therefore, the first major shortcoming of the existing literature is a lack of semantic-based approaches for retrieving business services. The second major shortcoming is that soft-computing based approaches, which are able to deal with complex tasks, have not been applied into processes of business service retrieval.

Chapter 3 Problem Definition

3.1 Introduction

In the first chapter, the importance of the semantic service annotation, semantic service querying and semantic service retrieval was presented. Then, the existing work in those research areas was reviewed in Chapter 2. It stated that many researchers proposed various techniques for semantically annotating and retrieving online services. Most of them focused on web services, but only Dong et al proposed the term-matching method called ECBR in order to semantically retrieve business services which are focused on in this research (Dong, Hussain & Chang 2011). To propose a methodology for semantic service retrieval, however, applying the soft-computing based approaches has not been investigated yet, and there have existed other limitations that need to be solved. In this chapter, the problem of the semantic service retrieval system is defined. Terminologies which relate to the problem definition are formally described in Section 3.2 , and then details of defining the problem are presented in Section 3.3 . In this thesis, the research problem is divided into five research issues which are described in Section 3.4 . The research questions and the research objectives are developed in Section 3.5 and Section 3.6 respectively. In Section 3.7 , the overview solution for solving the defined problem is proposed. Finally, the conclusion of this chapter is presented in Section 3.8 .

3.2 Key Concepts

This section presents a collection of terminologies and their formal definitions that are referred to in various sections in this thesis.

3.2.1 Semantics

Semantics is defined as the study of meaning by considering the relationship between words, phrases and sentences (Kearns 2000).

In this thesis, the focus is on annotating, querying and retrieving services based on semantic-based methodology.

3.2.2 Service

Service is defined as an online business service advertisement which is provided in service portal websites such as Yellow pages website. The purpose of a service is to convey the essential information, such as a service name, a service description, a service provider name and a provider's contact details, to the target customers.

In this thesis, the methodology for semantically annotating and retrieving only business services, not web services, is proposed. As a result, the terms 'service' and 'business service' are used interchangeably.

3.2.3 Service Metadata

Service Metadata is defined as the information of a business service that conveys more information about the service. The service metadata in this thesis consists of service provider name, service provider address, service provider contact details and service description.

In this thesis, service metadata is used for representing services. Given a user query, the service retrieval system returns relevant services in the term service metadata.

3.2.4 Service Provider

Service Provider is defined as a commercial agent who provides the information about a service to their customers.

This thesis focuses on only the service providers who propose the services via the service directory websites.

3.2.5 Service Ontology

Service Ontology is defined as a conceptual structure that represents the knowledge of services in the form of the relationships among service entities or service concepts.

The service ontology that is used in this thesis is the transport service ontology which consists of transport service concepts, such as air transport, rail transport and road transport concepts, and their relationships.

3.2.6 Service Concepts

Service Concepts is defined as a set of the general ideas of services. Each concept may relate to other concepts with the various relationships.

In this thesis, the service concept comprises the service concept name and concept descriptions.

3.2.7 Service Knowledge Base

Service Knowledge Base is defined as a system that represents knowledge of services. It consists of the service ontology, a set of service metadata, and a set of relationships between ontological service concepts and service metadata.

In this thesis, the service annotation methodology is proposed to generate the service concepts-service metadata relationships.

3.2.8 Service Annotation

Service Annotation is defined as a methodology that calculates the relevance score between every service metadata and every service concept, and then links service metadata to its relevant service concepts.

3.2.9 Service Querying

Service Querying is defined as a methodology that calculates the relevance score between a given user query and every service concept, and then returns a set of service concepts that is relevant to the query.

3.2.10 Service Retrieval

Service Retrieval is defined as a methodology that calculates the relevance score between a given user query and every service metadata, and then returns a set of service metadata that is relevant to the query.

3.2.11 Relevance Value

Relevance Value is defined as the value that presents the similarity or the relatedness between two entities.

In this thesis, there exist three types of relevance values; the annotation, the querying, and the retrieval relevance value.

3.2.12 Annotation Relevance Value

Annotation Relevance Value is defined as the relevance value between a service metadata and an ontological service concept.

In this thesis, given a service metadata, the annotation relevance value is applied to select the relevant service concepts and then annotate them to the service.

3.2.13 Querying Relevance Value

Querying Relevance Value is defined as the relevance value between a user query and an ontological service concept.

In this thesis, given a user query, the querying relevance value is applied to select the relevant service concepts and then return them to the retrieval system.

3.2.14 Retrieval Relevance Value

Retrieval Relevance Value is defined as the relevance value between a user query and a service metadata.

In this thesis, given a user query, the retrieval relevance value is applied to select the relevant service metadata and then return them to the user.

3.2.15 Relevant Services

Relevant Services is defined as the collection of service metadata whose relevance values are greater than a defined threshold.

3.2.16 Relevant Concept

Relevant Concept is defined as the collection of service concepts whose relevance values are greater than a defined threshold.

3.2.17 Soft Computing

Soft Computing is defined as a collection of the computational techniques that deal with imprecision, uncertainty, partial truth and approximation. The major Soft Computing techniques are Fuzzy Logic, Neural Computing, Evolutionary Computation, Machine Learning and Probabilistic Reasoning (Jin 2010).

This thesis applies Soft Computing techniques such as Neural Computing, Machine Learning and Fuzzy Logic to semantically retrieve business services.

3.3 Problem Overview and Problem Definition

In this thesis, the focus is on the service retrieval methodology for online business services. The service retrieval methodology assists to create a profitable connection between the service providers and the service consumers (or the customers). That is, the providers are able to productively advertise their services and the customers comfortably retrieve and access the business information. Because the number of the services provided on the Internet and the customers who use the online connection to reach the service information have been rapidly increasing, an efficient methodology for retrieving services is still desirable.

Semantic technologies have been applied to improve the performance of the service retrieval methodology because they enable a machine to discover the services by considering the meaning of words or texts that appear in the service documents. In this thesis, semantic service annotation approaches are applied in order to automatically understand the purpose of the services, and this assists the machine to precisely retrieve the services that are relevant to the user aspirations. Based on the literature review in the area of the semantic service annotation in Section 2.2 , it is seen that various researchers proposed the semantic based approaches for annotating the web services; the electronic services that are able to connect to each other on the Web. In contrast, this thesis focuses on annotating and retrieving the online business services. The differences between the business services and the web services are formally presented with three diverse perspectives, namely 1) service objective, 2) service format and 3) service size. The online business services are the commercial service information provided by the service owner, while the web services are pieces of programs that support the communication between human and machine, and machine and machine. Thus, the objective of the business services is to convey the service information from the service providers to the service consumers; on the other hand, the web services are applied for automatically calling web-based functions in order to accomplish defined tasks. The business services are described by using the natural language or the human language. It contains the essential information of the service such as the provider information and the details of the service. In contrast, the web services are represented by using the Web Services Description Language (WSDL) in order to enable the machine to understand the purpose and how to utilize that service. The size of both the business services and the web services are quite small. The web services provide small scripts to describe what the input, the function name and the output is, while the business service information is presented by using a collection of short phrases and sentences. Based on the above different characteristics between the business services and the web services, the service retrieval methodology for both service types needs to be tailored to each service type; to be fit for purpose. As mentioned previously, the business services are described in human text format. That is, the content of services probably contains some ambiguous words, phrases and sentences, which enable the machine to misunderstand the actual purposes of those services. As a result, dealing with this uncertain problem is a major challenge of this thesis. Soft computing provides the computational techniques that are utilized for dealing with imprecise, uncertain and approximate tasks. Unfortunately, based on the literature review in Section 2.2 , there exists no soft computing-based approach for semantically annotating the business services. In addition, the proposed approaches in the area of the semantic service annotation are either manual or semi-automated. That is, the approaches require the service providers to either completely annotate the services to defined service categories, or confirm whether the suggested service categories are correct. Consequently, those approaches are ineligible for dealing with the real world dynamic services.

Apart from applying the proper service annotations, querying by using the appropriate user queries is another important factor to improve the performance of the service retrieval system. The proper query enables the machine to correctly and quickly discover the relevant service information. Similar to the service annotation, the users input the queries in the form of unstructured text format. Words or phrases in the user query may not appear in the service information, although the meaning of the query is analogous to that service. On the other hand, the user query may not relate to the queried service, although that service information contains words in the given query. The uncertain and ambiguous issues are still addressed in the querying process and the semantic-based techniques are needed to solve those issues. To get better results from the querying process, several scholars have discussed how to prepare the proper query and various methods for expanding the given query have been proposed. In this thesis, the focus is on the semantic-based query expansion approach. The review of the service retrieval systems in Section 2.5 presented that some systems made use of the ontology in order to enlarge terms of the given query. The knowledge represented by using ontology returns terms that semantically relate to terms in the query. Those returned terms are added into the given query in order to increase the possibility of querying the relevant services. Based on the literatures, the ontology that the researchers applied may be either the general ontology; like WordNet, or the domain-specific ontology; however there exist no service retrieval systems that applies both general and domain-specific ontology. Expanding the query by using the general ontology assists the retrieval system to improve the performance of the querying process if the relevant services contain synonyms of the query terms. On the other hand, in the case of applying the domain-specific ontology, the querying performance of the system will be enhanced if the relevant services contain the technical terms that relate to the query. As a result, using only the general or the domain-specific ontology may not query all relevant services. Furthermore, similar to the service annotation issue, most of the proposed approaches in Section 2.5 retrieve only the web services which are not the focus of this thesis. Only the approach proposed by Dong et al. (Dong, Hussain & Chang 2011) was able to retrieve the business service advertisings, but they did not utilize any soft-computing based technique to enhance the system performance.

Based on the literature review and the previous above problem definition, the research issues, the research questions, and the research objectives are presented in Section 3.4 , Section 3.5 and Section 3.6 respectively.

3.4 Research Issues

Based on the literature review in Chapter 2, the limitations of existing literature are concluded as follows.

Research Issue 1:

Existing approaches for semantically annotating services have focused on only Web services, not online advertisements of business services. Moreover, they have not applied the soft computing techniques, which are able to better handle complex tasks, in the real world situations.

Research Issue 2:

Existing work on web service annotation proposes semi-automated approaches for service annotation which are unsuitable for the real world tasks that services are increasingly provided and updated.

Research Issue 3:

Existing work on semantic query expansion applies either a lexical database like WordNet or a domain-specific ontology to enlarge query terms. This is unsuitable for the real world tasks in which the users possibly use terms based on both lexical synonyms and related concepts.

Research Issue 4:

There is no existing work on fuzzy semantic based approach for retrieving business service information.

Research Issue 5:

Several semantic search methods just focus on the conceptual model. As a result, the validation process is not mentioned in many publications.

3.5 Research Questions

According to the research issues mentioned in Section 3.4 , the following research question is presented.

"How can semantic technologies coupled with soft computing approaches be used to achieve better service retrieval?"

This research question can be divided into 4 sub-questions as follows.

Research Question 1:

How can business service information be annotated automatically?

Research Question 2:

How can a query expansion process be used on business services to aid in better service retrieval?

Research Question 3:

How can business service information be retrieved effectively (with higher precision) ?

Research Question 4:

How can the developed semantic service annotation, semantic service querying, and semantic service retrieval techniques be validated on the real world dataset?

3.6 Research Objectives

In order to address the above research questions, the objectives of this thesis are defined as follows.

Research Objective 1:

To develop intelligent soft-computing based methods for automatic semantic service annotation.

Research Objective 2:

To develop semantic based methods for service query expansion.

Research Objective 3:

To develop intelligent soft-computing based methods for semantic service retrieval.

Research Objective 4:

To validate the above developed methods by building a prototype of service composition or service retrieval.

3.7 Research Approach to Problem Solving

Based on the above research objectives, this thesis focuses on developing and validating a semantic-based methodology for retrieving the business services. To solve the above research issues, it is necessary to utilize a systematic scientific approach in order to guarantee that the developed methodology is based on scientific theory. In this section, the existing scientific research methods are briefly presented and the method that has been selected to solve the problems in this thesis is described in detail.

3.7.1 Existing Research Methods

The research in information systems is divided into two primary groups; the science and engineering based approach and the social science approach. The science and engineering approach gains new knowledge by using experimental or measurable information (Peffers et al. 2007), while the social science approach generally uses survey and interview in order to gain new knowledge based on a systematic plan (Bryman 2015).

The science and engineering based research approach aims to discover different knowledge, different techniques, different methodologies or different devices for solving the defined problems. Gallier et al point out that three imperative hierarchical levels; conceptual level, perceptual level and practical level (Galliers 1992), are applied for properly developing something. The conceptual level, the first level, is concerned with creating and analyzing new ideas and concepts. The second level, the perceptual level, is applied for designing and implementing new methods and approaches. After that, the practical level is about testing and validating the proposed approaches by using the real world cases testing or laboratory testing.

The social science based research approach aims to investigate and explain the social phenomena. There exist two methods of the social science approach; namely

Quantitative method and Qualitative method. Social scientists may apply either the Quantitative or the Qualitative method. To investigate social claims, the Quantitative method works with measurable evidence and usually applies statistical models to analyze the raw data, while the Qualitative method focuses on subjective evidence and aims to explore deep knowledge by using personal observation and interview techniques. Unlike the science and engineering research approach, the social science research aims to understand the social evidence but it does not focus on creating a new thing.

This thesis focuses on developing a new methodology for semantically retrieving the business services. To accomplish this task, new ideas about the semantic service retrieval need to be defined and the proposed ideas are then implemented into an actual system. Thus, this thesis follows the science and engineering research approach in order to make the methodology more scientific. The detail of the chosen research method is presented in the next section.

3.7.2 Choice of Science and Engineering based Research Method

To develop a new semantic-based methodology for service retrieval, the science and engineering research method is selected and used. The overview of this method is displayed in Figure 3.1.

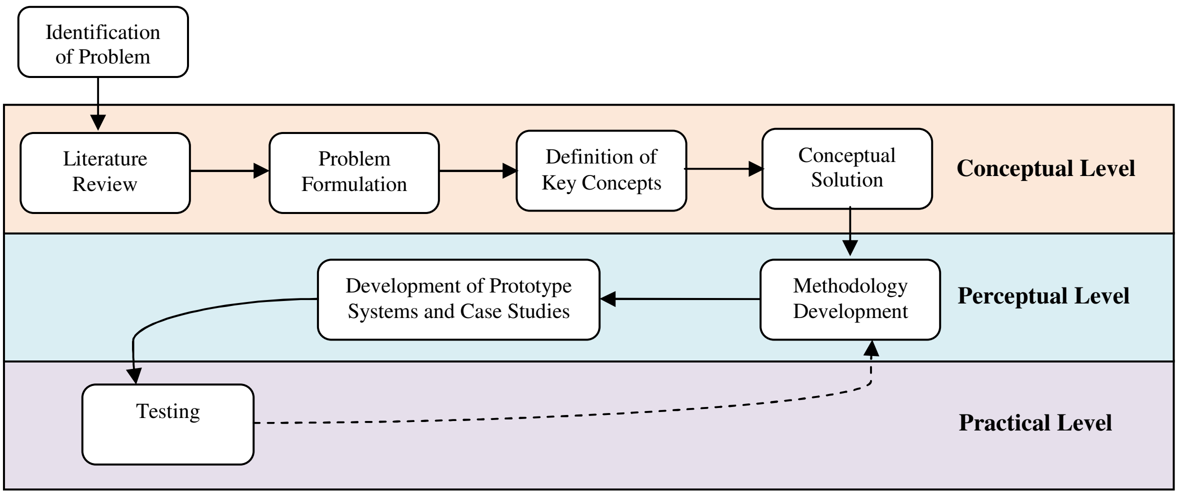


Figure 3.1 A science and engineering based research method

According to Gallier et al (Galliers 1992), three research levels; conceptual level, perceptual level and practical level, are defined to develop the proposed methodology. Firstly, the research problems in the area of semantic service retrieval were identified. Then, sub-processes in the conceptual level were applied to specify ideas and concepts of a new semantic-based service retrieval methodology. After that, sub-processes in the perceptual level were applied to develop and implement the proposed methodology and then tested it in the practical level.

Conceptual Level

The first level of the research method is about creating new ideas and concepts of a semantic-based service retrieval methodology. This research level consists of four processes; namely, literature review, problem formulation, definition of key concepts and conceptual solution.

Literature review

Based on the identified research problems, the literature in the area of the semantic service retrieval was reviewed in order to find out the research gaps or the research issues. Based on sub-problems that are focused on in this thesis, the literature was classified into three groups; namely, the semantic service annotation, the query expansion, and the semantic service retrieval.

Problem formulation

The research issues that were identified from reviewing the existing literatures were applied for formulating the research problem. In this process, research questions and research objectives were defined, as was presented in Section 3.4 , Section 3.5 and Section 3.6 respectively. Based on the literature review, the problem was divided into three sub-problems about service annotation, query expansion and service retrieval.

Definition of key concepts

After formulating problem, a set of key concepts; such as semantics, service, service metadata, service provider, service ontology, service concepts, service knowledge based, service annotation, service querying, service retrieval, relevance value, annotation relevance value, querying relevance value, retrieval relevance value, relevant service, relevant concept and soft computing, was defined in Section 3.2 . In the next process, those terms were used for defining terms in the conceptual solution.

Conceptual solution

This process is to conceptually generate solutions for the addressed research. Based on the problem formulation process, the solution was also partitioned into three main parts, which were described in Chapter 4. This thesis emphasized applying the soft computing techniques to solve the defined research problems.

Perceptual Level

This research level is concerned with the solution development and implementation. The perceptual level comprises the methodology development process and the development of prototype.

Methodology development

Based on the conceptual solution in the conceptual level, the methodology for semantic service retrieval was developed. The proposed methodology was subdivided into three parts; namely service annotation, service querying and service retrieval. The details of those methodologies are presented in Chapter 5, Chapter 6 and Chapter 7 respectively.

Development of prototype

After proposing the methodology, prototypes of the semantic service annotation, semantic service querying and semantic service retrieval were then developed. A case study was provided for the methodology validation. In this thesis, Java programming language was used to implement the methodology of the semantic service retrieval, and transport service domain was selected to be the case study. The detail of the prototype was demonstrated in Chapter 8.

Practical Level

The last research level is concerned with testing and validating the developed methodology. It contains the testing process. In this thesis, the proposed methodology was tested by using the transport service dataset. Moreover, the methodology was evaluated by using performance measures in the area of information retrieval; like precision, recall and f-measure, fallout rate, annotation rate, querying rate, retrieval rate, and all-measures combination. In the case when the experiments were unacceptable, the proposed methodology needed to be fine-tuned to improve its performance.

3.8 Conclusion

This chapter presents the problem definition of this research. In this thesis, the problems of the semantic service retrieval are categorized into three main sub-topics: namely, the semantic service annotation, the semantic service querying and the semantic service retrieval. Based on the literature review in those research areas, five research issues were identified. We subsequently proposed solution for each of the research issues, thereby resulting in research solution. The key concepts relating to this thesis were also described. The science and engineering based research method was selected to develop the semantic based methodology for retrieving business services. Furthermore, the reasons why this method was chosen and how each research process related to this thesis were explained.

Chapter 4 Solution Overview

4.1 Introduction

According to the literature review in Chapter 2 and the formal problem definition in Chapter 3, the semantic service retrieval issue is decomposed into three particular parts; the semantic service annotation, the semantic service querying and the semantic service retrieval. Consequently, five research issues and research questions are addressed. Based on the workflow of the science and engineering based research methodology, after defining the research problem, the conceptual solution for solving the issues of the semantic service retrieval is created.

The main content of this chapter presents an overview of the solutions for each of five service questions provided in Chapter 3. To assist the readers to understand the concepts of the whole content, the definitions relating to the semantic service retrieval are firstly described. The definitions of service, service ontology and service knowledge base are proposed in Section 4.2.1 , Section 4.2.2 and Section 4.2.3 respectively. An overview of the solution for the addressed research problem is presented in Section 4.3 . After that, the solution overviews of each research question are separately explained in Section 4.4 - Section 4.6 . Finally, the conclusion of this chapter is presented in Section 4.8 .

4.2 Definition of Service, Service Ontology and Service Knowledge Base

4.2.1 Service and Service Metadata

This thesis focuses on the methodology for semantically retrieving services. The term "service" in this thesis is defined as an electronic advertisement by the service provider for the purpose of business. The services that are focused on in this thesis are similar to the service advertisements in typical business directory websites; for example, Yellow Pages (YellowPages 2016), True Local (TrueLocal 2016) and Yelp (Yelp 2016). When users search for "transport service", some of business services in Yellow Pages, True Local and Yelp are retrieved as shown in Figure 4.1, Figure 4.2

and Figure 4.3 respectively. The Yellow Pages retrieves the service named Jayde Transport Services with its service details, while the True Local returns the services called "Extra Mile Movers" and "CTP Green Slip". On the other hand, the services like "Uber Sydney" and "Cooperate Personal Services" are retrieved by Yelp. After this, the term "service" and "business service" are used interchangeably in this thesis.

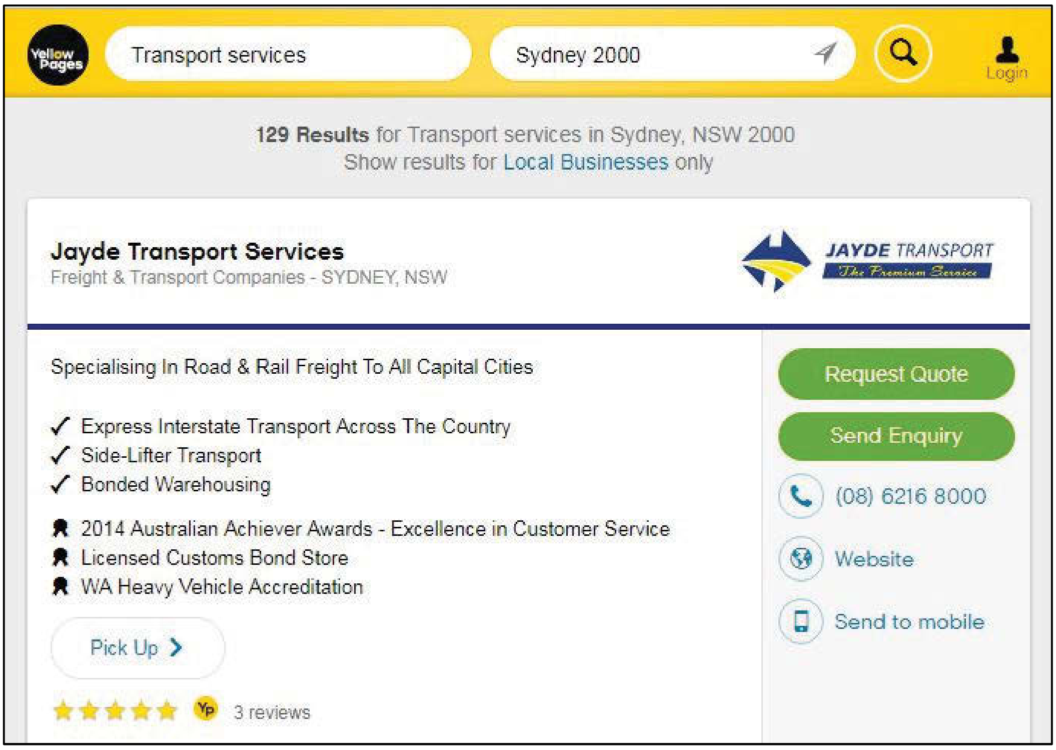


Figure 4.1 Example of the retrieved business service in Yellow Pages by using the query "transport services"

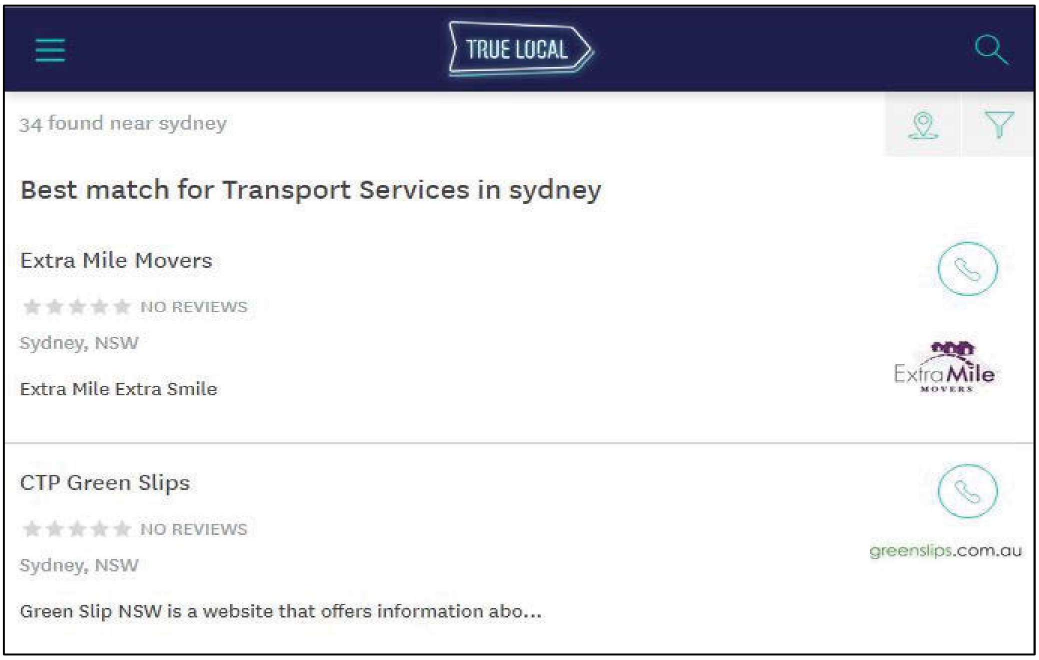


Figure 4.2 Example of the retrieved business service in True Local by using the query "transport services"

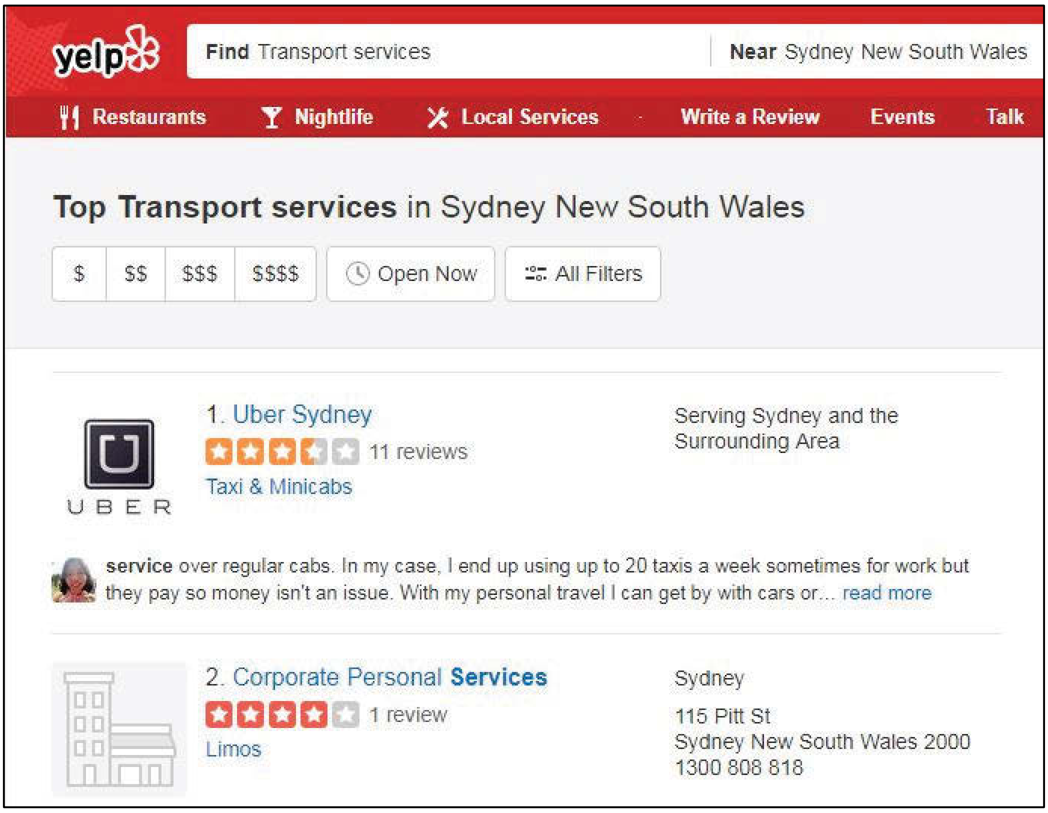


Figure 4.3 Example of the retrieved business service in Yelp by using the query "transport services"

In this thesis, the service metadata is used to represent the services. Based on the general business services, the content of the services contains essential information that is able to explain those services, for example, service name, service explanation and service contact details. It is obviously seen that a service consists of small pieces of text-based information.

To define the service metadata, the service structure defined by Dong et al (Dong, Hussain & Chang 2011) is applied in this thesis. Similar to the general services, the service metadata in this research consists of six properties; service name, service provider name, service provider address, service provider contact details, service description and relevant service concepts. The content of the first five properties of the service metadata is in the text format. For instance, the metadata of the service *"Jayde Transport Services"* in Figure 4.1 assigns the service name, the service provider name, the provider address, the provider contact details and the service description as *"Jayde Transport Services"*, *"Jayde Transport"*, *"Sydney NSW 2000"*, *"Tel:(08) 9454 5635"* and *"Specialising In Road & Rail Freight To All Capital Cities. Express Interstate Transport Across the Country. Side-lifter transport. Bonded Warehousing."* respectively. The last property, the relevant service concept, makes the service metadata different from the general business services. That is, this property contains a set of service concepts that are relevant to the service. In this case, the relevant service concepts of this service may be "Freight Container" and "Freight Container Storage" service concepts. To find out the relevant services, Dong et al (Dong, Hussain & Chang 2011) requires the service provider to define the service categories which directly relate to the service concepts, while we propose automated approaches which calculate the relatedness from the service description and the provider name. In this thesis, the service metadata is also called the service data entity metadata (SDE metadata). Thus, the term "service metadata" and "SDE metadata" are used interchangeably.

The service concepts represent ideas or concepts that are provided in the specific service domain, such as the transport service domain, health service domain and mining service domain. In this thesis, the service concepts are primary components of the service ontology which is applied for representing the knowledge of services.

The details of the service ontology and the service concepts are described in the next section.

4.2.2 Service Ontology

As previously mentioned, this thesis attempts to find out how to retrieve the business services semantically. The term "semantics" is defined in general as the study of meaning by considering the relationship between words, phrases and sentences (Wikipedia 2016). Apart from using the text-based relationships, the domain-specific ontology, called Service Ontology, is applied to define the semantics of the services and the queries.

In this thesis, the service ontology is used for representing the knowledge of services and the service relationships in a particular service domain. Each specific service domain composes several ideas which are related to each other. For example, the specific domain is the health service domain which may contain various concepts about health services, such as: anatomical pathology service, aromatherapy service, blood collection service, cardiology service and clinical service. Consequently, in this thesis, the service ontology consists of a set of service concepts and the relationships among those concepts. Based on the above example, anatomical pathology, aromatherapy, blood collection, cardiology and clinic are defined as the service concepts, and they have the “is-a” relationships with the health services.

Like the service metadata, the ontological structure that is proposed by Dong et al (2008b) is applied to define the service ontology in this thesis. The service ontology is a hierarchical structure and its structure is shown in Figure 4.4. The service concepts are able to be organized into different levels. The service concepts in the lower level are sub-domains of those in the upper level. The concept in level 0 refers to the main service domain, and the concepts in level 1 represent the sub-domain of level 0, and so on. That is, the service concepts in the upper level are more general than those in the lower level. On the other hand, the concepts in the lower level are more specific. The service concepts are categorized into two types; abstract service concept and actual service concept. The actual service concepts are concepts in the bottom level of the ontology, while the abstract concepts are in the other levels. The abstract concepts refer to the abstract domain and the sub-domain of service concepts. The actual concepts relate to real services and can be linked to service metadata.

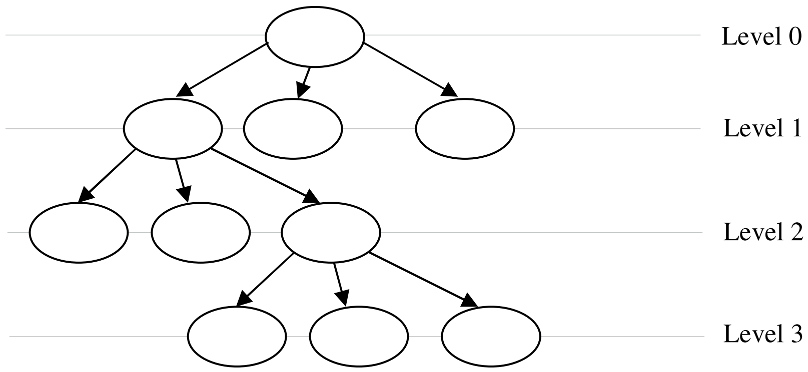


Figure 4.4 The structure of the service ontology

Similar to the service metadata, the service concept comprises three main properties; namely, service concept name, service concept descriptions and relevant services. As mentioned above, the service ontology is applied for giving the semantics of services in the focused domain. Thus, the last property, the relevant services, is applied for linking a service concept with services that their main ideas are close to that concept.

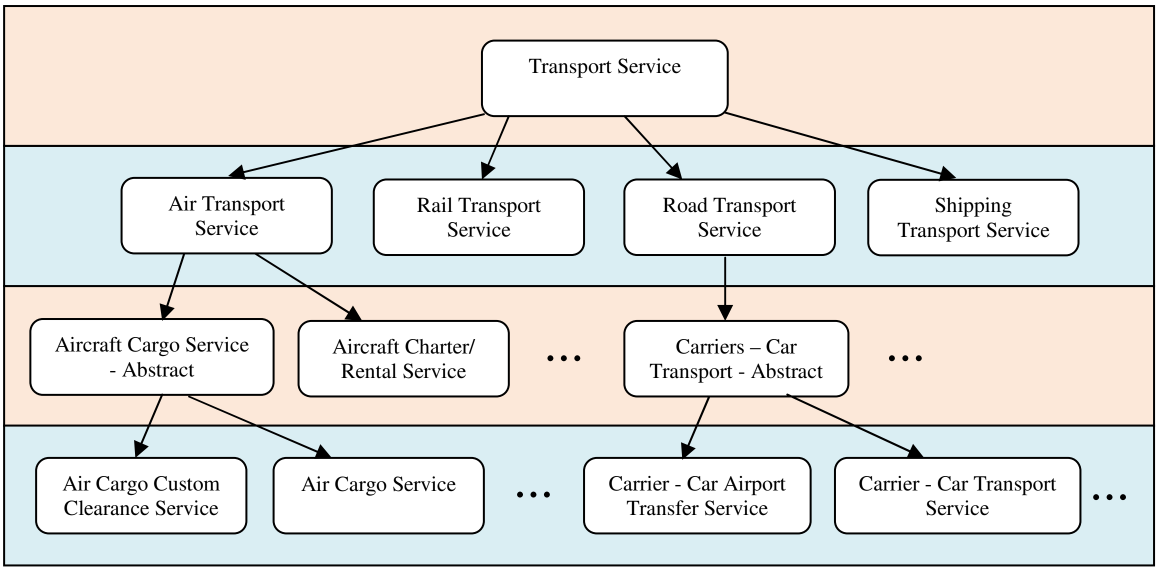


Figure 4.5 The transport service ontology

In this thesis, the focus is on the transport service domain and applies the ontology which is also provided by Dong et al. (2008b). They conducted a survey of transport service company websites in order to build a transport service ontology. The structure of the transport service ontology is shown in Figure 4.5. The transport services consists of four service sub-domains, including "Air Transport Service", "Rail Transport Service", "Road Transport Service" and "Shipping Transport Service". Each sub-domain comprises the abstract service concept; for example, the "Air Transport Service" sub-domain contains various abstract concepts such as "Air Cargo Service - Abstract" and "Aircraft Charter/Rental Service". The service concepts in the bottom layer are actual service concepts which are linked to relevant service metadata. As mentioned in Section 4.2.1 , the same proposed approaches are applied to calculate the relevance score between a service concept and a service metadata. A service concept is linked to the service metadata if the relevance score is greater than the threshold. The detail of this approach is described in Section 4.4 .

4.2.3 Service Knowledge Base

This section presents how to store the services and their semantics for the service retrieval methodology. The service knowledge base which is derived from Dong et al (2011) is used to store information about services.

The structure of the service knowledge base is shown in Figure 4.6. It is comprised of the domain-specific service ontology, like the transport service ontology, and a collection of the service metadata. Each service metadata can be linked to one or more service concepts, and each service concept can be linked to one or more services. For example, service metadata of the service 4 (SDE_4) is related to two service concepts. Moreover, there exists a service concept that is linked to both SDE_1 and SDE_4. The links represent the relevance between service metadata and service concepts. Thus, service concepts can act as ideas or semantics of services. That is, one service may have various meanings, while multiple services may mean to the same thing. To link a service to its concept, the link properties in the service metadata and the service concepts are set. For example, a service named "Airlines & Airline Agent Bookings" is annotated to a service concept named

"Airline Agent". Figure 4.7 shows the service metadata, the service concept and their relatedness.

In this thesis, the relatedness between service metadata and a service concept is called a "service annotation". The automated service annotation is an important issue that was addressed in Chapter 3. The details of the service annotation is explained in Section 4.4 .

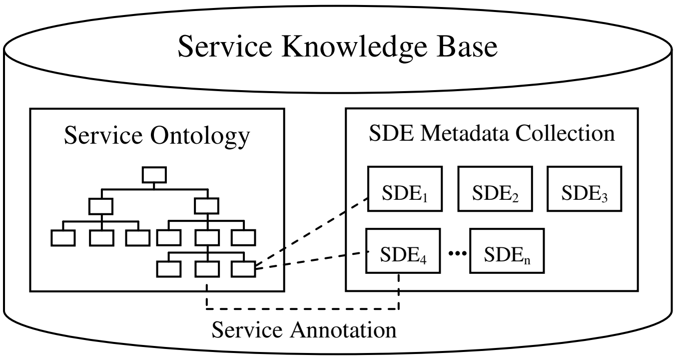


Figure 4.6 Service knowledge base

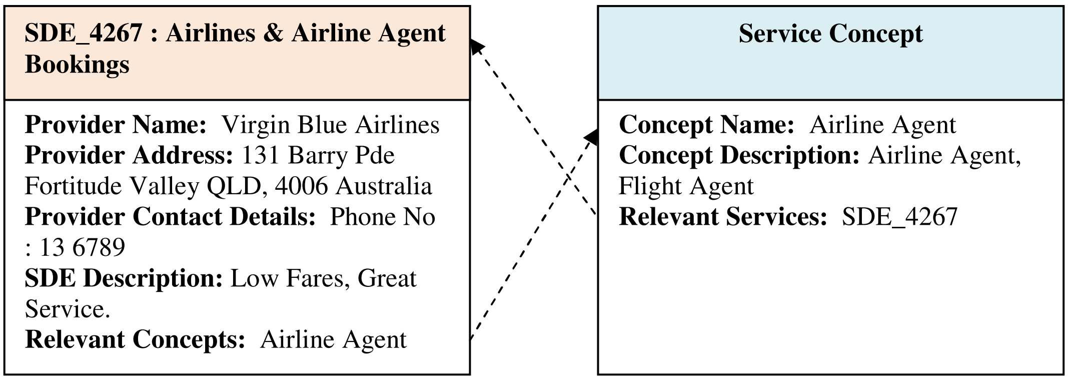


Figure 4.7 Service metadata, service concept and their relations

4.3 Overview of the Solution for Semantic Service Retrieval

As mentioned in Chapter 3, based on the literature review, the research question was defined as follows.

"How can semantic technologies coupled with soft computing approaches be used to achieve better service retrieval?"

To solve this question, both semantic-based and soft computing approaches are applied for developing the service retrieval methodology. Given a query, the proposed methodology enables a user to query services (SDEs) based on service concepts that are relevant to a query. For example, given query "Flight booking service", an SDE "Airlines & Airline Agents Bookings" provided by Air Niugini is retrieved.

The research solution for this proposed question is shown in Figure 4.8 and described as follows.

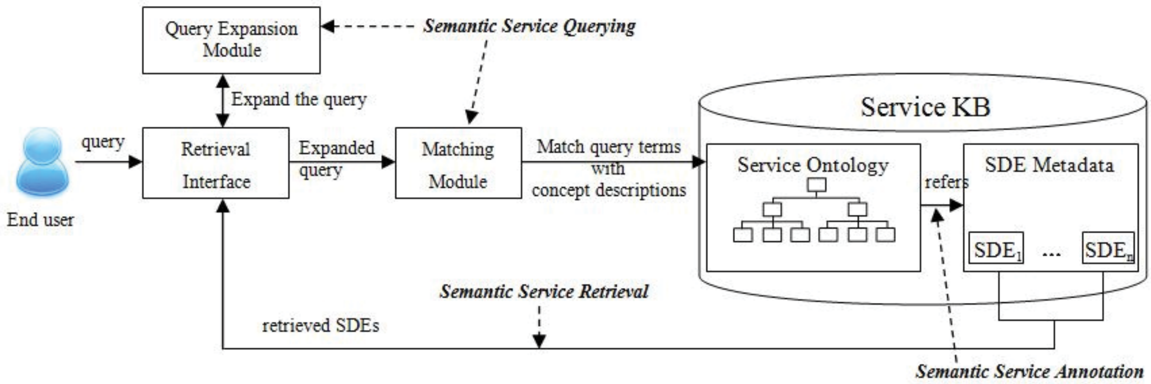


Figure 4.8 Workflow of the semantic service retrieval

Firstly, the proposed service retrieval methodology receives a query via the retrieval interface. It extracts a set of separating terms from the query. Then, the query terms are expanded based on their synonyms and related concepts. The retrieval interface sends the expanded query terms to the matching module. Next, the matching module computes the relevance score between the query terms and each service concept in the service ontology. Service concepts with the acceptable relevance score are selected as the relevant service concepts. Finally, the methodology then retrieves services that are annotated to the queried service concepts.

Based on the above proposed solution, an overview of the whole solution for this thesis is presented in Figure 4.9.

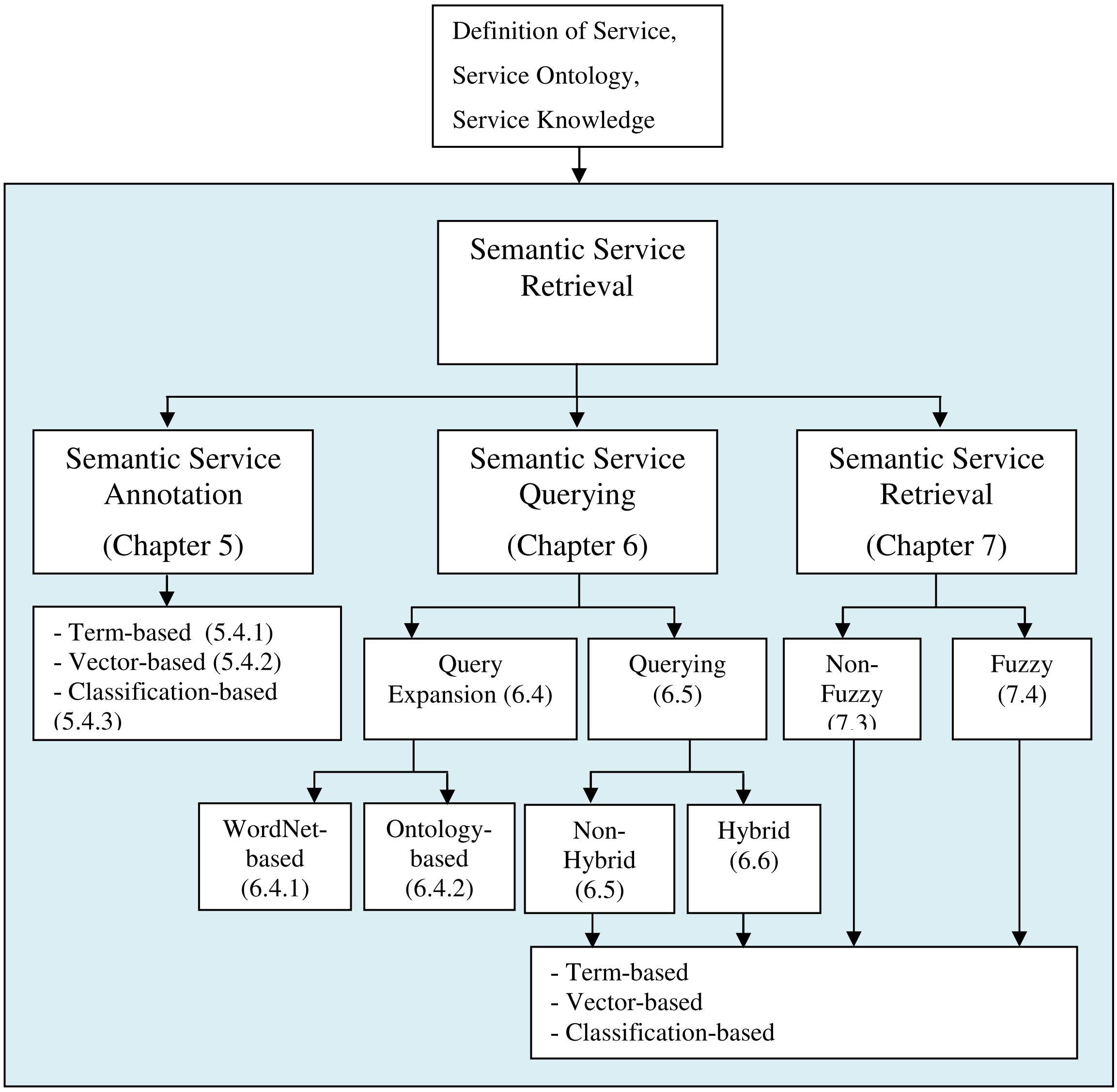


Figure 4.9 Overview of the whole solution for the semantic service retrieval

In this section, the overall solution for the semantic service retrieval is explained. Then, an overview of the solution for the four research questions is presented individually in Section 4.4 - Section 4.7 The proposed solution for semantic service retrieval is composed of three primary sub-solutions, namely, solution for semantic service annotation, solution for semantic service querying and solution for semantic service retrieval.

1) **Solution for semantic service annotation**

The semantic service annotation aims to semantically link services to their relevant service concepts in domain-specific service ontology. An overview of the solution for semantic service annotation is presented in Section 4.4 . The main idea is how to calculate the relatedness value between a service and a service concept. To solve this problem, term-based, vector-based and classification-based approaches are applied. Moreover, the proposed semantic service annotation methodology is also explained in Chapter 5.

2) Solution for semantic service querying

The semantic service querying aims to semantically discover ontological service concepts that are relevant to a given query. An overview of the solution for semantic service querying is presented in Section 4.5 . The solution for this issue is composed of the solution for query expansion and the solution for service querying. Furthermore, the proposed semantic service querying methodology is also explained in Chapter 6.

3) Solution for semantic service retrieval

Given a query, the semantic service retrieval aims to discover relevant services based on the service concepts from the semantic service querying. An overview of the solution for this problem is shown in Section 4.6 . A non-fuzzy based and a fuzzy based approach are proposed for retrieving services. The proposed semantic service retrieval methodology is presented in Chapter 7.

The semantic service retrieval methodology is able to retrieve the services by using the semantic service annotation and the semantic service querying methodologies. Thus, the service retrieval methodology is also divided into term-based, vector-based and classification-based approaches.

4.4 Overview of the Solution for Semantic Service Annotation

This section presents an overview of the solution for semantically annotating services. Based on the research issue 1 and the research issue 2 in Chapter 3, the research question was provided as follows:

"How can business service information be annotated automatically?"

Based on the literature reviews in Chapter 2, the existing service annotation approaches were developed only for web services and they were semi-automated. In this thesis, the semantic service annotation methodology is proposed for automatically finding out ideas of the services. The annotation methodology aims to index the services to their relevant service concepts. As mentioned in Section 4.2.3 , the service knowledge base comprises a collection of services and service ontology, and the links between the services and the service concepts are called the “service annotation”. It is noted that one service concept may be annotated to many services, meanwhile one service may be annotated to many service concepts.

An overview of the research solution for this proposed question is described as follows.

Step 1 : An environment for the semantic service annotation is set. The terms "service", "service metadata", "service ontology", "service concept" and "service knowledge base" are defined in Section 4.2.1 , Section 4.2.2 and Section 4.2.3 .

Step 2 : To annotate services to their relevant service concepts, the similarities between ontological concepts and SDE metadata are calculated. In this thesis, the proposed semantic-based annotation approaches are divided into three types; namely term-based, vector-based and classification-based approaches.

1) Term-based approach semantically annotates a service by matching terms in the service and each service concept. This approach calculates the service-service concept relevance score by using a semantic term-matching technique. In this thesis, the Extended Case-Based Reasoning (ECBR)-based annotation approach is proposed for service annotation. Details of the ECBR annotation approach is provided in Section 5.4.1 .
2) Vector-based annotation approach represents the content of a service metadata and a service concept by using vectors. The cosine similarity of those vectors is computed to reflect the relatedness between the service and the concept. In this thesis, the terms service and concept are represented by the Vector Space Model (VSM). Furthermore, Extended Vector Space Model (EVSM) is also proposed to include semantics into the existing VSM model. The Vector-based annotation approach is explained in Section 5.4.2

3) Classification-based annotation approach applies existing classification techniques in order to group the services. Each service group is referred to as an ontological service concept. This approach finds out relevant service concepts for a service by using classification methods. If a service is classified to a service concept, that service is related to that concept. In this thesis, the focus is on both Neural Network (NN) and Machine Learning (ML) to classify the services. Regarding the NN-based approach, Feed-Forward neural network (FF) and Radial Basis function Network (RBF) are applied. The ML-based approach applies K-Nearest Neighbor (KNN), Classification Tree (CT) and Support Vector Machine (SVM) algorithms. Detail of the Classification-based annotation approach is presented in Section 5.4.3 .

Step 3 : Degrees of relevance between concepts and services based on their similarity are calculated. In the simple case, the relevance score between a concept and a service is equal to their similarity which is calculated in step 2.

Step 4 : Services are linked to relevant ontological concepts based on their relevance scores. In this study, the ECBR and the Vector-based annotation approach link an SDE to a concept if their relevance score is greater than a defined Annotation Threshold *(AT)*. In contrast, the Classification-based approach links an SDE to a concept if the SDE is classified to the service concept.

In this thesis, the service annotation is divided into three types, namely single-label, multi-label and combine-label annotation. The single-label annotation enables a service to link to only one service concept, while the multi-label annotation allows the service to link to many service concepts. Regarding the combine-label annotation, the service normally links to many concepts the same as in the multi-label annotation; however, it is possible that the service cannot be annotated to any concept. In this case, the service will be linked to the most relevant concept.

To annotate the services, the proposed service annotation approaches, term-based, vector-based and classification-based approaches, are applied for all annotation types. There exists a difference in applying the classification-based approach. KNN, CT and SVM are used for multi-label annotation; meanwhile only KNN and CT are used for single-label and combine-label annotation. The different proposed semantic service annotation approaches are concluded and summarized in Figure 4.10.

Single-Label				Multi-Label				Combine-Label			
Term-based	Vector-based	Classification-based		Term-based	Vector-based	Classification-based		Term-based	Vector-based	Classification-based	
ECBR	VSM	NN	ML	ECBR	VSM	NN	ML	ECBR	VSM	NN	ML
	EVSM	FF	KNN		EVSM	FF	KNN		EVSM	FF	KNN
		RBN	CT			RBN	CT			RBN	CT
							SVM				

Figure 4.10 Semantic service annotation approaches

4.5 Overview of the Solution for Semantic Service Querying

This section presents an overview of the solution for semantic service querying. Based on the research issue 3 in Chapter 3, the research question was provided as follows.

"How can a query expansion process be used on business services to aid in better service retrieval?"

Given a query, the semantic service querying aims to discover service concepts that are relevant to the query. The main idea of the semantic service querying is similar to the one of the semantic service annotation in Section 4.4 . The service annotation attempts to get relevant concepts of a given service, while the service querying aims to get relevant concepts of a given query. Thus, the solutions of the service annotation and service querying are quite similar. However, a query expansion process is also added into the solution of service querying.

An overview of the research solution for this proposed question is described as follows:

Step 1 : A given query is firstly semantically expanded in order to increase the possibility of matching additional service concepts. The output of this step is called as "the expanded query". In this thesis, WordNet and a domain-specific ontology are applied to enlarge the query.

1) WordNet-based approach applies a large lexical database, like WordNet, for expanding the query by using synonyms of terms in that query. Details of the WordNet-based query expansion approach are described in Section 6.4.1 .
2) Ontology-based approach applies a domain-specific service ontology, like the transport service ontology, to get terms that conceptually relate to the query terms. Then, those terms are used to enlarge the query. Details of the Ontology-based query expansion approach are presented in Section 6.4.2 .

Step 2 : To query the service concepts, the similarities between the expanded query from step 1 and service concepts are computed. In this study, non-hybrid based and hybrid based service querying approaches are proposed.

1) Non-hybrid based approach calculates the similarity from the query that is expanded by using either WordNet-based or Ontology-based approach. As with the semantic annotation, three approaches, term-based, vector-based and classification-based approaches, are applied for getting the query-service concept similarity. A difference is that inputs of the service querying approaches are the expanded query and the content of service concept, while those of the service annotation approaches are the content of service and service concept. Details of the non-hybrid based service querying approach are presented in Section 6.5 .
2) Hybrid based approach computes the similarity from the query that is expanded by using both WordNet-based and Ontology-based approach. As with the non-hybrid based approach, the hybrid approach applies the same term-based, vector-based and classification-based service querying approaches to compute the similarity. Details of the hybrid based approach are presented in Section 6.6 .

The approaches proposed in the solution of the semantic service annotation are also applied for querying the service concepts. Furthermore, the semantic-based approach for expanding the query is proposed.

Step 3 : Similar to the semantic service annotation, the relevance score between a query and a concept is equal to the similarity from step 2.

Step 4 : Based on the relevance score from step 3, service concepts with the acceptable relevance value are selected to be the relevant service concepts. The ECBR and the Vector-based service querying approaches query a concept if the relevance score between a query and a concept is greater than a defined Query Threshold *(QT)*. In contrast, in the Classification-based approach, service concepts that the query is classified to are returned.

4.6 Overview of the Solution for Semantic Service Retrieval

In this section, an overview of the solution for semantic service retrieval is presented. Based on the research issue 4 in Chapter 3, the research question was provided as follows.

"How can business service information be retrieved effectively?"

Semantic service querying aims to retrieve services that are relevant to the given relevant service concepts from Section 4.5 . Based on the literature review in Chapter 2, although few researchers proposed the semantic method for retrieving business services, the approach applied only the term-matching method which is unsuitable with the real world situations. In this thesis, based on the proposed service annotation and service querying approaches, the semantic service retrieval methodology based on term-matching, vector model and classification techniques are proposed. Furthermore, fuzzy logic is also used in order to deal with ambiguous service content.

After receiving the service concepts that are relevant to the given query from the previous solution, the services that relate to those service concepts are retrieved. As mentioned previously, the relatedness between a service and a concept is represented by the service annotation. Consequently, the main idea of the semantic service retrieval is to discover services that are annotated to the queried concepts. In this thesis, non-fuzzy based and fuzzy based service retrieval approaches are proposed.

1) Non-fuzzy based approach retrieves all services that are annotated to the queried concepts. Based on the proposed annotation and service querying

approaches in Section 4.4 and Section 4.5 , the non-fuzzy based approach is divided into three main approaches; term-based, vector-based and classification-based approaches. Given a query *Q,* the term-based service retrieval approach gets the relevant service concepts and the service annotations from the term-based service querying approach and the term-based service annotation approach respectively. In the same way, the vector-based approach gets the results from the vector-based annotation and vector-based service querying; meanwhile the classification-based approach gets the results from the classification-based annotation and service querying. The non-fuzzy based approach defines that the relevance scores of the service annotation and the service querying parts are crisp values. It retrieves all services that are annotated to the service concepts from the semantic service querying part. Details of the non-fuzzy based service retrieval approach are presented in Section 7.3 .

2) Fuzzy-based approach applies the fuzzy logic to retrieve the services. It considers that the relevance values of the service annotation, service querying and service retrieval are fuzzy. Fuzzy variables of those relevance values and fuzzy rules, which are used for calculating the relevance score of the service retrieval, are defined. A service will be retrieved if its relevance score is greater than a defined Retrieval Threshold *(RT)*. Like the non-fuzzy based approach, the fuzzy-based approach is subdivided into the term-based, the vector-based and the classification-based approaches. Details of the fuzzy-based service retrieval approach are provided in Section 7.4 .

4.7 Overview of the Solution for Validation of the proposed Methodology for Semantic Service Retrieval

This section presents an overview of the solution for validation of the proposed methodology. Based on the research issue 5 in Chapter 3, the research question was provided as follows.

"How can we validate the developed semantic service annotation, semantic service querying, and semantic service retrieval techniques?"

In order to validate the developed approaches, a prototype of the service retrieval system with the transport service domain is built. Then, the performances of the developed semantic-based approaches and of the existing approaches are compared. In order to evaluate the performance of semantic-based approaches, four performance measures from the area of information retrieval (Baeza-Yates & Ribeiro-Neto 1999), namely precision, recall, f-measure, and fallout rate are applied. Annotation rate, querying rate, and retrieval rate are used to evaluate the service annotation, service querying, and service retrieval respectively. Moreover, a combination of the previous measures is also used for the performance evaluation. The details of those measures for service annotation, service querying, and service retrieval are described as follows.

4.7.1 Validation of Semantic Service Annotation

Precision

Precision is used to measure the accuracy of the service annotation approach. In this thesis, the precision for a single service is the fraction of the annotated concepts that are relevant to a service for all annotated concepts.

$$Precision(S) = \frac{|annotated\ concepts| \cap |relevant\ concepts|}{|annotated\ concepts|}$$

In this experiment, the average precision is the summation of precision values for each service divided by the number of services (n).

$$AVG\_Precision(T) = \frac{\sum_{i=1}^{n} Precision(S_i)}{n}$$

Recall

Recall is used to measure the effectiveness of the service annotation approach. In this thesis, the recall for a single service is the fraction of the annotated concepts which are relevant to a service for all relevant concepts.

$$Recall(S) = \frac{|annotated\ concepts| \cap |relevant\ concepts|}{|relevant\ concepts|}$$

Like the precision, the average recall is the summation of recall values for each service divided by the number of services (n).

$$AVG\_Recall(T) = \frac{\sum_{i=1}^{n} Recall(S_i)}{n}$$

F-measure

F-measure is used to weigh the performance value between precision and recall. In this thesis, the f-measure can be represented as follows.

$$F\_measure(S) = \frac{1}{\frac{1}{Precision(S)} + \frac{1}{Recall(S)}}$$

The average f-measure is the summation of the f-measure values for each service divided by the number of services (n).

$$AVG\_F\_measure(T) = \frac{\sum_{i=1}^{n} F\_measure(S_i)}{n}$$

Fallout Rate

Fallout Rate is used to measure the failure that the approach annotates non-relevant concepts. That is, the fallout rate value of a good annotation approach should be low.

$$Fallout\_Rate(S) = \frac{|annotated\ concepts| \cap |non\_relevant\ concepts|}{|non\_relevant\ concepts|}$$

The average fallout rate is the summation of the fallout rate values for each service divided by the number of services (n).

$$AVG\_Fallout\_Rate(T) = \frac{\sum_{i=1}^{n} Fallout\_Rate(S_i)}{n}$$

Annotation Rate

Annotation rate is used to measure the annotation capability of the developed approach. In this thesis, the annotation rate is the fraction of the number of annotated services for all services.

$$Annotation\_Rate = \frac{|annotated\ services|}{|services|}$$

All-measures

All-measures is used to measure the overall performance of the developed approach. In this thesis, the all-measures is the combination of average precision, average recall, average f-measure, average fallout rate, and annotation rate. This thesis defines that precision is the most important factor of the performance measure, while other factors have equal priority. Consequently, the weights of average precision and other factors are 0.4 and 0.15 respectively.

$$All\_Measures = (0.4 * AVG\_Precision) + (0.15 * (AVG\_Recall + AVG\_F\_Measure + (1 - AVG\_Fallout) + Annotation\_Rate)$$

4.7.2 Validation of Semantic Service Querying

Precision

Precision is used to measure the accuracy of the service querying approach. In this thesis, the precision for a single query is the fraction of the queried concepts that are relevant to a query for all queried concepts.

$$Precision(Q) = \frac{|queried\ concepts| \cap |relevant\ concepts|}{|queried\ concepts|}$$

In this experiment, the average precision is the summation of precision values for each query divided by the number of queries.

$$AVG\_Precision(T) = \frac{\sum_{i=1}^{n} Precision(Q_i)}{n}$$

Recall

Recall is used to measure the effectiveness of the service querying approach. In this thesis, the recall for a single query is the fraction of the queried concepts which are relevant to a query for all relevant concepts.

$$Recall(Q) = \frac{|queried\ concepts| \cap |relevant\ concepts|}{|relevant\ concepts|}$$

Like the precision, the average recall is the summation of recall values for each query divided by the number of queries.

$$AVG\_Recall(T) = \frac{\sum_{i=1}^{n} Recall(Q_i)}{n}$$

F-measure

F-measure is used to weigh the performance value between precision and recall. In this paper, the f-measure can be represented as follows.

$$F\_measure(Q) = \frac{1}{\frac{1}{Precision(Q)} + \frac{1}{Recall(Q)}}$$

The average f-measure is the summation of the f-measure values for each query divided by the number of queries.

$$AVG\_F\_measure(T) = \frac{\sum_{i=1}^{n} F\_measure(Q_i)}{n}$$

Fallout Rate

Fallout Rate is used to measure the failure that the approach queries non-relevant concepts. That is, the fallout rate value of a good service querying should be low.

$$Fallout\_Rate(Q) = \frac{|queried\ concepts| \cap |non\_relevant\ concepts|}{|non\_relevant\ concepts|}$$

The average fallout rate is the summation of the fallout rate values for each query divided by the number of queries.

$$AVG\_Fallout\_Rate(T) = \frac{\sum_{i=1}^{n} Fallout\_Rate(Q_i)}{n}$$

Querying Rate

Querying rate is used to measure the querying capability of the developed approach. In this thesis, the querying rate is the fraction of the number of queried query samples for all testing query samples.

$$Querying\_Rate = \frac{|queried\ query\ samples|}{|testing\ query\ samples|}$$

All-measures

All-measures is used to measure the overall performance of the developed approach. In this thesis, the all-measures is the combination of precision, recall, f-measure, fallout rate, and querying rate.

$$All\_Measures = (0.4 * AVG\_Precision) + (0.15 * (AVG\_Recall + AVG\_F\_Measure + (1 - AVG\_Fallout) + Querying\_Rate)$$

4.7.3 Validation of Semantic Service Retrieval

Precision

Precision is used to measure the accuracy of the retrieval system. In this thesis, the precision for a single query is the fraction of the retrieved services that are relevant to a query for all retrieved services.

$$Precision(Q) = \frac{|retrieved\ services| \cap |relevant\ services|}{|retrieved\ services|}$$

In this experiment, the average precision is the summation of precision values for each query divided by the number of queries.

$$AVG\_Precision(T) = \frac{\sum_{i=1}^{n} Precision(Q_i)}{n}$$

Recall

Recall is used to measure the effectiveness of the retrieval system. In this thesis, the recall for a single query is the fraction of the retrieved services which are relevant to a query for all relevant services.

$$Recall(Q) = \frac{|retrieved\ services| \cap |relevant\ services|}{|relevant\ services|}$$

Like the precision, the average recall is the summation of recall values for each query divided by the number of queries.

$$AVG\_Recall(T) = \frac{\sum_{i=1}^{n} Recall(Q_i)}{n}$$

F-measure

F-measure is used to weigh the performance value between precision and recall. In this thesis, the f-measure can be represented as follows.

$$F\_measure(Q) = \frac{1}{\frac{1}{Precision(Q)} + \frac{1}{Recall(Q)}}$$

The average f-measure is the summation of the f-measure values for each query divided by the number of queries.

$$AVG\_F\_measure(T) = \frac{\sum_{i=1}^{n} F\_measure(Q_i)}{n}$$

Fallout Rate

Fallout Rate is used to measure the failure that the approach retrieves non-relevant services. That is, the fallout rate value of a good retrieval system should be low.

$$Fallout\_Rate(Q) = \frac{|retrieved\ services| \cap |non - relevant\ services|}{|non - relevant\ services|}$$

The average fallout rate is the summation of the fallout rate values for each query divided by the number of queries.

$$AVG\_Fallout\_Rate(T) = \frac{\sum_{i=1}^{n} Fallout\_Rate(Q_i)}{n}$$

Retrieval Rate

Retrieval rate is used to measure the retrieval capability of the developed approach. In this thesis, the retrieval rate is the fraction of the number of retrieved query samples for all testing query samples.

$$Retrieval\_Rate = \frac{|retrieved\ query\ samples|}{|testing\ query\ samples|}$$

All-measures

All-measures is used to measure the overall performance of the developed approach. In this thesis, the all-measures is the combination of precision, recall, f-measure, fallout rate, and retrieval rate.

$$All\_Measures = (0.4 * AVG\_Precision) + (0.15 * (AVG\_Recall + AVG\_F\_Measure + (1 - AVG\_Fallout) + Retrieval\_Rate)$$

4.8 Conclusion

This chapter presents the overview solution of the semantic service retrieval. To understand the proposed problems and solutions, the definition of service, service ontology and service knowledge base were formally defined. In this thesis, the focus is on retrieving only online business services which are represented by service metadata or SDE metadata. The SDE metadata and the service ontology are stored in the service knowledge base. Given a query, relevant services are retrieved from the service knowledge base. Based on the research issues and questions outlined in Chapter 3, the solution of the semantic service retrieval are divided into four solutions; the solution of semantic service annotation, semantic service querying, semantic service retrieval and validation of the proposed methodology. The semantic service annotation aims to annotate services to their relevant concepts, while the semantic service querying aims to query service concepts that are relevant to a given query. Then, the semantic service retrieval retrieves relevant services by considering the service annotation and the queried concepts from the previous step. To validate the proposed methodologies, the information retrieval measures, precision, recall, f-measure and fallout rate, were applied.

In the next chapter, the proposed semantic service annotation methodology, which is the first important part of the semantic service retrieval, is explained in detail.

Chapter 5 Semantic Service Annotation

5.1 Introduction

Nowadays, many businesses advertise their own service information to customers via websites. Typically customers search for this information using search engines, like Google or Bing. However these search engines are not designed to search for services and as a result, customers often retrieve a large amount of irrelevant information. When a customer needs to complete a task with more than one service, the task has to be separated into subtasks and search for each sub-task performed separately. Because generic systems on the Web do not support a service perspective, a framework that combines and represents business services and knowledge sharing is required (Nachira et al. 2007).

Semantic technologies have played an important role in service retrieval and service querying. Annotating services semantically enables machines to understand the purpose of services and can further assist in intelligent and precise service retrieval, selection and composition. A key issue in semantically annotating services is the manual nature of service annotation. Manual service annotation requires a large amount of time and updating happens infrequently, hence annotations may get out-of-date due to service description changes. Although some researchers have studied semantic service annotation, they have only focused on web services not business service information. Moreover, their approaches are semi-automated, and still require service providers to select appropriate service annotations.

Based on the previous limitations of existing approaches, this thesis proposes new semantic approaches that automatically annotate a service to relevant service concepts. Given a service (SDE) and the domain-specific service ontology, service terms and concept terms are extracted and sent to the semantic service annotation module in order to calculate relevance scores between the SDE and every service concept in the ontology. The aim of the service annotation module is to link or annotate the SDE to relevant service concepts. For example, a service named "Abel Rent A Car" is annotated to two service concepts named "Bus Rental" and "Truck Rental".

In this chapter, three kinds of semantic service annotation approaches, namely, *ECBR*, *Vector-based* and *Classification-based approach*, are proposed in order to automatically annotate services to their own relevant concepts. The main workflows of all proposed approaches are identical. A difference among them is how to compute the relevance scores between service metadata and ontological service concepts.

In the next section, the workflow of the semantic service annotation methodology is presented. Each step of the methodology and details of all proposed service annotation approaches are then described in the rest of the chapter.

5.2 Semantic Service Annotation Methodology

In this section, the workflow of the semantic service annotation methodology is presented in Figure 5.1. To annotate an SDE *S* to its relevant service concepts, the steps involved in the semantic service annotation methodology are as follows:

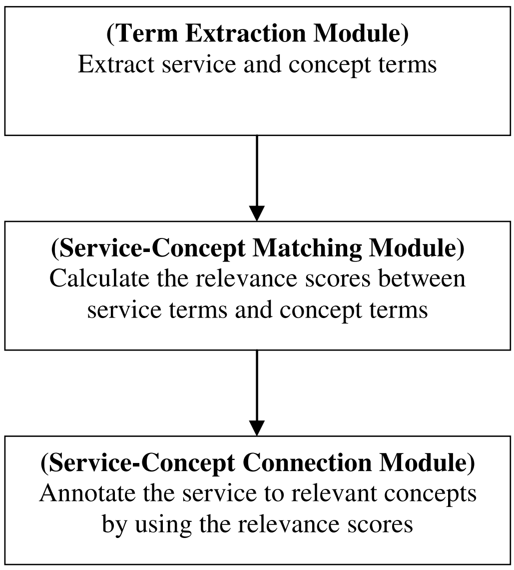


Figure 5.1 Flow Chart of the Semantic Service Annotation Methodology.

1. **The term extraction module** extracts service terms and concept terms from service description and service provider name of *S,* and concept descriptions of each service concept in the service ontology.

2. **The service-concept matching module** calculates relevance scores between service terms of *S* and concept terms of each service concept.

3. **The service-concept connection module** annotates or links the service *S* to relevant service concepts based on the relevance scores that are returned from step 2.

The semantic service annotation methodology comprises three sub-modules; namely *term extraction*, *service-concept matching,* and *service-concept connection module*, which are described in Section 5.3 , 5.4 and 5.5 respectively. As mentioned in the previous section, the proposed annotation approaches consist of the same main working steps. The difference among them is the relevance score calculation. Therefore, the content of the service-concept matching module is separated into three sub-sections; the service-concept matching modules for ECBR, vector-based and classification-based annotation approach in Section 5.4.1 , 5.4.2 and 5.4.3 respectively. Finally, the experimental results of semantic service annotation are displayed in Section 5.6 and the work is concluded in Section 5.7

5.3 Term Extraction Module

The first step of the semantic service annotation methodology is to extract service terms and concept terms from services and ontological service concepts. Those service and concept terms are fed to the service-concept matching module to compute relevance values between services and concepts. In this thesis, the term extraction module is divided into two parts; namely service term extraction and concept term extraction. The pseudo-code of the term extraction module is shown in Figure 5.2. The service term extraction extracts service terms from an SDE, while the concept term extraction extracts concept terms from a domain-specific service ontology.

```
Program Term_Extraction(SDE S, Ontology O)
   service_terms = Service_Term_Extraction(S)
   concept_terms = Concept_Term_Extraction(O)
end Program
```

Figure 5.2 Pseudo-code of the term extraction module

5.3.1 Service Term Extraction

Given an SDE, the service term extraction sub-module extracts the service terms which are applied for calculating a relevance score between the SDE and service concepts. Each SDE contains information of the service such as service name, service provider, service description, address, location, and contact detail. The structure of any SDE is presented in Figure 5.3. The principal idea of the proposed annotation approaches is to find the semantics of a service by considering terms in a service provider name and a service description. Pseudo-code of the service term extraction is shown in Figure 5.4.

```
Structure SDE
   id
   name
   provider
   description
   address
   location
   contact
end Structure
```

Figure 5.3 Data structure of a service description entity (SDE)

```
Program Service_Term_Extraction(SDE S)
   s_description = S.service_description
   s_provider = S.provider
   Terms:[t1, t2, ..., tn] = split s_description and s_provider
      by using space and punctuations
   service_terms = create an empty list
   For each ti in Terms :
      tstem = stem ti
      if (tstem is not a stop word)
         search tstem in WordNet3.1
         if (tstem is in WordNet 3.1 and tstem is noun)
            add tstem into service_terms
         end if
      end if
   end for
   return service_terms
End Service_Term_Extraction
```

Figure 5.4 Pseudo-code of the service term extraction module

Given an SDE *S,* its description and provider name are split into terms by using spaces and punctuation like period, comma, colon, semicolon, question mark, hyphen and exclamation mark. The module stems all split terms by deleting suffixes and then removes all stop words, for example, a, an, any. The remaining terms are searched in WordNet3.1. If a term has the meaning in WordNet3.1 and its type is a noun, the term is added to the service term list. After processing all split terms, the service term list is returned.

For example, an SDE *S* is provided by “Corporate Air”, and its description is “24/7 Australia Wide and International Charter - Established 1972”. After *S* is processed, the service term extraction module gets service terms such air, 24/7, Australia and charter.

5.3.2 Concept Term Extraction

The concept term extraction sub-module extracts the concept terms from the service ontology and uses those terms for computing a relevance score in the next service-concept matching module. The structure of each service concept in the service ontology is shown in Figure 5.5. Each service concept contains information such as concept name and concept descriptions. This module generates two types of the concept terms; 1) concept terms of each service concept and 2) concept terms of all service concepts.

```
Structure Service Concept
   id
   name
   descriptions
end Structure
```

Figure 5.5 Data structure of a service concept

The concept terms of each service concept is a list that contains *n* lists of terms that are extracted from a concept description, where *n* is a number of concept descriptions of a service concept. For example, a service concept "Airline_Agent" has two concept descriptions; namely "Airline Agent" and "Flight Agent". The concept terms of this service concept is {{"airline", "agent"}, {"flight", "agent"}}. These concept terms are used for computing the relevance score in the ECBR annotation approach in Section 5.4.1

The pseudo-code of the concept term extraction for each service concept is shown in Figure 5.6. To extract the concept terms, the thesis focuses on only actual service concepts which are at the bottom of the service ontology. Similar to the service term extraction, for each service concept, the concept term extraction module splits its concept descriptions into terms, stems those terms, and removes all stop words. Terms that are a noun and exist in WordNet3.1 are selected and added to the concept term list. After processing all descriptions of all actual service concepts, the concept term list is returned.

```
Program Concept_Term_Extraction_Each_Concept(Service_Concept C)
   c_description:[desc_1, desc_2, ..., desc_n] = concept_i.descriptions
   concept_terms = create an empty list
   For each desc_i in c_description:
      Terms:[t_1, t_2, ..., t_n] = split desc_i by using space and punctuations
      concept_terms_each_desc = create an empty list
      For each t_i in Terms :
         t_stem = stem t_i
         if (t_stem is not a stop word)
            search t_stem in WordNet3.1
            if (t_stem is in WordNet3.1 and t_stem is noun)
               add t_stem into concept_terms_each_desc
            end if
         end if
      end for
      add concept_terms_each_desc into concept_terms
   end for
   return concept_terms
End Concept_Term_Extraction_Each_Concept
```

Figure 5.6 Pseudo-code of the concept term extraction module for each service concept

```
Program Concept_Term_Extraction_All_Concept(Ontology O)
   For each concept_i in O :
      if (concept_i is an actual concept)
         c_description:[desc_1, desc_2, ..., desc_n] = concept_i.descriptions
         For each desc_i in c_description:
            Terms:[t_1, t_2, ..., t_n] = split desc_i by using space and
punctuations
            concept_terms = create an empty list
            For each t_i in Terms :
               t_stem = stem t_i
               if (t_stem is not a stop word)
                  search t_stem in WordNet3.1
                  if (t_stem is in WordNet3.1 and t_stem is noun)
                     add t_stem into concept_terms
                  end if
               end if
            end for
         end for
      end if
   end for
   return concept_terms
End Concept_Term_Extraction_All_Concept
```

Figure 5.7 Pseudo-code of the concept term extraction module for all service concepts

In contrast, the concept terms of all service concepts is a list of all terms that are extracted from every concept description of all actual service concepts. For example, the ontology has two actual service concepts named "Airline_Agent" and "Bus_Rental". The descriptions of the concept "Airline_Agent" are "Airline Agent" and "Flight Agent", while the descriptions of the concept "Bus_Rental" are "Bus Rental" and "Bus Charter". The concept terms of all service concepts in the ontology are {"airline", "agent", "flight", "bus", "rental", "charter"}. These concept terms are used in the vector-based and the classification-based annotation approach in Section 5.4.2 and 5.4.3 respectively.

The pseudo-code of the concept term extraction for each service concept is shown in Figure 5.7. The process of this algorithm is quite similar to the algorithm in Figure 5.6, but this algorithm combines all terms of every service concept into a term list.

5.4 Service-Concept Matching

The service-concept matching module receives service terms from the service term extraction module and connects to the domain-specific service ontology. The matching scores between those service terms and concept descriptions in service concepts are calculated in order to retrieve service concepts that are relevant to an SDE. To compute the service-concept matching scores, in this thesis, three semantic annotation approaches; namely Extended Case-Based Reasoning (ECBR)-based, Vector-based and Classification-based annotation approach, are proposed and described in Section 5.4.1 , 5.4.2 and 5.4.3 respectively.

5.4.1 ECBR annotation approach

The ECBR model was proposed by (Dong, Hussain & Chang 2011). Their research proposed a framework for a service focused crawler. They applied the ECBR model to calculate the similarity between SDE metadata and service concepts. Similar to our work, the ECBR model matches terms in SDE metadata with terms in service concepts. Given an SDE *S* and a service concept *C,* every term in *S* will be compared with every term in descriptions of *C*. For each description of *C,* if a term in *S* exactly matches one in the concept description, the matching score is 1;

otherwise, the score is 0. Matching scores of all pairs of service-concept terms are then summed up and normalized. Because the service concept may have many concept descriptions, the similarity score between an SDE and a service concept for the ECBR model is the maximum similarity score between SDE and each concept description.

Inspired by the previous ECBR model for focused crawling, the ECBR approach for annotating services is proposed. The ECBR algorithm presented in Dong et al. (2011) is improved by considering synonyms of terms and using the average similarity score instead of the maximum score. This is because service terms of an SDE *S* should be similar to all the service descriptions of a concept *C* if *S* is relevant to *C*. Moreover, using the synonyms for computing the relevance score is beneficial in the case that a service term and a concept term are not exactly matched but they refer to the same meaning.

Given an SDE *S* and a service concept *C,* the service terms of *S* and concept terms of *C* are generated by using the algorithm in Section 5.3.1 and 5.3.2 respectively. The service term list of *S* is $\{st_1, st_2, ..., st_m\}$; where st_i is the *i*-th service term and *m* is a number of service terms of *S*. The concept terms of *C* are $\{\{ct_{(1,1)}, ct_{(1,2)}, ..., ct_{(1,l1)}\}, \{ct_{(2,1)}, ct_{(2,2)}, ..., ct_{(2,l2)}\}, ..., \{ct_{(n,1)}, ct_{(n,2)}, ..., ct_{(n,ln)}\}\}$, where $ct_{(j,k)}$ is the *k*-th concept term of the *j*-th concept description of *C, lj* is a number of concept terms of the *j*-th concept description of *C,* and *n* is a number of concept descriptions of *C*. The pseudo-code of the service-concept matching method for the ECBR annotation approach is shown in Figure 5.8.

The relevance score between an SDE *S* and a service concept *C* is the average value of the similarities between the service terms of *S* $\{st_1, st_2, ..., st_m\}$ and the concept terms of each concept description C_i $\{ct_{(1,1)}, ct_{(1,2)}, ..., ct_{(1,li)}\}$. The similarity value between *S* and C_i is the summation of matching values between every service term and every concept term divided by the number of service terms, matching synonyms and concept terms. The matching value between a service term and a concept term is 1 if the service term and the concept term are identical. The matching value is 0.5 if the service term is a synonym of the concept term, otherwise the value is 0.

```
Input:  A list of extracted service terms ST: {st1, st2, …, stm}
        A list of extracted concept terms CT:
           {{ct(1,1), ct(1,2), ..., ct(1,l1)},
            {ct(2,1), ct(2,2), ..., ct(2,l2)}, ...,
            {ct(n,1), ct(n,2), ..., ct(n,ln)}}
Output: A relevance score RS
Algorithm Service-concept matching for ECBR annotation approach
RS = 0
for each {ct(i,1), ct(i,2), ..., ct(i,li)} in CT
   RSi = 0
   syn = 0
   for each stj in ST
      for each ct(i,k) in {ct(i,1), ct(i,2), ..., ct(i,li)}
         if stj is equal to ct(i,k)
            RSi = RSi + 1
         else if stj is a synonym of ct(i,k)
            RSi = RSi + 0.5
            syn = syn + 1
         end if
      end for
   end for
   RSi = RSi/((m+syn)*li)
   RS = RS + RSi
end for
RS = RS/n
```

Figure 5.8 Pseudo-code of service-concept matching by ECBR annotation approach

5.4.2 Vector-based annotation approach

Apart from considering the number of matching terms between service and concept terms in ECBR approach, this thesis also proposes new automated approaches for service annotation called the VSM-based approach and Extended VSM (EVSM)-based approach. Business services are annotated by linking to relevant service concepts in the domain-specific service ontology. Each business service and each service concept are represented by a vector. A service is then linked to a concept if their representing vectors are similar. Inspired by the work of (Garcés, Olivas & Romero 2003; Olivas, Garcés & Romero 2003) on an FIS-CRM model – a concept-based mechanism for representing documents – services are represented by using an Extended VSM-based model. The proposed model weights terms in a service relating their synonyms.

The pseudo-code of the service-concept matching method by the vector-based annotation approach is shown in Figure 5.9. Given an SDE *S* and a service concept *C*, the service terms of *S* and the concept terms of *C* from the term extraction module in Section 5.3 are fed to the representation module. In the representation module, SDE and service concepts in a domain-specific service ontology are represented by vectors. In the matching module, the cosine similarity between vectors of an SDE and a service concept are calculated to find their degree of association. The similarity values range from 0 to 1. If an SDE completely differs from a service concept, the similarity value is 0. In contrast, the value is 1 if an SDE and a service concept are the same. This means that the higher the similarity value, the more closely an SDE relates to a service concept. The working process of the VSM-based and EVSM-based approaches is shown in Figure 5.10 and Figure 5.11 respectively.

```
Input:  A list of extracted service terms ST: {st_1, st_2, …, st_m}
        A list of extracted concept terms CT:
           {{ct_(1,1), ct_(1,2), ..., ct_(1,l1)},
            {ct_(2,1), ct_(2,2), ..., ct_(2,l2)}, ...,
            {ct_(n,1), ct_(n,2), ..., ct_(n,ln)}}
Output: A relevance score RS
Algorithm Service-concept matching for Vector-based annotation approach
SDE-Vector = representation_module(ST)
Concept-Vector = representation_module(CT)
RS = cosine_similarity(SDE-Vector, Concept-Vector)
```

Figure 5.9 Pseudo-code of service-concept matching for Vector-based annotation approach

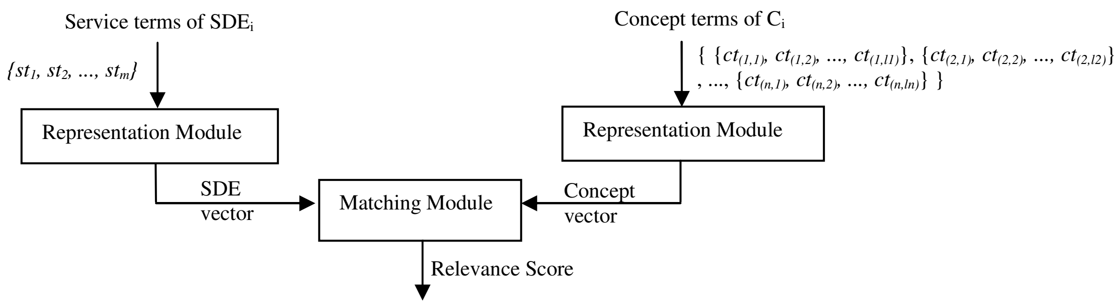


Figure 5.10 Workflow of the VSM-based annotation approach

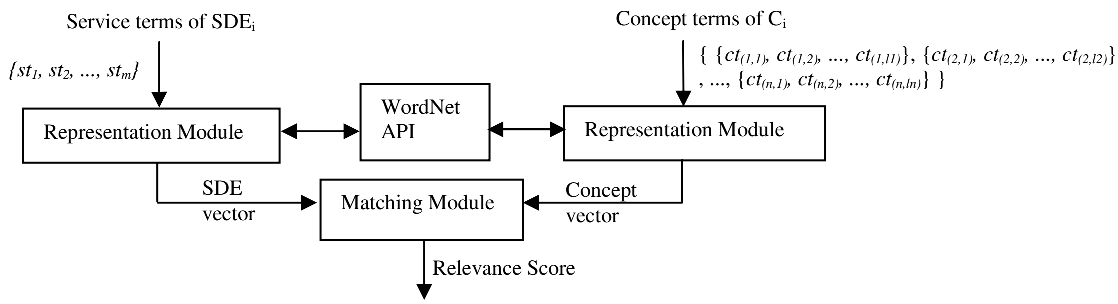


Figure 5.11 Workflow of the EVSM-based annotation approach

Although the VSM-based and extended VSM-based approaches appear similar, their representation modules are different. The VSM-based approach creates the representing vectors of SDEs and service concepts using a basic vector space model (VSM). The extended VSM-based approach also includes the semantics of terms of the original VSM. Figure 5.11 shows that the box, ‘WordNet API’, is included in the extended VSM-based approach for the purposes of retrieving synonyms to create a vector. By contrast, the VSM-based approach is not concerned with semantics, and ‘WordNet API’ is not required. Details of the representation module are described as follows.

Representation Module

The purpose of this module is to generate a vector that represents each SDE or service concept. These vectors are used in the next step to determine the similarities between them.

1) Representation Module for VSM-based Service Annotation Approach

In this section, we present a method to create a vector for use in a vector space model (VSM) - also known as a term vector model – which is typically used to represent text documents. In this thesis, both SDE metadata and service concepts are defined as text documents, each containing a name and its associated descriptions. VSM model, therefore, is used to create vectors for both SDE metadata and service concepts.

SDE metadata and service concepts are represented as vectors with *N* dimensions, in the VSM, where *N* is the number of concept terms in the service knowledge base. The value of the vector is the number of times a term occurs in a document. Vectors for an SDE are created by extracting the SDE name and SDE description as a document. Vectors for a service concept are created by extracting only its concept descriptions, again as a document.

2) Representation Module for Extended VSM-based Service Annotation Approach

Inspired by the work of (Garcés, Olivas & Romero 2003; Olivas, Garcés & Romero 2003) on a concept-based mechanism for representing documents, a representing vector is generated by considering not only the number of times a term occurs but also the synonyms for each term in an SDE and service concepts. The resulting Extended VSM (EVSM)-based vector is a semantic representation, while VSM-based vector is non-semantic. The main purpose of this module is to increase the weight of a term if its synonyms appear in an SDE or a service concept. The pseudo-code of the representation module for an extended VSM-based annotation approach is shown in Figure 5.12.

```
program Representation_Module
   begin
      IF (a document is an SDE) THEN
         terms = noun words from SDE Name and Description
      ELSE IF (a document is a service concept) THEN
         terms = noun words from concept descriptions
      END IF
      Create VSM-based vector
      For each term in terms
         synonyms = get synonyms of the term from WordNET API
         IF synonyms is not empty THEN
            For each syn in synonyms
               weight(syn) = C(syn) + C(term)*sqrt(1/T(term))
            END FOR
            weight(term) = C(term) + C(term)*sqrt(1/T(term))
         ELSE weight(term) = C(term)
         END IF
      END FOR
   end.
```

Figure 5.12 Pseudo-code of the representation module for the Extended VSM-based service annotation approach

First, the module creates a VSM-based vector for an SDE or a service concept, according to the workflow described in Figure 5.11. The module, then, connects to the WordNet API to find a set of synonyms for each term. After retrieving a set of synonyms for each term in the vector, the module readjusts the weight of the term **x** and its synonyms s_x in an SDE or a service concept as follows:

$$Weight(x) = C(x) + C(x) * \sqrt{1/_{T(x)}}$$

$$Weight(s_x) = C(s_x) + C(x) * \sqrt{1/_{T(x)}}$$

Where: $Weight(x)$ is the readjustment weight of term x; $C(x)$ is the number of occurrences of term x; $T(x)$ is number of words having the same meaning as term x; $Weight(s_x)$ is the readjustment weight of a synonym for term x; and $C(s_x)$ is the number of occurrences of a synonym for term x.

For example, the term 'storage' occurs three times in the SDE description of SDE1 and the term 'warehousing' occurs twice in the SDE description of SDE2. The term 'storage' and the term 'warehousing' are synonyms. The representing VSM-based and extended VSM-based vectors of SDE1 and SDE2 are shown in Figure 5.13 and Figure 5.14 respectively.

	storage	warehousing
SDE_1	3	0
SDE_2	0	2

Figure 5.13 VSM-based vectors of SDE_1 and SDE_2

	storage	warehousing
SDE_1	5.121	2.121
SDE_2	1.414	3.414

Figure 5.14 Extended VSM-based vectors of SDE_1 and SDE_2

5.4.3 Classification-based annotation approach

Apart from using term matching and vector space model based methods, the thesis also proposes a classification based approach for semantic service annotation. Given a service and a domain-specific service ontology, the service is annotated to ontological concepts whose descriptions relate to the service description. To solve this problem, this thesis regards the service annotation task as a classification problem. That is, service concepts in the ontology act as service classes that any service can belong to. SDE metadata are classified into groups based on existing service concepts. In this case, service concepts act as classes in the classification problem. If an SDE *S* is classified as service concept *C,* it means that *S* should be relevant to *C*. The output of this module is a set of relevant concepts with their relevance scores.

To solve the service classification problem, both artificial neural networks and machine learning algorithms are applied. The Neural Network (NN)-based annotation approaches apply Feed-Forward (FF) neural networks and Redial Basis Function (RBF) networks, while the Machine Learning (ML)-based annotation approach applies K-Nearest Neighbor (KNN), Classification Tree (CT) and Support Vector Machine (SVM) algorithm.

The service terms from the term extraction module are sent to the service-concept matching module for classification-based service annotation approach in order to calculate the relevance value and discover the relevant service concepts. In this case, the service-concept module contains a service classifier which is able to classify services into proper service classes. In this thesis, an SDE can be annotated into either a single service concept or multiple concepts. If the service description is specific, the SDE may be annotated to only one service concept. On the other hand, if the service description is quite general, the SDE should be annotated to many concepts. Consequently, two types of service classifier such as single-label and multi-label service classifier are proposed. Given an SDE, the output of the first type is only one relevant concept, while the output of the second one is zero or more concepts. The descriptions of both classification types are explained as follows.

Single-Label Service Classifier

The single-label classifier is able to classify an SDE into the most relevant service class. The classifier applies supervised learning algorithms to learn a pattern of the service annotation. In this case, the input of the classifier is the service terms, while the output is a proper service class which is defined by a service concept in the service ontology. To create the single-label classifier, both artificial neural network and machine learning algorithm are applied.

1) Neural Network-based Single-Label Classifier

Two-layered feed-forward neural networks; Multi-Layer Perceptron (MLP) and the Radial-Basis Function networks (RBF), are used for creating the classifier. The structure of the neural network is shown in Figure 5.15.

Input nodes of the neural network refer to concept terms of all service concepts. As mentioned before, each service concept consists of the concept name and concept description. The concept terms of a service concept *C* are terms that appear in concept descriptions of *C*. To extract the concept terms, the same technique with the service term extraction is used. For example, the concept terms of the concept "Airline Agent", whose descriptions are "Airline Agent" and "Flight Agent", are "airline", "flight", and "agent". Thus, the number of input nodes is equal to the number of concept terms from all service concepts.

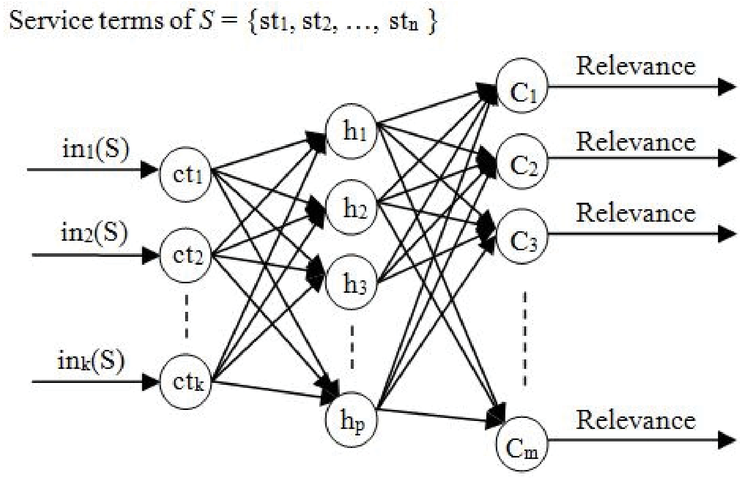


Figure 5.15 The structure of feed-forward neural network for the single-label service classifier

Given service terms of the SDE from the previous step, the input value of each input node is calculated from the occurrences of the concept term in the SDE. Synonyms of the concept term are also used to define the input value. Given an SDE *S,* the input value of the input node *i*; $in_i(S)$, is calculated as follows

$$in_i(S) = \sum_{j=1}^{n} f(st_j, ct_i)$$

$$f(x,y) = \begin{cases} 1, & if\ x\ is\ equal\ to\ y \\ 0.5, & if\ x\ is\ a\ synonym\ of\ y \\ 0, & otherwise \end{cases}$$

, where the SDE *S* consists of *n* service terms, st_j is the *j*-th service term of *S,* and ct_i is the concept term of input node *i.*

The input values of the neural network present the similarities between terms in the SDE and terms in the service concept description. That is, the SDE *S* should be classified as a service concept *C* if terms in *S* are similar to terms in *C.*

The hidden layer consists of sigmoid neurons. The number of hidden neurons ranges from 10 to 100 neurons. Output nodes of the neural network refer to actual service concepts of domain-specific service ontology. Therefore, the number of output nodes depends on the number of actual service concepts in service ontology. Output value in each output node is the relevance score between input SDE and the service concept in that node.

The scaled conjugate gradient back-propagation algorithm, which assists the artificial neural network to converge faster than the basic back-propagation algorithm, is applied in order to train the neural network.

2) Machine Learning-based Single-Label Classifier

Apart from using the neural network-based classifier, machine learning algorithms; K-Nearest Neighbor (KNN) and Classification Tree (CT), are also applied for classifying a service into the most relevant service concept. KNN and CT algorithm are multi-classes classifications which are suitable for a single-label service classification task. That is, an SDE *S* is able to be categorized into one service concept from several actual service concepts in the ontology.

The input of both KNN-based and CT-based classifier is an input-term vector with *n* elements; where *n* is a number of concept terms from the algorithm in Figure 5.7. On the other hand, the output of the ML-based classifier is a service class or service concept. Similar to the input values for the NN-based classifier, given an SDE *S,* the input value of element *i* of the input-term vector; $\text{input\_vector}_i(S)$, is calculated as follows:

$$input\_vector_i(S) = \sum_{j=1}^{n} f(st_j, ct_i)$$

$$f(x,y) = \begin{cases} 1, & if\ x\ is\ equal\ to\ y \\ 0.5, & if\ x\ is\ a\ synonym\ of\ y \\ 0, & otherwise \end{cases}$$

, where the SDE *S* consists of *n* service terms, st_j is the *j*-th service term of *S,* and ct_i is the *i*-th concept term of the concept term list.

Multi-Label Service Classifier

The multi-label classifier is able to classify an SDE into several relevant service classes. Like the single-label classifier, the multi-label classifier applies both artificial neural network and machine learning algorithms for learning a pattern of the service annotation. The input of the classifier is the service terms, the same as the single-label classifier. In contrast, the output is a set of relevant service classes. The multi-label annotation is suitable for annotating services in which their descriptions are general. As a result, those services are related to many service concepts. For example, a service provider "Abel Rent A Car" has a service description as "Rent New Cars From $ 29* A Day, Trucks From $ 66* Mates Rates". It may be annotated to both "Car Rental" and "Truck Rental" service concepts. The multi-label service classifier is divided into two types such as the neural network based and the machine-learning based classifier.

1) Neural Network-based Multi-Label Classifier

Although an MLP and an RBF are also applied in the multi-label classifier, their structure differs from the structure in the single-label approach. The structure of the neural network for the multi-label classifier is shown in Figure 5.16.

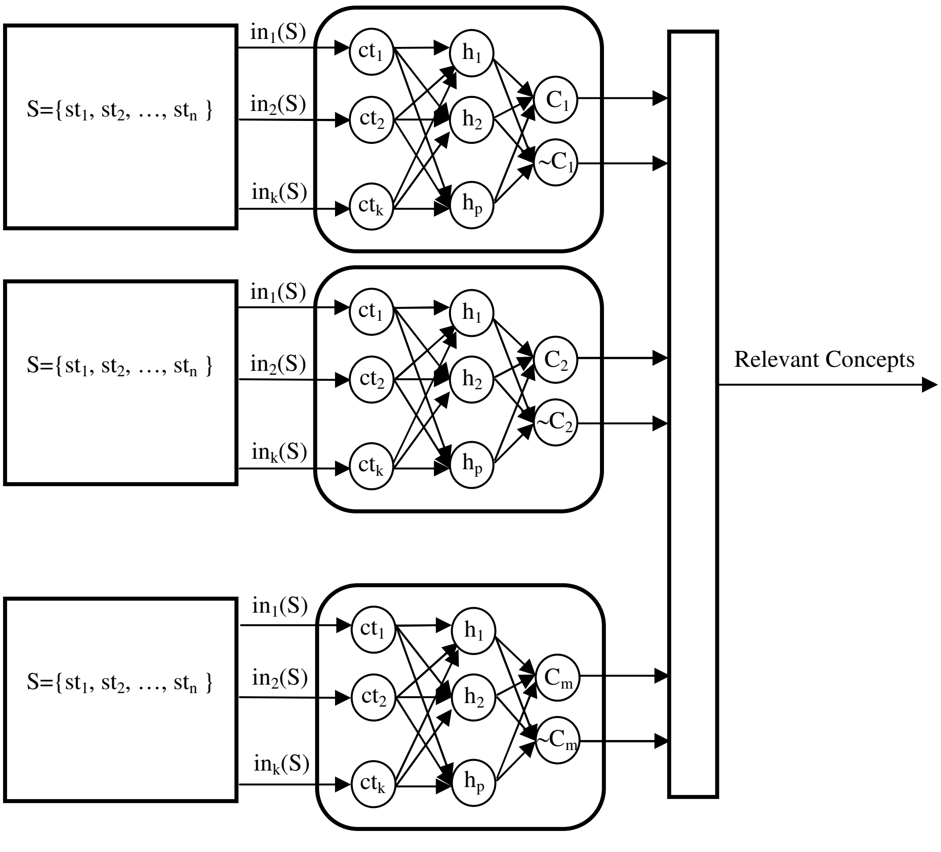


Figure 5.16 The structure of feed-forward neural network for the multi-label service classifier

The multi-label classification uses many multilayer feed-forward neural networks, while the single-label one uses only one neural network to classify a service. Regarding a single-label classification, a neural network is applied for multiclass classification. A number of classes in the network are equal to the number of service concepts that are used for annotation, and an SDE is categorized into one of those concepts. In contrast, a number of neural networks are a number of service concepts, and each network is applied for binary classification. That is, given a neural network for concept *C,* the learning algorithm is able to define whether an SDE belongs to *C*.

Learning process in the multi-label classification is similar to the single-label approach. A neural network is created for each service concept and separately trained. Service terms are fed to input nodes in each neural network, and the input values are calculated by considering terms and their synonyms in service concepts. Given service terms of an SDE *S,* an output from each neural network shows if *S* is classified to a service concept *C* or not. After that, all service concepts that *S* belongs to, are combined together and the result is then returned as the output of the multi-label classification.

2) Machine Learning-based Multi-Label Classifier

To classify an SDE into multiple concepts, machine learning algorithms; K-Nearest Neighbor (KNN), Classification Tree (CT), and Support Vector Machine (SVM), are applied to create a multi-label classifier. While KNN and CT algorithm are multiclass classification methods, SVM algorithm is a binary classification method. Similar to the NN-based multi-label classifier, the Machine Learning (ML)-based multi-label classifier consists of n ML-based sub-classifiers; where n is the number of service concepts. One sub-classifier is for recognizing one service concept. That is, a sub-classifier $Classifier_i$ recognizes whether an SDE belongs to a service concept C_i. Then, the outputs from all sub-classifiers are combined as a list of service concepts that the SDE belongs to.

The input of each ML-based sub-classifier is an input-term vector with n dimensions; where n is the number of concept terms extracted from all actual service concepts. The values of each dimension of the input-term vector are defined as the same as those in the ML-based single-label classifier. While the output domain of the ML-based single-label classifier is a set of all service concepts, the output domain of a sub-classifier *Classifier*$_i$ is a set of two classes; a service concept C_i and *not Class*$_i$. This is a reason why the binary classification method like SVM can be used for creating the multi-label classifier, while it needs to use only the multiclass classification methods for creating the single-label classifier.

Data Set Preparation for the Classification-based Annotation Approach

As mentioned previously, one way for annotating services is to classify the services into classes. If a service *S* belongs to a class *C, S* will be annotated to *C.* To solve the classification problem, supervised learning algorithms in the neural network and the machine learning are applied. Those algorithms need a set of correct example data, called training data, for teaching a machine to correctly recognize data pattern. The example data in the training set consists of the input data and the expected target. After the training process is completed, the machine is ready for processing the new input data. In this section, the training data preparation for classifying service metadata into service concepts is described as follows.

To prepare the training data, the information from the transport ontology, which is provided by (Dong, Hussain & Chang 2011), is accessed. The ontology consists of service concepts that relate to transport services. An SDE *S* is the information of a service *S.* It provides a provider name, a service category, a service description, an address, contact detail, and the URI links which connect a service to concepts. Those links are defined by considering the service categories which are assigned by the service providers. Some SDEs from the service dataset are chosen and their URI links are applied to create the training data.

1) Training data for a single-label classification

For each chosen SDE, the input data is a set of service terms which are extracted from its service provider and the service description. In this case, only the service terms that are in a set of concept terms are focused on. The concept terms are extracted from descriptions of all service concepts that are used for annotation.

The target is a service concept which is linked by that SDE. Normally, given the ontology, an SDE can connect to service concepts. With this situation, we need to randomise only one service concept from all relevant service concepts to create the training set for a single-label classification approach.

Regarding the training data, the input data and the target data of a single-label classification approach are shown in Figure 5.17. There exist *n* SDEs, *k* service concepts, and *m* concept terms. Regarding the input data, an SDE consists of a term vector which contains the number of concept term (*Ct*) occurrence in that SDE. Similar to the input data, an SDE is comprised of a concept vector that shows which concept the SDE belongs to.

For example, a service provider is "Abel Rent A Car", and its description is"Rent New Cars From $ 29* A Day, Trucks From $ 66* Mates Rates". The target service concepts are "Car Rental" and "Truck Rental". To create the training data, the input data and the target data are defined. The input data is a set of services terms. Regarding this example, the service terms are "car", "29", "day", "truck", "66", "mate", "rate" and "rent". A set of concept terms is {car, day, truck, rent, …, agent}, and it does not contain the words "29", "66", "mate" and "rate". Because it is a single-label classification, only one service concept is randomly picked up.

In this example, "Truck Rental" is chosen to be the target of the training data. Moreover, Ct_1 and Ct_2 are assumed as "car" and "day" respectively. The service concept "Car Rental" and "Truck Rental" are defined as C_1 and C_2 respectively.

Given the transportation ontology, there exist 261 service concepts which are used for annotation services. There are 258 concept terms which are extracted from those service concepts. Thus, given an example of the training data, its input data is a vector of term occurrence for all 258 concept terms, and its target data is represented by a vector of the annotated concept for all 261 service concepts.

	SDE_1	SDE_2	…	SDE_1
Ct_1	2			
Ct_2	1			
⋮				
Ct_m	0			

(a) The input data

	SDE_1	SDE_2	…	SDE_n
C_1	0			
C_2	1			
⋮				
C_k	0			

(b) The target data

Figure 5.17 An example of the training data for a single-label service classifier.

2) Training data for a multi-label classification

For each chosen SDE, the input data is a set of service terms which are extracted from its service provider and the service description. The target is a set of service concepts which are linked by that SDE. Regarding the structure of the training data for a multi-label annotation approach, its input data is the same as one in a single-label approach. However, the target data for a multi-label classification is different. In this case, the target data consists of 261 concept vectors, and each vector represents the belonging values of 2 classes; namely *class 1* and *class 2*. Given a vector *V* of a service concept *C,* if the value of *class 1* is 1, it means a service is classified into the service concept *C*. The structure of the training data of a multi-label classification approach is shown in Figure 5.18. Similar to a single-label classification, there are *n* SDEs, *k* service concepts, and *m* concepts. The input data consists of *n* term vectors of each SDE, and the vector represents the term occurrences of each SDE. However, the target data is separated into *k* service concepts. For each concept, an SDE contains a concept vector which consists of 2 classes such as C_1 and C_2. Given an SDE *S* and a service concept *C,* if *S* is classified to *C,* the value of C_1 in the concept vector will be 1. Otherwise, if *S* is not classified to *C,* the value of C_2 in the vector will be 1.

	SDE_1	SDE_2	…	SDE_n
Ct_1	2			
Ct_2	1			
⋮	⋮			
Ct_m	0			

(a) The Input data

	SDE_1	SDE_2	…	SDE_n
C_1	0			
C_2	1			

(b) The target data for the concept “Bus Rental”

	SDE_1	SDE_2	…	SDE_n
C_1	1			
C_2	0			

(c) The target data for the concept “Truck Rental”

Figure 5.18 An example of the training data for a multi-label classifier.

For example, a service provider is “Abel Rent A Car”, and its description is “Rent New Cars From \$ 29* A Day, Trucks From \$ 66* Mates Rates”. The target service concepts are “Car Rental” and “Truck Rental”. The input data is created from services terms that appear in the concept terms. In this example, the service terms consist of the word “car”, “29”, “day”, “truck”, “66”, “mate”, “rate” and “rent”. A set of concept terms is assumed as {car, day, truck, rent, …, agent}, and it does not contain the words “29”, “66”, “mate” and “rate”. Moreover, Ct_1 and Ct_2 are assumed as “car” and “day” respectively. The service concept “Car Rental” and “Truck Rental” are defined as C_1 and C_2 respectively.

5.5 Service-Concept Connection

The service-concept connection module uses the relevance score from the previous service-concept module to annotate an SDE. The SDE is linked to relevant concepts in the service ontology by using URI. As mentioned previously, the purposes of services may be either specific or general. If the purpose of the service is specific, it should refer to only one service concept. In contrast, if the purpose is more general, the service may relate to many service concepts.

In this work, the connection between a service and its relevant concepts are categorized into three types, namely, single-label, multi-label, and combine-label connection type. The single-label based connection type chooses only the most relevant service concept for annotating an SDE, while the multi-label based type annotates multiple relevant concepts. Unfortunately, it is possible that no relevant concept is returned from the multi-label based annotation type because the relevance scores are less than the threshold. In this case, the combine-label based approach is applied for combining results from both single-label and multi-label based approaches. The single-label based, the multi-label based, and the combine-label based connection type for each proposed service annotation approach are described in Section 5.5.1 , 5.5.2 and 5.5.3 respectively.

5.5.1 Single-Label Connection Type

The service-concept connection module with the single-label connection type links an SDE to only the most relevant service concept. As mentioned in the introduction part, three service annotation approaches; such as ECBR, vector-based and classification-based approach, are proposed. To discover the most relevant service concept, the ECBR and the vector-based approach apply the annotation threshold *(AT)* and the relevance scores between an SDE and every actual service concept in the service ontology in order to select the most relevant concept. That is, by comparing with other service concepts, the service concept *C* is selected, if *C* gives the highest relevance score and that score is greater than *AT*. In contrast, the classification-based annotation approach gets the most relevant concept from the single-label service classifier.

5.5.2 Multi-Label Connection Type

The service-concept connection module with the multi-label connection type is able to link an SDE to many relevant service concepts. Similar to the single-label connection type, the ECBR and the vector-based annotation approach apply the relevance scores from the service-concept matching module and a defined annotation threshold *(AT)* to find relevant concepts. That is, all service concepts with the relevance score that is greater than *AT* are selected to be the annotation. On the other hand, the classification-based annotation approach gets the relevant concepts from the multi-label service classifier.

5.5.3 Combine-Label Connection Type

Although a multi-label based approach is designed for semantically annotating services to relevant service concepts, it is possible that an SDE may not be annotated to any service concept. In the case of the ECBR and the vector-based annotation approach, this is because no service concept gives the relevance score being greater than *AT*. In the case of the classification-based approach, this is because an SDE is classified to non-belonging class in all neural networks. As a result, many services cannot be annotated, and it makes the performance of annotation worse. To deal with this problem, a combine-label based approach is proposed.

The combine-label connection type is a hybrid way of the single-label and multi-label connection type. In the case that an SDE has no relevant concept by using the multi-label connection type, the module applies the single-label connection for finding at least one related concept. This enables a service annotation system to get more results and it may improve the performance of the system.

Given an SDE *S,* if there is no relevant service concept by using the multi-label connection type, the combine-label connection in the ECBR and the vector-based annotation approach returns a service concept with the highest relevance score, although that value is less than *AT*. In contrast, in the case of the classification-based approach, if there is no result from the multi-label classifier, the relevant concept from the single-label classifier will be returned. Otherwise, the result from the multi-label classifier will be returned instead.

5.6 Experiments

This section presents the experimental results of the proposed service annotation approaches; the ECBR, the vector-based and the classification-based approach. Based on types of service-concept connection, the results of each approach are grouped into three sub-groups such as single-label, multi-label and combine-label annotation. To evaluate the performance of the service annotation approaches, the average precision, average recall, average f-measure, average fallout rate, annotation rate (percentage of hitting service samples), and all-measures combination are applied. The value of the all-measures combination is calculated as follows:

$$All\_measures = (0.4 * avg\_precision) + (0.15 * avg\_recall) + (0.15 * avg\_fmeasure) + (0.15 * (1 - avg\_fallout)) + (0.15 * annotation\_rate)$$

5.6.1 ECBR Based Approach

To evaluate the ECBR approach for single-label, multi-label and combine-label semantic service annotation, performance is tested by setting the Annotation Threshold *(AT)* from 0.05 to 0.5 with an increment of 0.05. That is, given an SDE *S,* a service concept or multiple service concepts can be annotated to *S* if its or their relevance scores are higher than *AT*. The relevance scores between *S* and a service concept are in the range of 0 and 1. The score is 0 if *S* is not relevant to the concept, while the score is 1 if *S* is totally relevant to the concept. Although *AT* values can be in the range of 0 and 1, *AT* in this thesis are less than 0.5 because the ECBR approach cannot annotate any SDE when *AT* is 0.5.

Single-label annotation

Table 5.1 The experiments of ECBR approach for a single-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.05	31.53%	17.45%	21.53%	0.16%	79.64%	45.38%
0.1	31.48%	16.40%	20.71%	0.16%	48.87%	40.47%
0.15	48.91%	26.09%	32.64%	0.12%	20.81%	46.48%
0.2	56.10%	32.32%	39.02%	0.10%	9.28%	49.52%
0.25	75.00%	62.50%	66.67%	0.06%	0.90%	64.50%
0.3	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
0.35	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
0.4	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
0.45	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
0.5	-	-	-	-	-	-

The results from Table 5.1 show that the increment of *AT* tends to increase the precision and recall values of the ECBR approach from 31.53% in precision and 17.45% in recall to 100% in both precision and recall, with a setting for *AT* as 0.05

and 0.45, respectively. By contrast, the number of annotated samples declined from 79.64% to 0.23%. Moreover, the ECBR approach for single-label annotation cannot annotate any SDE when *AT* is 0.5.

Multi-label annotation

The results from Table 5.2 demonstrated that the precision value and the recall value tended to be higher when the *AT* increased. The results are from 11.68% in precision and 33.88% in recall to 100% in both precision and recall with a setting for *AT* as 0.05 and 0.45 respectively. The annotation rate decreased from 79.64% to 0.23%. That is, given 442 testing SDEs, the approach could not annotate 90 and 441 SDEs with a setting of *AT* as 0.05 and 0.45 respectively.

From the experiments in Table 5.2, a setting of *AT* as 0.05 returned the worst result. Although the approach was able to annotate a majority of the testing SDEs, the correctness of annotation was too low, around 10%. This was because the approach annotated an SDE to a service concept even though the relevance value between the SDE and the concept was low. This resulted in the approach to annotate a lot of irrelevant service concepts. Although a setting of *AT* as 0.3 gave the best result with 100% in precision and recall, the annotation rate was too low which was less than 1%. Similar to the result in the ECBR single-label approach, this approach cannot annotate any SDE when *AT* is set as 0.5.

Table 5.2 The experiments of ECBR approach for a multi-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.05	11.68%	33.88%	14.07%	1.68%	79.64%	38.56%
0.1	19.30%	30.44%	19.07%	0.84%	48.87%	37.35%
0.15	45.94%	36.96%	38.66%	0.28%	20.81%	47.80%
0.2	54.88%	32.32%	38.21%	0.11%	9.28%	48.91%
0.25	75.00%	62.50%	66.67%	0.06%	0.90%	64.50%
0.3	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
0.35	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
0.4	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
0.45	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
0.5	-	-	-	-	-	-

Combine-label annotation

Table 5.3 shows the results of ECBR approach for combine-label service annotation. In the case that the result of the ECBR based multi-label approach is empty, this approach returns a service concept with the highest relevance score, even though the score is less than *AT*. The increment of *AT* tended to increase the precision but decrease the recall values of the ECBR approach from 10.12% in precision and 28.17% in recall to 26.53% in precision and 14.59% in recall, with a setting of *AT* as 0.05 and 0.25 respectively. Then, the values of precision and recall were not changed. By comparing with other ECBR annotation approaches, although the precision and recall results of this combine-label based approach were worse than both single-label based and multi-label based approach, the annotation rate was better and stable, with the value 96.38%. That is, this approach was able to annotate 426 from 442 SDE samples.

Table 5.3 The experiments of ECBR approach for a combine-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.05	10.12%	28.17%	11.88%	1.43%	96.38%	39.30%
0.1	20.35%	21.71%	17.21%	0.51%	96.38%	43.36%
0.15	25.88%	16.94%	19.34%	0.20%	96.38%	45.22%
0.2	26.41%	14.59%	17.97%	0.17%	96.38%	44.88%
0.25	26.53%	14.59%	18.04%	0.17%	96.38%	44.94%
0.3	26.53%	14.59%	18.04%	0.17%	96.38%	44.94%
0.35	26.53%	14.59%	18.04%	0.17%	96.38%	44.94%
0.4	26.53%	14.59%	18.04%	0.17%	96.38%	44.94%
0.45	26.53%	14.59%	18.04%	0.17%	96.38%	44.94%
0.5	26.53%	14.59%	18.04%	0.17%	96.38%	44.94%

5.6.2 Vector-Based Approach

To evaluate the vector-based approach for single-label, multi-label and combine-label semantic service annotation, performance is tested by setting the Annotation Threshold *(AT)* from 0.1 to 0.9 with an increment of 0.1. That is, given an SDE *S,* a service concept or multiple service concepts can be annotated to *S* if its/their vector-based relevance scores are greater than *AT*. Moreover, the vector-based approaches are divided into two groups; VSM-based and EVSM-based approach so the results are presented separately based on type of vector-based approach.

Single-label annotation

1) VSM-based Annotation Approach

The experimental results of the VSM-based single-label annotation approach are shown in Table 5.4. The increment of *AT* tended to increase the precision and recall but decrease the annotation rate from 23.04% in precision, 12.03% in recall and 92.31% in annotation rate to 60% in precision, 36.67% in recall and 3.39% in annotation rate, with a setting of *AT* as 0.1 and 0.7 respectively. Most results in precision and recall were less than 50% which were unsuitable for annotating services in the real world system. Although setting *AT* as 0.7 gave 60% in precision, the approach was able to annotate just about 3% of all services.

Table 5.4 The experiments of VSM-based approach for a single-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.1	23.04%	12.03%	15.15%	0.17%	92.31%	42.11%
0.2	21.99%	11.15%	14.27%	0.18%	86.43%	40.55%
0.3	24.46%	12.10%	15.70%	0.17%	73.08%	39.89%
0.4	33.67%	17.52%	22.40%	0.15%	44.34%	41.09%
0.5	31.00%	17.42%	21.73%	0.16%	22.62%	36.64%
0.6	26.00%	16.00%	19.33%	0.17%	11.31%	32.37%
0.7	60.00%	36.67%	44.44%	0.09%	3.39%	51.66%
0.8	0.00%	0.00%	0.00%	0.23%	0.45%	15.03%
0.9	-	-	-	-	0.00%	-

2) EVSM-based Annotation Approach

The results of the EVSM-based approach for single-label service annotation are displayed in Table 5.5. Similar to the VSM-based single-label approach, the precision and recall values tended to increase when *AT* increased. The performance in precision and recall of this approach was a bit less than the VSM-based approach, but the annotation rate increased. This is because the EVSM approach creates a vector by considering the synonyms of the service terms. However, the overall values of precision and recall were less than 50%.

Table 5.5 The experiments of EVSM-based approach for a single-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.1	21.83%	11.35%	14.24%	0.18%	96.38%	42.00%
0.2	20.75%	10.46%	13.34%	0.18%	90.50%	40.42%
0.3	23.21%	11.78%	15.03%	0.17%	76.02%	39.68%
0.4	29.29%	15.20%	19.32%	0.16%	54.07%	39.98%
0.5	28.45%	15.66%	19.66%	0.16%	26.24%	35.59%
0.6	26.00%	16.00%	19.33%	0.17%	11.31%	32.37%
0.7	60.00%	36.67%	44.44%	0.09%	3.39%	51.66%
0.8	0.00%	0.00%	0.00%	0.23%	0.68%	15.07%
0.9	-	-	-	-	0.00%	-

Multi-label annotation

1) VSM-based Annotation Approach

The results from Table 5.6 demonstrated that the precision value tended to be higher when the *AT* increased. In contrast, the recall value and the annotation rate tended to be lower. Although a setting of *AT* as 0.7 gave good performance with 60% in precision and 36.67% in recall, the VSM-based approach was able to annotate only 15 service samples (3.39%) from 442 testing samples. On the other hand, setting *AT* as 0.1 could annotate almost all the testing samples, the precision was too low to apply this approach in the real service annotation system.

Table 5.6 The experiments of the VSM-based annotation approach for a multi-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.1	8.87%	46.69%	12.72%	2.74%	92.31%	40.90%
0.2	7.87%	30.82%	10.31%	1.88%	86.43%	37.00%
0.3	13.31%	30.01%	14.98%	1.04%	73.08%	37.88%
0.4	21.11%	32.02%	21.99%	0.73%	44.34%	38.09%
0.5	29.58%	28.58%	26.94%	0.51%	22.62%	38.48%
0.6	27.50%	20.00%	22.17%	0.30%	11.31%	33.98%
0.7	60.00%	36.67%	44.44%	0.09%	3.39%	51.66%
0.8	0.00%	0.00%	0.00%	0.23%	0.45%	15.03%
0.9	-	-	-	-	0.00%	-

2) EVSM-based Annotation Approach

The results of the EVSM-based approach for multi-label service annotation are presented in Table 5.7. Similar to the performance comparison in the single-label annotation type, the EVSM based multi-label approach gave a performance which was similar to the VSM based approach, but the annotation rate in the EVSM approach were better.

Table 5.7 The experiments of the EVSM-based annotation approach for a multi-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.1	7.56%	49.65%	11.58%	3.28%	96.38%	41.17%
0.2	7.24%	35.96%	10.47%	2.31%	90.50%	38.09%
0.3	12.78%	32.42%	15.40%	1.25%	76.02%	38.50%
0.4	23.02%	29.78%	22.15%	0.71%	54.07%	40.00%
0.5	27.66%	24.43%	23.92%	0.51%	26.24%	37.18%
0.6	27.50%	20.00%	22.17%	0.31%	11.31%	33.98%
0.7	60.00%	36.67%	44.44%	0.11%	3.39%	51.66%
0.8	0.00%	0.00%	0.00%	0.23%	0.68%	15.07%
0.9	-	-	-	-	0.00%	-

Combine-label annotation

1) VSM-based Annotation Approach

The results of the VSM-based approach for combine-label service annotation are shown in Table 5.8. The increment of *AT* tended to increase the precision but decrease the recall from 8.87% in precision and 46.67% in recall to 23.04% in precision and 12.03% in recall, with a setting of *AT* as 0.1 and 0.7 respectively. The purpose of the combine-label annotation type is to increase the annotation rate. That is, if the multi-label based approach cannot annotate any SDE, a service concept with the highest relevance score will be returned. Consequently, the annotation rate were identical, although *AT* values changed.

Table 5.8 The experiments of the VSM-based annotation approach for a combine-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.1	8.87%	46.69%	12.72%	2.74%	92.31%	40.90%
0.2	9.81%	30.45%	11.44%	1.77%	92.31%	38.79%
0.3	14.21%	26.21%	14.58%	0.87%	92.31%	40.52%
0.4	17.01%	19.00%	14.95%	0.46%	92.31%	40.67%
0.5	22.69%	14.77%	16.42%	0.26%	92.31%	42.56%
0.6	23.22%	12.52%	15.49%	0.19%	92.31%	42.31%
0.7	23.04%	12.03%	15.15%	0.17%	92.31%	42.11%
0.8	23.04%	12.03%	15.15%	0.17%	92.31%	42.11%
0.9	23.04%	12.03%	15.15%	0.17%	92.31%	42.11%

2) EVSM-based Annotation Approach

The results of the EVSM-based approach for combine-label service annotation are presented in Table 5.9. Similar to the VSM-based combine-label approach, the precision values tended to increase, while the recall values tended to decrease when *AT* was higher. Moreover, for all *AT* values, the annotation rate were identical with the value 96.38%. Like the performance in single-label and multi-label annotation type, the annotation rate in the EVSM-based approach was greater than the one in the VSM-based approach.

Table 5.9 The experiments of the EVSM-based annotation approach for a combine-label service annotation

AT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Annotation Rate	All-measures
0.1	7.56%	49.65%	11.58%	3.28%	96.38%	41.17%
0.2	9.14%	35.29%	11.54%	2.17%	96.38%	39.81%
0.3	13.60%	27.62%	14.53%	1.03%	96.38%	41.07%
0.4	18.32%	19.52%	15.83%	0.49%	96.38%	42.01%
0.5	21.62%	13.73%	15.40%	0.27%	96.38%	42.43%
0.6	22.01%	11.82%	14.57%	0.19%	96.38%	42.19%
0.7	21.83%	11.35%	14.24%	0.18%	96.38%	42.00%
0.8	21.83%	11.35%	14.24%	0.18%	96.38%	42.00%
0.9	21.83%	11.35%	14.24%	0.18%	96.38%	42.00%

5.6.3 Classification Based Approach

The dataset from (Dong, Hussain & Chang 2011) was used. The data set was divided into two groups: the domain-specific service ontology, and the SDE metadata. This experiment applied the transport service ontology that consists of 261 actual service concepts. Service metadata which were crawled from the transport category of the Australian Yellow Pages website was applied. In total, 2948 samples of SDEs were found.

The Neural Network Toolbox in Matlab was applied to model a two-layer feed-forward neural network and a radial basis function network. The number of input nodes depended on the number of concept terms, which were extracted from descriptions of actual service concepts. In total, 257 input nodes existed in this experiment. The number of output nodes depended on the number of actual service concepts in the ontology. Based on the transport service ontology, 261 output nodes existed.

Supervised learning approaches were used to annotate services. Input data and target data needed to be applied to simulate the neural network. The 2948 SDE samples were divided into three sub-datasets - 2064 samples (70%) were used for training and 442 samples (15%) were used for each of validation and testing.

Single-label annotation

1) Neural Network Based Approach - Feed Forward Neural Network

To evaluate the FF single-label based approach, the performance of the FF-based method was tested by setting the number of hidden neurons from 10 to 90 with an increment of 10. The performance of the FF single-label based approach is shown in Table 5.10.

The results in Table 5.10 show that the increment of hidden neurons assisted the FF-based approach to increase the precision and recall values for service annotation. The precision and recall values of the best performance of this method were around 63% and 41% respectively, while the best performance of ECBR was about 41% in precision and 21% in recall. Moreover, the FF-based approaches were able to annotate all samples, while ECBR was only able to annotate 43.21% because the service terms of un-annotated samples did not match the concept terms of each service concept.

Table 5.10 The experiments of Feed-forward Neural Network approach for a single-label service annotation

Neurons	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
10	23.08%	19.85%	20.50%	0.17%	100.00%	45.26%
20	0.00%	0.00%	0.00%	0.23%	100.00%	29.97%
30	38.24%	26.47%	29.64%	0.14%	100.00%	53.69%
40	38.46%	27.00%	30.29%	0.14%	100.00%	53.96%
50	38.91%	29.69%	32.19%	0.14%	100.00%	54.83%
60	44.57%	29.54%	33.85%	0.13%	100.00%	57.32%
70	56.56%	37.46%	42.95%	0.10%	100.00%	64.67%
80	42.99%	28.00%	32.04%	0.13%	100.00%	56.18%
90	63.35%	41.57%	47.81%	0.08%	100.00%	68.73%

2) Neural Network Based Approach - Radial Basis Function Network

To evaluate the RBF single-label based approach, the spread value of the radial basis-based method is set from 0.1 to 0.9 with an increment of 0.1. The performance of the RBF single-label based approach is shown in Table 5.11.

Similar to the FF single-label based approach, in Table 5.11, the increment of spread value of radial basis function network tended to increase the performance of service annotation. Moreover, the RBF was able to annotate the whole testing services, and the best performance of this approach was also around 63% in precision and 41% in recall the same as the FF based approach.

The experimental results demonstrate that the performance of the NN-based approach outperforms the ECBR approach. However, it seems that the recall value of NN-based approaches, which are less than 50%, may be too low to apply to a real-world annotation task. This is because the proposed approaches are single-label classification tasks. Unfortunately, in the real world, an SDE may relate to several service concepts.

Table 5.11 The experiments of RBF approach for a single-label service annotation

Spread	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
0.1	60.41%	38.65%	44.84%	0.09%	100.00%	66.67%
0.2	61.54%	39.22%	45.60%	0.09%	100.00%	67.32%
0.3	61.31%	39.67%	45.82%	0.09%	100.00%	67.34%
0.4	61.31%	39.67%	45.82%	0.09%	100.00%	67.34%
0.5	61.31%	39.67%	45.82%	0.09%	100.00%	67.34%
0.6	62.90%	41.23%	47.47%	0.08%	100.00%	68.45%
0.7	60.63%	40.03%	45.91%	0.09%	100.00%	67.13%
0.8	53.17%	35.20%	40.26%	0.11%	100.00%	62.57%
0.9	48.64%	32.52%	37.07%	0.12%	100.00%	59.88%

3) Machine Learning Based Approach - K-Nearest Neighbor Algorithm

The performance of the KNN-based approach for single-label annotation is presented in Table 5.12. The precision and recall values were in the range of 50-60% and 30-37% respectively, which were much better than the ECBR single-label annotation. By comparing with the Neural Network single-label based approaches; both FF-based and RBF-based methods, the NN approaches were a little better in both precision and recall. The best performance of the KNN single-label based approach returned approximately 4% less in precision and 5% less in recall than the FF single-based approaches. Similar to the FF single-label based approaches, although the annotation rate of the KNN approach was 100% and the precision value was above 50%, the recall value needed to be improved.

Table 5.12 The experiments of KNN-based approach for a single-label service annotation

K	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
1	57.69%	36.22%	42.22%	0.10%	100.00%	64.83%
2	50.23%	29.86%	35.52%	0.11%	100.00%	59.88%
3	59.05%	36.35%	42.73%	0.09%	100.00%	65.47%
4	57.92%	35.97%	42.16%	0.10%	100.00%	64.87%
5	53.85%	34.56%	40.09%	0.10%	100.00%	62.72%
6	58.60%	36.16%	42.50%	0.09%	100.00%	65.22%
7	58.37%	36.71%	42.91%	0.09%	100.00%	65.28%
8	56.33%	34.95%	41.08%	0.10%	100.00%	63.92%
9	54.75%	33.37%	39.46%	0.10%	100.00%	62.81%
10	55.66%	34.28%	40.36%	0.10%	100.00%	63.44%

4) Machine Learning Based Approach - Classification Tree Algorithm

The performance of the CT single-label based annotation is shown in Table 5.13. The results show that this approach was able to annotate all testing services, and its precision was good; however, its recall value was below 40% and needed to be increased.

Table 5.13 The experiments of Classification Tree based approach for a single-label service annotation

Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
60.18%	39.29%	45.26%	0.09%	100.00%	66.74%

By comparing with other single-label based approaches, the performance of the CT-based method was a little better than the KNN method, especially the recall value, while both FF and RBF based method outperformed the CT-based method. Similar to other Neural Network and Machine Learning based approaches, the CT approach performed much better than ECBR approach in both precision and annotation rate.

Multi-label annotation

Given an SDE *S,* multiple service concepts are annotated to *S* if *S* is classified or relevant to those concepts. The multi-label based annotation approaches are evaluated by considering the precision, recall, and annotation rate. The results of the ECBR approach are compared with those of the Neural Network based approaches.

1) Neural Network Based Approach - Feed Forward Neural Network

The performance of the Feed-Forward Neural Network based approach for multi-label service annotation is shown in Table 5.14. To do the test, the number of hidden neurons is set from 10 to 90 with an increment of 10. The precision values of all cases are high, which are more than 70%. A setting of the number of neurons as 10 returns the best precision with 90.81%, while the worst 72.03% in precision comes with a setting of neurons as 30. Moreover, in all cases of neurons settings, the recall values are also quite good which are in the range of 66% to 82%. That is, given an SDE *S,* the approach is able to correctly annotate *S* to more than 60% of all relevant service concepts.

Table 5.14 The experiments of Feed-forward Neural Network approach for a multi-label service annotation

Neurons	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
10	90.81%	81.36%	83.39%	0.03%	28.73%	80.34%
20	73.97%	67.14%	67.99%	0.08%	59.50%	73.77%
30	72.03%	66.11%	66.55%	0.10%	60.86%	72.82%
40	74.77%	69.12%	69.94%	0.10%	48.42%	73.01%
50	78.76%	73.61%	73.34%	0.07%	54.30%	76.68%
60	81.60%	73.97%	75.08%	0.08%	60.63%	79.08%
70	74.77%	70.16%	69.51%	0.10%	57.69%	74.50%
80	75.89%	72.27%	71.24%	0.10%	60.18%	75.90%
90	77.02%	67.84%	69.66%	0.08%	56.11%	74.84%

Unfortunately, this approach is unable to annotate a lot of SDEs; around 40-70%. Although the performance of this approach reaches about 90% in precision, it is able to annotate only about 30% of the whole testing SDEs. However, when the performance of FF neural network approach is compared with the performance of ECBR approach, the FF-based approach is still much better than the ECBR one. With the annotation rate in the range of 45% to 50%, the FF-based approach returns 74.77% in precision and 69.12% in recall, while the ECBR approach returns about 33% in precision and 47% in recall.

2) Neural Network Based Approach - Radial Basis Function Network

To test the performance of the RBF-based approach, the spread value of the network is set from 0.1 to 0.9 with an increment of 0.1. The performance of this approach is displayed in Table 5.15. Similar to the performance of the FF-based approach, the precision and recall values are quite high, which are in the range of 65-95% and 57-83% respectively. In details, the maximum precision value of this approach (93.84%) is a little better than the one of the FF-based approach (90.81%), while the minimum precision value of this approach (66.29%) is quite a lot lower. Like the precision value, this approach returns a little better maximum recall value (82.58%) than the FF-based approach.

The annotation rate of this approach reaches 85.97%, while the maximum value in the FF-based approach is just 60.86%. By considering the best case of both approaches, the RBF-based approach with a setting of spread as 0.8 performs

93.84% in precision, 81.79% in recall, and 33.03% in annotation rate while the FF-based approach with a setting of neurons as 10 performs 90.81% in precision, 81.36% in recall, and 28.73% in annotation rate. This shows that the performance of the RBF-based approach is a little better that the FF-based approach in precision and the annotation rate.

Table 5.15 The experiments of RBF approach for a multi-label service annotation

Spread	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
0.1	66.29%	58.60%	60.71%	0.11%	85.97%	72.29%
0.2	66.78%	58.51%	60.90%	0.11%	85.97%	72.50%
0.3	67.22%	58.93%	61.32%	0.10%	85.75%	72.77%
0.4	70.75%	60.05%	63.18%	0.09%	81.22%	73.96%
0.5	77.05%	66.42%	69.69%	0.07%	69.00%	76.58%
0.6	84.16%	73.91%	76.87%	0.04%	50.00%	78.78%
0.7	91.11%	79.49%	82.87%	0.03%	40.72%	81.90%
0.8	93.84%	81.79%	85.14%	0.02%	33.03%	82.53%
0.9	92.62%	82.58%	85.14%	0.02%	27.60%	81.34%

3) Machine Learning Based Approach - K-Nearest Neighbor Algorithm

To evaluate the KNN-based annotation approach, the number of *K* is set from 1 to 10 with an increment of 1. The performance of this approach is presented in Table 5.16. The precision values of all settings of *K* are in the range of 50-60%, while the recall values are in the range of 45-57%. The best case, which returns the maximum precision value, is the performance of a setting of *K* as 1. It gives 59.63% in precision, 56.20% in recall, and 100% in annotation rate. It is interesting that the annotation rates of all setting cases of *K* are high, more than 95%, and the approach still keeps around 50% in precision and recall respectively.

The result shows that the KNN based approach absolutely outperforms the ECBR approach. Although the worst case of this approach performs 51.23% in precision, 56.30% in recall, and 100% in annotation rate, it is still much better than the best case of ECBR approach with a setting of *AT* as 0.9. By comparing with the Neural Network based approaches (FF-based and RBF-based approach), the overall precision and recall values in this KNN-based approach are much lower, while, on the other hand, the overall annotation rates are much higher.

Table 5.16 The experiments of KNN-based approach for a multi-label service annotation

K	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
1	59.63%	56.20%	55.96%	0.16%	100.00%	70.65%
2	51.23%	56.30%	50.98%	0.24%	100.00%	66.55%
3	57.01%	50.40%	51.98%	0.13%	100.00%	68.14%
4	54.98%	52.64%	51.98%	0.16%	99.10%	67.53%
5	55.94%	49.62%	51.27%	0.14%	95.25%	66.78%
6	54.83%	49.24%	50.42%	0.14%	99.10%	66.73%
7	54.80%	48.17%	50.10%	0.14%	96.61%	66.13%
8	54.40%	47.84%	49.75%	0.15%	97.74%	66.04%
9	56.22%	49.44%	51.42%	0.13%	94.57%	66.78%
10	53.15%	47.59%	48.86%	0.15%	96.83%	65.23%

4) Machine Learning Based Approach - Classification Tree Algorithm

The performance of the Classification Tree based annotation approach is shown in Table 5.17. Its results are quite good with 69.06%, 65.61% and 87.33% in precision, recall, and annotation rate respectively. By comparing with the KNN-based approach, the classification tree based approach performs better in precision and recall, but its annotation rate is around 8-13% less than the KNN-based approach.

Similar to the KNN-based approach, the classification tree based approach much outperforms the ECBR based approach with all performance measures, and it performs better than the Neural Network based approach in the annotation rate measure. By considering the precision and recall values, the KNN-based approach is not better than the Neural Network based approach. However, by considering the case having the close annotation rate, this approach performs better than the RBF-based approach with a setting of Spread as 0.1.

Table 5.17 The experiments of Classification Tree based approach for a multi-label service annotation

Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
69.06%	65.61%	64.54%	0.14%	87.33%	75.23%

5) Machine Learning Based Approach - Support Vector Machine

The result of multi-label annotating services by the SVM-based approach is presented in Table 5.18. Although the annotation rate reaches 100% and the recall value is more than 65%, the precision value is too low and less than 10%. It means that the support vector machine may be unsuitable for annotating services to multiple service concepts.

Table 5.18 The experiments of SVM-based approach for a multi-label service annotation

Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
6.83%	66.40%	12.09%	3.66%	100.00%	43.95%

Combine-label annotation

1) Neural Network Based Approach - Feed Forward Neural Network

The performance of the FF combine-label based annotation approach is shown in Table 5.19. It can be clearly seen that the annotation rates of all cases are 100%. That is, all testing SDEs are able to be annotated. Because the combine-label approach aims to increase the annotation rate by combining the results of the single-label and the multi-label approach all together, the performance of this approach is decreased when comparing with the FF multi-label based approach from Table 5.14. However, by comparing with the FF single-label approach from Table 5.10, this approach provides a little better performance.

The precision and recall values of this approach are in the range of 32-60% and 28-51% respectively, while the FF multi-label based approach returns the precision and recall in the range of 75-91% and 65-82% respectively. This indicates that adding the results from the single-label approach into those in the multi-label approach may sharply drop the performance of the annotation system. However, the results of this approach are acceptable because some cases return around 50% in precision and recall, which is still better than the performance of ECBR approach.

Table 5.19 The experiments of Feed-forward Neural Network approach for a combine-label service

Neurons	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
10	32.20%	28.30%	29.12%	0.16%	100.00%	51.47%
20	44.01%	39.95%	40.45%	0.14%	100.00%	59.64%
30	52.88%	45.40%	46.70%	0.13%	100.00%	64.95%
40	40.95%	37.25%	37.86%	0.15%	100.00%	57.62%
50	48.88%	44.97%	45.06%	0.13%	100.00%	63.04%
60	57.40%	49.30%	50.89%	0.12%	100.00%	67.97%
70	56.71%	50.47%	51.09%	0.12%	100.00%	67.90%
80	56.30%	48.87%	49.60%	0.12%	100.00%	67.27%
90	59.28%	48.53%	51.10%	0.11%	100.00%	68.64%

2) Neural Network Based Approach - Radial Basis Function Network

The performance of the RBF combine-label based annotation approach is presented in Table 5.20. The precision and recall values are in the range of 48-63% and 36-53% respectively, which are not bad performances. By considering the precision value, the best result is from setting the Spread as 0.6, which gives 62.90% in precision and 50.28% in recall, and it outperforms the best case of the FF combine-label based approach from Table 5.19. By comparing the worst cases of both combine-label approaches, the RBF-based approach returns 48.64% and 36.76% in precision and recall respectively, while the FF-based one performs 32.30% in precision and 28.30% in recall. That is, based on the combine-label approach, the RBF-based approach outperforms the FF-based approach.

Table 5.20 The experiments of RBF approach for a combine-label service annotation

Spread	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
0.1	59.48%	51.96%	54.08%	0.12%	100.00%	69.68%
0.2	61.03%	52.45%	55.00%	0.12%	100.00%	70.51%
0.3	60.80%	52.90%	55.22%	0.12%	100.00%	70.52%
0.4	61.31%	51.66%	54.50%	0.10%	100.00%	70.43%
0.5	61.31%	51.32%	54.27%	0.10%	100.00%	70.35%
0.6	62.90%	50.28%	53.88%	0.09%	100.00%	70.77%
0.7	60.63%	48.00%	51.59%	0.09%	100.00%	69.18%
0.8	53.17%	40.84%	44.23%	0.11%	100.00%	64.01%
0.9	48.64%	36.76%	40.02%	0.12%	100.00%	60.96%

Similar to the results in the FF combine-label based approach, the RBF combine-label approach has a lesser performance than the RBF multi-label approach, and it improves a little on the performance of the RBF single-label approach.

3) Machine Learning Based Approach - K-Nearest Neighbor Algorithm

The performance of the KNN combine-label based annotation approach is shown in Table 5.21. The overall precision and recall values are in the range of 51-60% and 46-57% respectively. By comparing with the KNN multi-label based approach in Table 5.16, it seems that their results are extremely similar because the annotation rates of the KNN multi-label approach are close to 100%. The best performance, with setting K as 1, of the combine-label approach is the same as the multi-label one. The minor changes in the combine-label approach are little decrements of the precision and recall values with settings of K as 4 to 10. Like the KNN multi-label approach, the KNN combine-label approach much improves the recall values of the KNN single-label approach.

By comparing with other combine-label approaches; FF-based and RBF-based approach, the KNN based approach performs well in recall. With the best case of all approaches, the KNN provides 56.20% in recall, while the recall values of FF and RBF approach are 48.53% and 50.28% respectively. In contrast, the precision value of the KNN is close to the one of the FF-based approach, but it is a little less than the one of the RBF based approach.

Table 5.21 The experiments of KNN-based approach for a combine-label service annotation

K	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
1	59.63%	56.20%	55.96%	0.16%	100.00%	70.65%
2	51.23%	56.30%	50.98%	0.24%	100.00%	66.55%
3	57.01%	50.40%	51.98%	0.13%	100.00%	68.14%
4	54.49%	52.17%	51.51%	0.16%	100.00%	67.32%
5	53.28%	47.27%	48.84%	0.14%	100.00%	65.71%
6	54.34%	48.79%	49.97%	0.14%	100.00%	66.53%
7	52.94%	46.53%	48.40%	0.14%	100.00%	65.39%
8	53.85%	47.21%	49.15%	0.15%	100.00%	65.97%
9	54.98%	47.55%	49.67%	0.13%	100.00%	66.55%
10	53.28%	46.87%	48.36%	0.15%	100.00%	65.57%

4) Machine Learning Based Approach - Classification Tree Algorithm

The performance of the classification tree combine-label based approach is presented in Table 5.22. The precision and recall value of this approach are around 63% and 59%, which are better than the best performance in the KNN combine-label based approach, which is approximately 60% in precision and 56% in recall.

By comparing with the classification tree multi-label approach, the precision and recall value of this approach decrease around 6%. In contrast, this approach improves the performance of the classification tree single-label approach around 2% in precision and 20% in recall.

Regarding the Neural Network combine-label approaches, the classification tree approach performs better than both FF and RBF based approaches in recall. The recall value of the classification tree approach is about 59% while the recall of the FF and RBF based approaches are 48.53% and 50.28% respectively, and all approaches are able to annotate the whole testing SDE metadata. The precision value of the classification tree approach is close to the one of the RBF approach, which is greater than the one of the FF-based approach by around 3%.

Table 5.22 The experiments of Classification Tree based approach for a combine-label service annotation

Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
62.58%	59.18%	58.37%	0.14%	100.00%	72.64%

5.7 Experiment Summary

Based on the experimental results in Section 5.6 , the best results of the proposed semantic based approaches for single-label, multi-label and combine-label service annotation are presented in Table 5.23, Table 5.24 and Table 5.25 respectively. Based on the experimental results, it can be summarized that:

1) Based on the results in all-measures, the proposed ECBR approach outperforms other approaches in single-label and multi-label service annotation. However, the ECBR approach performed 85.03% in all-

measures, but the annotation rate was only 0.23% which was too low to apply it in the real world situation. That is, only one service from 442 testing services was annotated and there was only one concept that was relevant to that service. As a result, the ECBR approach performed 100% in precision, recall and f-measure. Unfortunately, the ECBR approach may not perform well in other testing datasets. In contrast, the FF-based and RBF-based approaches also performed well in both single-label and multi-label annotation. Especially the multi-label annotation, the FF-based and RBF-based approaches performed 80-83% in all-measures, more than 80% in precision, recall and f-measure, and 28-33% in annotation rate which was much better than the annotation rate in the ECBR. In this thesis, the precision value has the most priority; as a result, the all-measures value in the ECBR was a little better than the FF-based and RBF-based approaches. In the future, different service datasets are used for testing the proposed approaches to find approaches that perform well in overall datasets.

2) Based on the results in all-measures, the proposed CT-based approach outperforms other approaches in combine-label service annotation. Overall, the performance of the classification-based approaches (FF, RBF, KNN and CT) is much better than the ECBR and the vector-based approaches (VSM and EVSM) in all performance measures. Although the CT-based approach gave the best all-measures value for combine-label annotation, the FF-based and RBF-based approaches performed better in single-label and multi-label annotation. As mentioned previously, more service datasets should be used to find approaches that are suitable for all datasets.
3) Regarding the vector-based annotation approaches, the performance values of the VSM-based approach and the EVSM-based approach are quite close to each other. That is, using EVSM model to represent services is unable to improve the performance of VSM-based approach. However, the EVSM-based approach tends to increase the number of annotated services. Moreover, the performance values of both VSM-based and EVSM-based approaches are worse than the ECBR and the classification-based approaches.

Table 5.23 The best results of semantic based approaches for single-label service annotation

Approach	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
ECBR	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
VSM	60.00%	36.67%	44.44%	0.09%	3.39%	51.66%
EVSM	60.00%	36.67%	44.44%	0.09%	3.39%	51.66%
FF	63.35%	41.57%	47.81%	0.08%	100.00%	68.73%
RBF	62.90%	41.23%	47.47%	0.08%	100.00%	68.45%
KNN	59.05%	36.35%	42.73%	0.09%	100.00%	65.47%
CT	60.18%	39.29%	45.26%	0.09%	100.00%	66.74%

Table 5.24 The best results of semantic based approaches for multi-label service annotation

Approach	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
ECBR	100.00%	100.00%	100.00%	0.00%	0.23%	85.03%
VSM	60.00%	36.67%	44.44%	0.09%	3.39%	51.66%
EVSM	60.00%	36.67%	44.44%	0.11%	3.39%	51.66%
FF	90.81%	81.36%	83.39%	0.03%	28.73%	80.34%
RBF	93.84%	81.79%	85.14%	0.02%	33.03%	82.53%
KNN	59.63%	56.20%	55.96%	0.16%	100.00%	70.65%
CT	69.06%	65.61%	64.54%	0.14%	87.33%	75.23%
SVM	6.83%	66.40%	12.09%	3.66%	100.00%	43.95%

Table 5.25 The best results of semantic based approaches for combine-label service annotation

Approach	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Annotation Rate	All-measures
ECBR	25.88%	16.94%	19.34%	0.20%	96.38%	45.22%
VSM	22.69%	14.77%	16.42%	0.26%	92.31%	42.56%
EVSM	21.62%	13.73%	15.40%	0.27%	96.38%	42.43%
FF	57.40%	49.30%	50.89%	0.12%	100.00%	67.97%
RBF	62.90%	50.28%	53.88%	0.09%	100.00%	70.77%
KNN	59.63%	56.20%	55.96%	0.16%	100.00%	70.65%
CT	62.58%	59.18%	58.37%	0.14%	100.00%	72.64%

5.8 Conclusion

This chapter describes the proposed semantic service annotation approaches which are the first part of the semantic service retrieval system. Semantic annotation enables the system to understand the purpose of services and further assists in intelligent and precise service retrieval.

In this thesis, the purpose of a service is represented by using service concepts. An SDE is annotated to relevant service concepts in the domain-specific ontology. In this chapter, new semantic service annotation approaches; namely, the ECBR, vector-based and classification-based annotation approach, are proposed. The main workflows of those approaches are identical. That is, given an SDE and a service concept, a service description and concept descriptions are extracted into service terms and concept terms respectively. Then, the relevance score between service terms and concept terms is calculated, and the service is annotated to the concept if their relevance score is high. The difference among those approaches is how to calculate the relevance score between a service and a concept.

The ECBR approach is a term matching based approach, while the vector-based approach represents a service and a concept by using a vector and calculates the relevance score by using the cosine similarity. In contrast, the classification-based approach treats the service annotation problem as the service classification problem. They apply artificial neural networks and machine learning algorithms to classify services into service classes or service concepts. Moreover, for each annotation approach, the service-concept connections are divided into three types which are single-label, multi-label and combine-label annotation.

To evaluate the performance of the proposed semantic approaches for annotating services, precision, recall, f-measure, fallout rate, annotation rate, and all-measures combination are applied. The results show that the ECBR approach gives the best performance in single-label and multi-label annotation, while the classification-based approaches very clearly outperform both the ECBR and the vector-based approaches in combine-label annotation.

Chapter 6 Semantic Service Querying

6.1 Introduction

Apart from the semantic service annotation in Chapter 5, semantic service querying is another important part for semantically retrieving services. In this thesis, concepts in the domain-specific ontology are applied to represent the purpose of text-based service queries. Given a query *Q,* the semantic service querying approach is able to return service concepts that are relevant to *Q*. For example, the querying approach receives a query "Vessel agent consultant" and then queries a relevant service concept "Ship_Broker_Consultation" from the transport service ontology. Those relevant concepts from the querying process and the service annotations in Chapter 5 are employed for semantically retrieving services in Chapter 7.

To query the relevant concepts, a query is separated into terms and semantically expanded by the query expansion methods. Then, querying approaches apply all expanding terms to calculate the relatedness between the query and service concepts in the ontology. That is, the focus of this chapter is in two main parts, including the query expansion and the querying part. Regarding the query expansion part, WordNet-based and the ontology-based methods are proposed. While the former applies synonyms in WordNet database, the latter applies ontological terms in the service ontology to enlarge the query terms. Regarding the querying part, similar to the semantic service annotation in Chapter 5, three semantic service querying approaches; ECBR, vector-based and classification-based approach, are proposed. The main workflows of all proposed querying approaches are the same, but their techniques for computing the relevance score between the query and a service concept are different.

In the next section, the main workflow of the semantic service querying methodology is displayed. Then, each step of the methodology is explained in details in the rest of the chapter.

6.2 Semantic Service Querying Methodology

In this section, the workflow of the semantic service querying methodology is outlined in Figure 6.1. Given a query *Q,* the semantic service querying approach is able to discover a service concept or service concepts that are relevant to *Q*. The steps involved in the semantic service querying methodology are as follows:

1. Term extraction module extracts query and concept terms from the query *Q* and descriptions of every service concept in the service ontology.

2. Query expansion module expands each term in the query terms by using related words.

3. Querying module calculates relevance scores between the expanded query terms and the concept terms of each service concept. Then, the module queries service concepts that are relevant to *Q* based on the relevance score.

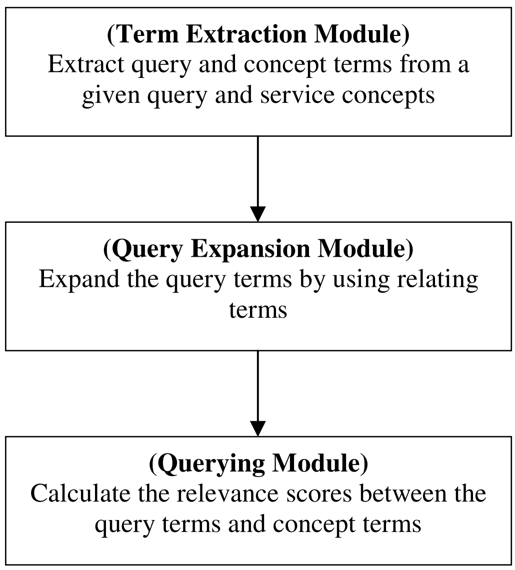


Figure 6.1 Flow Chart of the Semantic Service Querying Methodology

The proposed semantic service querying methodology consists of three sub-modules such as term extraction, query expansion, and querying module. The details of each sub-module are explained in Section 6.3 , 6.4 and 6.5 respectively. In this thesis, several semantic-based approaches are proposed for querying services. Although the main workflows of those approaches are identical, they apply different query expansion methods and calculate the relevance score by using different techniques. To expand the query terms, two query expansion methods, namely WordNet-based and ontology-based methods, are proposed. The WordNet-based query expansion method applies synonyms in WordNet, while the ontology-based approach applies ontological terms in the service concepts. The explanations of those query expansion methods are presented in Section 6.4.1 and 6.4.2 respectively.

Similar to the semantic service annotation, three service querying approaches, including the ECBR, the vector-based and the classification-based approach, are proposed and described in Section 6.5.1 , 6.5.2 and 6.5.3 respectively. In Section 6.6 , in order to improve the performance of service querying, the hybrid querying approach is proposed. The main idea is to combine the querying results by using both WordNet-based and ontology-based query expansion methods. Finally, the experimental results of the proposed semantic service querying approaches and the conclusion of this chapter are demonstrated in Section 6.7 and Section 6.8 respectively.

6.3 The Term Extraction Module

The first step of the semantic service querying methodology is to extract query terms and concept terms from a query and the service concepts in the service ontology. The query terms are sent to the query expansion module in order to expand the query terms with related terms from WordNet and the service ontology, while the concept terms are applied for calculating relevance scores between the query and every service concept in the ontology. The pseudo-code of the term extraction module is shown in Figure 6.2. Regarding the concept terms, the module has to extract the concept terms for each service concept and all service concepts. The pseudo-codes of the concept term extraction sub-module for the semantic service querying are the same as one of the semantic service annotation in Section 5.3.2 Thus, in this section, only the query term extraction sub-module is focused on and described in Section 6.3.1 .

```
Program Term_Extraction(Query Q, Ontology O)
   query_terms = Query_Term_Extraction(Q)
   concept_terms = Concept_Term_Extraction(O)
end Program
```

Figure 6.2 Pseudo-code of the term extraction module

6.3.1 Query Term Extraction

Given a query *Q,* the query term extraction sub-module extracts the query terms which are expanded and then sent to the querying module for calculating relevance scores between *Q* and every service concept. A query *Q* is a text that consists of one or several words, and it may contain uppercase letters, lowercase letter, numbers, and special characters, for example, “Ship Broker Consultation”, “cab drop off” and “limo”.

The pseudo-code of the query term extraction sub-module is presented in Figure 6.3. Basically, the module uses spaces and punctuation marks, including space, full stop, comma, colon, semicolon, hyphen, brackets, slash, question mark and exclamation mark, to separate a query *Q* into terms. Each single term is stemmed to get its general term. Then, the stopwords, for instance, “a”, “an”, “the”, “that”, are removed. Similar to the service term extraction module for semantic service annotation in Section 5.3.1 , WordNet 3.1 is applied for selecting meaningful noun words to be the query terms denoted as *QT*. An example of input and output of the query term expansion module is presented in Figure 6.4. Given a query "Vessel agent consultant", the set of query terms {vessel, agent, consultant} is provided.

```
Program Query_Term_Extraction(Query Q)
   Terms:[t1, t2, ..., tn] = split Q by using space and punctuations
   query_terms = create an empty list
   For each ti in Terms :
      tstem = stem ti
      if (tstem is not a stopword)
         search tstem in WordNet3.1
         if (tstem is in WordNet 3.1 and tstem is noun)
            add tstem into query_terms
         end if
      end if
   end for
   return query_terms
End Query_Term_Extraction
```

Figure 6.3 Pseudo-code of the query term extraction module

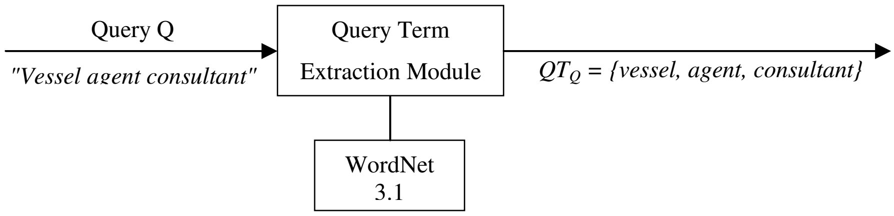


Figure 6.4 An example of input and output of the query term extraction module

6.4 Query Expansion Module

The query expansion module receives the query terms from the term extraction module and then enlarges them with their related terms. Using only terms in the query may be not enough to get the precise and relevant outputs because the relevant output may contain other words that refer to the same or similar meaning, but those words are not in the query. However, if the query is expanded with many unrelated terms, much irrelevant output will be returned and the performance of querying will be unsatisfactory. Therefore, the query system requires an efficient approach to properly enlarge the query terms.

The pseudo-code of the query expansion module is presented in Figure 6.5. Given a query term *i* (qt_i), expanding terms of qti are found by considering the semantics of the query term. The terms that have the same meaning or refer to the same topic as qt_i are added into a set of expanding query terms *QE*. The output of the query expansion module is called the Expanded Query Term (*EQT*), which is the combination of a set of query terms *(QT)* and a set of expanding terms (*QE*). To find the meaning and the topic area of any query term *qt*, a database of English words, like WordNet and a domain-specific ontology, like the transportation ontology respectively, are applied. An example of input and output of the query expansion module is shown in Figure 6.6. Synonyms of the term "agent" are "broker" and "factor", and synonyms of the term "consultant" are "advisor" and "adviser". In contrast, the term "vessel" has no synonym. That is, given a set of query terms {vessel, agent, consultant} from the term extraction module, the output of the query expansion module is {vessel, agent, consultant, broker, factor, advisor, adviser}.

```
Program Query_Expansion(QueryTerms QT:{qt1, qt2, ..., qtn})
   if (expand QT by using WordNet)
      QE:{qe1, qe2, ..., qem} = WordNet_based_query_expansion(QT)
   else if (expand QT by using Ontology)
      QE:{qe1, qe2, ..., qem} = Ontology_based_query_expansion(QT)
   end if
   copy QT to EQT
   For each qei in QE :
      add qei into EQT
   end for
   return EQT
End Query_Expansion
```

Figure 6.5 Pseudo-code of the query expansion module

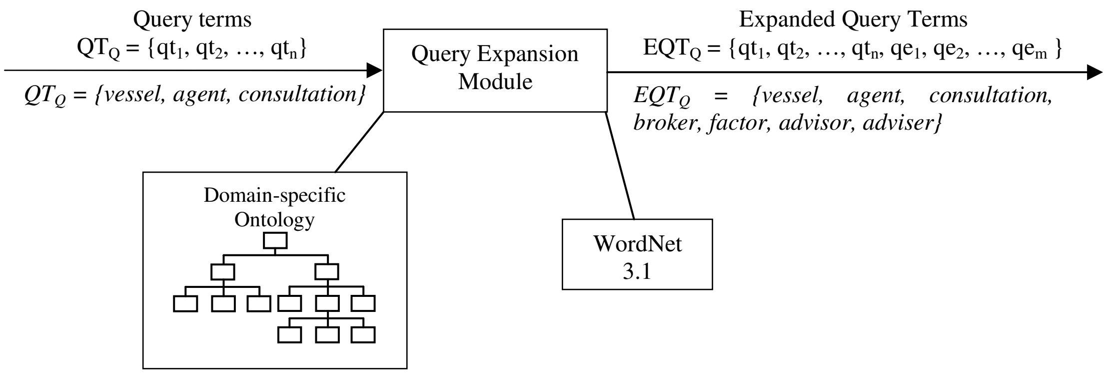


Figure 6.6 An example of input and output of the query expansion module

In this thesis, two approaches are proposed for enlarging the query terms, namely WordNet-based expansion approach and ontology-based expansion approach. The WordNet-based expansion approach uses WordNet to find expanding terms, while the ontology-based approach applies a domain-specific service ontology to find related terms of a query term. The workings of WordNet-based and ontology-based query expansion approaches are explained in Section 6.4.1 and 6.4.2 respectively.

6.4.1 WordNet-based Query Expansion Approach

A WordNet-based approach applies a large database of English words called WordNet to expand the query terms by considering the meaning of each query term. WordNet was developed by the Cognitive Science Laboratory of Princeton University. English words containing nouns, verbs, adjectives and adverbs in WordNet are categorized into sets of synonyms which are called synsets. In this thesis, those synsets in WordNet are employed to achieve synonyms or related terms of that query term for the purpose of query expansion.

As mentioned above, the query terms should be enlarged by using appropriate related terms. If the query terms are extended by using terms that have a weak relation with the query term, the performance of the querying process may be reduced. Because of this reason, two techniques for the WordNet-based approach, namely *All-senses expansion* and *Proper-sense expansion* technique, are suggested.

1) All-Senses Expansion Technique

The pseudo code of the WordNet-based approach with All-senses expansion technique is displayed in Figure 6.7. Given a query *Q,* the query expansion module receives a set of query terms (QT_Q) from the query term extraction module. Firstly, the system creates a set of expanded query terms (EQT_Q) and assigns all query terms in QT_Q to EQT_Q. For each query term qt_i in QT_Q, the system obtains synonyms from all senses of the query term qt_i and then adds those synonyms into the EQT_Q.

```
Function WordNet-based approach with All-senses technique (QT_Q : {qt_1, qt_2,
…, qt_n})
EQT_Q = QT_Q
For each qt_i in QT_Q
      Syns = Lookup synonyms from all senses of qt_i from WordNet database
      For each synonym in Syns
         Add synonym in EQT_Q
      End for
End for
Return EQT_Q
End WordNet-based approach with All-senses technique
```

Figure 6.7 Pseudo-code of the WordNet-based approach with All-senses expansion technique

For example, *QT* of a query "Vessel agent consultant" is assigned as {vessel, agent, consultant}. For each term in *QT*, its synonyms of all senses are returned from WordNet. For example, the noun term "vessel" comprises 3 senses of meaning; 1) a tube in which a body fluid circulates, 2) a craft designed for water transportation, and 3) an object used as a container (especially for liquids). While a synonym of the first sense of meaning is "vas", one of the second sense is "water craft" and the third sense has no synonym. Therefore, synonyms of the term "vessel" are "vas" and "water craft". Using the same process, synonyms of "agent" are "factor" and "broker", while those of "consultant" are “advisor” and “adviser”. Consequently, the output which is returned from the query expansion module is {vessel, agent, consultant, vas, watercraft, factor, broker, advisor, adviser}.

2) Proper-Sense Expansion Technique

The pseudo code of the WordNet-based approach with Proper-sense expansion technique is displayed in Figure 6.8. Like the All-senses expansion technique, firstly, *EQT* is created and contains all query terms in *QT*, which is sent from the query term extraction module. For each query term qt_i, the system then generates pairs of query terms which consist of qt_i and others terms in *QT*. That is, assuming *QT* is $\{qt_1, qt_2, \ldots, qt_n\}$, a set of pairs of query term qt_1 is $\{(qt_1, qt_2), (qt_1, qt_3), \ldots, (qt_1, qt_n)\}$. For each pair ($qt_i$, qt_j), the semantic similarity between each sense of $\mathrm{qt_i}$ and $\mathrm{qt_j}$ is calculated by using an existing semantic word similarity method. In this thesis, LCH word similarity method (Leacock & Chodorow 1998) is applied. It calculates the similarity by considering the length of the shortest path between two words. Then, the module selects a pair of senses that gives the maximum similarity value. Assumed that a pair ($qt_{i\text{-}sense\_A}$, $qt_{j\text{-}sense\_B}$) is selected, it means that the relatedness value between term qt_i with sense *A* and term qt_j with sense *B* is greater than other pairs of senses of (qt_i, qt_j). To expand the query terms, qt_i is extended by using its synonyms in sense *A*, while qt_j is enlarged by using its synonyms in sense *B*.

For instance, the same as the example in the previous section, *QT* is {vessel, agent, consultant}. Based on the WordNet 3.1, the number of senses of the word "vessel", "agent" and "consultant" is 3, 6 and 1 respectively. In this case, the module has to calculate similarity values of pairs of senses; ($\mathrm{vessel_{sense\_1}}$, $\mathrm{agent_{sense\_1}}$), ($\mathrm{vessel_{sense\_1}}$, $\mathrm{agent_{sense\_2}}$), ..., ($\mathrm{vessel_{sense\_1}}$, $\mathrm{agent_{sense\_6}}$), ($\mathrm{vessel_{sense\_2}}$, $\mathrm{agent_{sense\_1}}$), ..., ($\mathrm{vessel_{sense\_2}}$, $\mathrm{agent_{sense\_6}}$), ($\mathrm{vessel_{sense\_3}}$, $\mathrm{agent_{sense\_1}}$), ..., ($\mathrm{vessel_{sense\_3}}$, $\mathrm{agent_{sense\_6}}$), ($\mathrm{vessel_{sense\_1}}$, $\mathrm{consultant_{sense\_1}}$), ..., ($\mathrm{vessel_{sense\_6}}$, $\mathrm{consultant_{sense\_1}}$), ($\mathrm{agent_{sense\_1}}$, $\mathrm{consultant_{sense\_1}}$), ..., ($\mathrm{agent_{sense\_6}}$, $\mathrm{consultant_{sense\_6}}$), by using an existing LCH word similarity method. In this example, ($\mathrm{vessel_{sense\_1}}$, $\mathrm{agent_{sense\_1}}$), ($\mathrm{vessel_{sense\_3}}$, $\mathrm{consultant_{sense\_1}}$) and ($\mathrm{agent_{sense\_1}}$, $\mathrm{consultant_{sense\_1}}$) are selected. Based on the database in WordNet 3.1, only $\mathrm{vessel_{sense\_1}}$ and $\mathrm{consultant_{sense\_1}}$ have synonyms, including "vas", "advisor" and "adviser". Therefore, after enlarging the query terms *QT*, the *EQT* is {vessel, agent, consultant, vas, advisor, adviser}.

```
Function WordNet-based approach with Proper-senses technique
 (QT_Q : {qt_1, qt_2, …, qt_n})
   EQT_Q = QT_Q
   For each qt_i in QT_Q
      Bigram_qt_i = {(qt_i, qt_i+1), (qt_i, qt_i+2), …, (qt_i, qt_n)}
         For each (qt_i, qt_j) in Bigram_qt_i        // i ≠ j
            For each sense of meaning of qt_i
               For each sense of meaning of qt_j
                  Similarity = calculate similarity between sense-qt_i and 
sense-qt_j
                  Add Similarity to Similarity List
                  Add a pair of senses to Pair-Senses List
               End For
            End For
            Max_sim = get maximum similarity from Similarity List
            Selected_pair = get a pair-sense with the maximum similarity
                            from Pair-Senses List
            Proper_sense = a pair of senses of (qt_i, qt_j) in the 
Selected_pair
            Syns = Lookup synonyms from Proper_sense of qt_i and qt_j
                      from WordNet database
            For each synonym in Syns
               Add synonym in EQT_Q
            End for
         End for
   End for
   Return EQT_Q
End WordNet-based approach with Proper-senses technique
```

Figure 6.8 Pseudo-code of the WordNet-based approach with Proper-sense expansion technique

6.4.2 Ontology-based Query Expansion Approach

Although expanding the query by using its synonyms sounds reasonable and may improve the performance of service querying, words that can be used for enlarging the query term may not need to have only the same meaning as the query. Words that appear in the same topic or same concept are also able to be used for query expansion because those words refer to the same thing and are related. Additionally, the developers who create the ontology may refer to the concept by using technical terms or special words that are not in the dictionary. Consequently, applying only synonyms to enlarge the query terms may not be enough to improve the querying performance.

Because of these reasons, the ontology-based approach is also proposed. The approach employs a domain-specific ontology to achieve words that relate to a query term. The ontology used for query expansion is the same ontology used for service annotation. In this thesis, we apply the transport service ontology which is a four-level hierarchical knowledge structure and consists of many service concepts. Elements of each service concept are a concept name and concept descriptions. For example, a service concept named "Airline_Agent" comprises two concept descriptions such as "Airline Agent" and "Flight Agent". The main concept of this approach is to use terms on the same topic or, in this case, the same service concept with a given query term. That is, based on the above example, the term "airline", "flight" and "agent" are used for the purpose of query expansion if a query term is "airline", "flight" or "agent".

Like the WordNet-based expansion approach, enlarging a query term with too many words, which have both strong and weak relatedness, may decrease the performance of service querying. Therefore, two techniques for the ontology-based query expansion approach, namely *All-related terms expansion* and *Most-related terms extraction*, are provided.

1) All-Related Terms Expansion Technique

The process of the ontology-based approach with All-related terms expansion technique is shown in Figure 6.9. Firstly, the query expansion module creates an *EQT* with all query terms in *QT*. For each query term qt_i, the module connects to a co-occurrence matrix in order to get all terms that relate to qt_i. Finally, those related terms are added into the *EQT*.

```
Function Ontology-based approach with All-related terms technique
 (QT_Q : {qt_1, qt_2, …, qt_n})
   EQT_Q = QT_Q
   For each qt_i in QT_Q
      RelatedTerms = Lookup all related terms of qt_i from
                        the ontology-based co-occurrence matrix
      For each related_term in RelatedTerms
         Add related_term in EQT_Q
      End for
   End for
   Return EQT_Q
End Ontology-based approach with All-related terms technique
```

Figure 6.9 Pseudo-code of the ontology-based approach with All-related terms expansion technique

The co-occurrence matrix is an *n*-by-*n* matrix; where *n* is a number of noun terms which are extracted from service concepts at the bottom level of the ontology. That is, each row and each column refer to each term in the service concepts. Assumed that m_{ij} is an element at row i and column j of matrix m, the value of m_{ij} is a normalization of a number of times that $term_i$ and $term_j$ occur in the same service concept.

The pseudo-code for creating a co-occurrence matrix is shown in Figure 6.10. The query expansion module needs to connect to a domain-specific ontology in order to extract a list of terms in focused service concepts. In this case, there exist m actual service concepts (sc); $SC : \{sc_1, sc_2, ..., sc_m\}$. For each service concept sc_i, its service descriptions are separated into terms, stemmed and then all stopwords are removed. A set of terms in each concept description is $TermList : \{t_1, t_2, ..., t_k\}$; where k is a size of *TermList*. Only meaningful noun terms are added to a set of concept terms ($CT : \{ct_1, ct_2, ..., ct_n\}$), and an empty n-by-n matrix called *Matrix* is created. Next, values of elements in the co-occurrence matrix are the co-occurrence values between terms in all service concepts. For each service concept, pairs of terms (t_i,t_j) in its concept description are generated. The module finds indexes of both t_i and t_j in *CT* as *index-i* and *index-j* respectively. Then, a value of $Matrix_{index\text{-}i,\ index\text{-}j}$ and $Matrix_{index\text{-}j,\ index\text{-}i}$ are added by 1.

```
Function create_cooccurrence_matrix(SC : {sc1, sc2, …, scm})
   CT = {}
   For each sc in SC
      TermList = extract terms from concept descriptions of sc
      For each term in TermList
         Add term into CT
      End for
   End for
   Matrix = Create an empty n-by-n matrix m ; where n is a number of terms
in CT
   For each sc in SC
      TermList = extract terms from concept descriptions of sc
                 ; where TermList = {t1, t2, …, tk}
      For each ti in TermList
         For each tj in {tj+1, tj+2, …, tk}
            index-i = index of ti in CT
            index-j = index of tj in CT
            Matrixindex-i, index-j = Matrixindex-i, index-j + 1
            Matrixindex-j, index-i = Matrixindex-j, index-i + 1
         End for
      End for
   End for
End create_cooccurrence_matrix
```

Figure 6.10 Pseudo-code of the co-occurrence matrix creation method

For example, the service ontology consists of three actual service concepts; namely "Ship_Broker_Consultation", "Ship_Broker_Delivery" and

"Ship_Chartering". Concept descriptions of the service concept "Ship_Broker_Consultation" are "Barge Broker Consultation", "Ship Broker Consultation", "Vessel Broker Consultation", "Barge Broker Consultant", "Ship Broker Consultant" and "Vessel Broker Consultant", while descriptions of the concept "Ship_Broker_Delivery" are "Barge Broker Delivery", "Ship Broker Delivery" and "Vessel Broker Delivery". In contrast, the concept "Ship_Chartering" contains six descriptions; "Barge Rental", "Ship Rental", "Vessel Rental", "Barge Charter", "Ship Charter" and "Vessel Charter". Based on those service concepts, the concept term list (*CT*) is {barge, ship, vessel, broker, consultant, consultation, delivery, rental, charter}. Following the pseudo-code in Figure 6.10, the co-occurrence of the previous example is shown in Figure 6.11. Regarding the same query example in Section 6.4.1 , the ontology-based approach with All-related terms expansion technique enlarges a given query "vessel agent consultant" and then returns the *EQT* {vessel, agent, consultant, barge, ship, broker, delivery, rental, charter, consultation}.

	barge	ship	vessel	broker	delivery	rental	charter	consultant	consultation
barge	-	5	5	9	3	3	3	3	3
ship	5	-	5	9	3	3	3	3	3
vessel	5	5	-	9	3	3	3	3	3
broker	3	3	3	-	3	-	-	3	3
delivery	1	1	1	3	-	-	-	-	-
rental	2	2	2	-	-	-	3	-	-
charter	2	2	2	-	-	3	-	-	-
consultant	2	2	2	6	-	-	-	-	3
consultation	2	2	2	6	-	-	-	3	-

Figure 6.11 An example of the co-occurrence matrix

2) Most-Related Term Extraction Technique

The process of the ontology-based approach with the most related term extraction technique is presented in Figure 6.12. Like the All-related term expansion technique, given a set of query terms; $QT : \{qt_1, qt_2, ..., qt_n\}$, the *EQT* is created and added to all query terms in *QT*. To retrieve a related term of a query term qt_i, the query expansion module selects a term t_j that $Matrix_{qti,tj}$ is greater than other $Matrix_{qti,tk}$; where *Matrix* is a co-occurrence matrix, *k* is an index of terms in a list of concept terms *CT*, and *k* is not equal to *i* and *j*. The related term is added into *EQT*.

```
Function Ontology-based approach with Most-related term technique
(QT_Q : {qt_1, qt_2, …, qt_n})
   EQT_Q = QT_Q
   For each qt_i in QT_Q
      MostRelatedTerm = Lookup a related term with maximum relatedness value
of qt_i
                           from the ontology-based co-occurrence matrix
      Add MostRelatedTerm into EQT_Q
   End for
   Return EQT_Q
End Ontology-based approach with Most-related term technique
```

Figure 6.12 Pseudo-code of the ontology-based approach with Most-related term expansion technique

For example, given a query "vessel agent consultant" and based on the co-occurrence matrix in Figure 6.11, The *EQT* that is returned from the ontology-based approach with the Most-related term expansion technique is {vessel, agent, consultant, broker}.

6.5 Querying Module

As mentioned in Section 6.2 the querying module receives a set of expanded query terms (*EQT*), which contains extracted terms from a query *Q* and noun words relating to those terms, and then queries relevant service concepts. That is, the service querying process is quite similar to the annotation process. The annotation process attempts to find the relatedness between a service and a service concept, while the querying process finds the relatedness between a query and a service concept. Regarding the annotation methods, a service description of an SDE is extracted into service terms, and the method calculates the similarity value between those service terms and terms in concept descriptions of a service concept. Likely, the service querying process extracts query terms from a query, expands them, and then computes the similarity value between those query terms and concept terms. Therefore, in this thesis, the ideas of annotating services are applied into the service querying process.

Similar to the service annotation module in Chapter 5, three semantic service querying approaches, namely ECBR, Vector-based and Classification-based querying approach, are proposed and explained in Section 6.5.1 , 6.5.2 and 6.5.3 respectively. In addition, three querying types; single-label, multi-label and combine-label querying, are also proposed. Given a query *Q,* the single-label based querying approach returns only one relevant service concept, while the multi-label based querying approach queries multiple relevant concepts. The combine-label approach applies the multi-label approach to get relevant concepts, but, in the case that the result is empty, the result from the single-label approach is returned.

6.5.1 ECBR Querying Approach

The ECBR querying approach is the same as the ECBR annotation approach in Section 5.4.1 . It applies an idea of semantic term matching to find service concepts that are relevant to a query. Every query term in the expanded query term list (*EQT*) of a query is matched with every concept term of a service concept. If the query term is the same as the concept term, the matching value is 1. If the query term is a synonym of the concept, the matching value is 0.5; otherwise, the value is 0. The relevance score between a query and a service concept is the summation of matching scores of all (query term, concept term) pairs divided by the number of query terms.

The ECBR querying approach is categorized into single-label based, multi-label based and combine-label based approach. Given a query, the output of the single-label ECBR approach is a service concept that returns the highest relevance value and that score is greater than a defined query threshold *(QT).* In contrast, the multi-label based approach returns all service concepts whose relevance scores are greater than *QT*. The output of the combine-label based approach is the same as the multi-label approach, but, in the case that the output is empty, a service concept that gives the highest relevance score is returned.

6.5.2 Vector-based Querying Approach

To query relevant service concepts, for each service concept *C,* the vector-based querying approach represents a query *Q* and a service concept *C* by using a vector and then calculates the cosine similarity between both vectors. A service concept *C* is queried if the similarity value is greater than the threshold. Similar to the vector-based annotation approach in Chapter 5, the vector-based querying approach is divided into two types, namely a single-label vector-based and a multi-label vector-based approach. Given a query, the single-label approach gets only one relevant service concept, while the multi-label one is able to return multiple relevant service concepts.

1) Single-Label Vector-based Querying

The pseudo-code of the single-label vector-based querying process is shown in Figure 6.13. The querying module receives a set of expanded query terms (*EQT*) from the query expansion module and then creates an *EQT*-vector with p elements, where p is the number of concept terms (CT). Each element of *EQT*-vector is calculated from the number of query terms or their synonyms matching with the concept term ct_i, where i is an index of the concept term and $1 \leq i \leq p$. Like the vector-based annotation, the *EQT* is represented by using VSM and EVSM vector. The process of creating those vectors is the same as the process in Section 5.4.2 .

```
Function Single-label Vector-based Querying
   (EQT: {eqt1, eqt2, …, eqtn}, SC : {sc1, sc2, …, scm})
   EQT-Vector = create a term-occurrence vector of EQT
   For each sci in SC
      Sc_desc = get a set of descriptions of sci; Sc_desc = {scd1, scd2, …, scdp}
      For each scdi in Sc_desc
         Concept_terms = meaningful noun words in scdi
         Add all terms in Concept_terms in CT
      End for
      SC-Vector = create a term-occurrence vector of CT
      sim_val = cosine-similarity(EQT-Vector, SC-Vector)
      add sim_val into SIM-List
   End for
   Return a service concept sc with the maximum value of SIM-List
End Single-label Vector-based Querying
```

Figure 6.13 Pseudo-code of the single-label vector-based querying approach

To discover a service concept that is relevant to a query *Q*, the module also generates a vector that represents all descriptions of each service concept, called Service Concept Vector (*SC-vector*), and computes the cosine similarity between *EQT-vector* and *SC-vector*. The service concept that returns the maximum similarity is then selected.

2) Multi-Label Vector-based Querying

The pseudo-code of the multi-label vector based approach is displayed in Figure 6.14. The whole process of this querying approach is the same as the single-label vector based approach. Like the single-label based approach, the multi-label based approach has to create vectors of a query and every service concept and then calculate the similarity values. The difference is that it is able to select many concepts whose similarity reaches the defined query threshold *(QT).*

```
Function Multi-label Vector-based Querying
(EQT: {eqt1, eqt2, …, eqtn}, SC : {sc1, sc2, …, scm})
   EQT-Vector = create a term-occurrence vector of EQT
   For each sci in SC
      Sc_desc = get a set of descriptions of sci; Sc_desc = {scd1, scd2, …,
scdp}
      For each scdi in Sc_desc
         Concept_terms = meaningful noun words in scdi
         Add all terms in Concept_terms in CT
      End for
      SC-Vector = create a term-occurrence vector of CT
      sim_val = cosine-similarity(EQT-Vector, SC-Vector)
      add sim_val into SIM-List
   End for
   Return a set of service concepts {sc1, sc2, …, scq}
         that their similarity values are greater than a threshold
         ; q is the number of relevant service concepts
End Multi-label Vector-based Querying
```

Figure 6.14 Pseudo-code of the multi-label vector-based querying approach

6.5.3 Classification-based Querying Approach

Apart from using vector-based techniques, the querying problem is also treated as a classification task. That is, based on a given query *Q,* this approach attempts to categorize *Q* into relevant service classes which are represented by service concepts. In this research, pattern classification techniques in Neural Networks and Machine Learning are applied in order to classify a query into one or many service concepts. The principal idea of this approach is similar to the classification-based approach for semantic service annotation in Section 5.4.3 which classifies a service into relating one or more service concepts. Therefore, in this section, the proposed classification-based querying approaches are briefly explained.

1) Single-Label Classification-based Querying

Given a query *Q,* the single-label based approach discovers the most relevant service class. The working process of this approach is presented in Figure 6.15. To solve the service querying problem, both neural network and machine learning techniques are employed. Thus, to describe the overview of this approach, query classifiers of both techniques are called *MACHINE*. Firstly, the approach needs to generate a set of concept terms which are used for calculating the input values of the *MACHINE*s. The meaningful noun terms in the descriptions of each concept in the service ontology are extracted to be concept terms. Secondly, a query classifier, *MACHINE*, is created with *k*-input nodes and *m*-output nodes, where *k* is the number of concept terms and *m* is the number of service concepts. Then, based on a training dataset, pairs of a query and a relevant concept; (*q, sc*), are fed to the *MACHINE* for training. After training, the *MACHINE* is ready to use for querying a relating service concept of any query.

```
Function Single-label Classification-based Querying
   (EQT: {eqt1, eqt2, …, eqtn}, SC : {sc1, sc2, …, scm})
   For each scdi in Sc_desc
      Concept_terms = meaningful noun words in scdi
      Add all terms in Concept_terms in CT
   End for (CT: {ct1, ct2, …, ctk})
   MACHINE = Create a two-layer neural network or a machine learning
with k input nodes and m output nodes
   Train MACHINE
   ---------------------------------------------------------------------
   // Run the trained query classifier
   A relevant concept RC <= Run the MACHINE
   Return RC
End Single-label Classification-based Querying
```

Figure 6.15 Pseudo-code of the single-label classification-based querying approach

An example of the neural network-based machine for single-label query classifier is displayed in Figure 6.16. It is definite that the structures of the neural network for classifying the services in Figure 5.15 and the queries in Figure 6.16 are almost identical. The difference is that the network of the single-label query classifier receives a set of expanded query terms (*EQT*) as the input, while the network of the single-label service classifier receives a set of service terms.

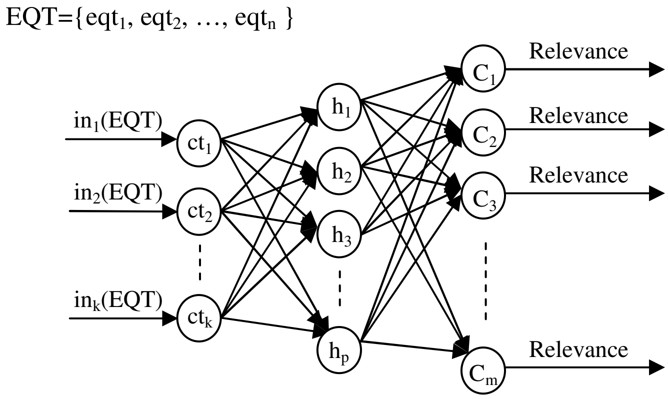


Figure 6.16 The structure of feed-forward neural network for the single-label query classifier

Input Value of a query classifier

Given a set of expanded query terms (*EQT: {eqt_1, eqt_2, …, eqt_n}*) from the query expansion module, an input value of each input node in the query classifier is computed from matching score between query terms in *EQT* and a concept term of that input node. If a query term is same as a concept term, their matching score is 1. If a query term is a synonym of a concept term, their matching score is 0.5; otherwise, the score is 0. That is, the input value of the input node *i* *in_i(EQT)*, is calculated as follows.

$$in_i(EQT) = \sum_{j=1}^{n} matching(eqt_j, ct_i)$$

$$matching(x, y) = \begin{cases} 1, & if\ x\ is\ equal\ to\ y \\ 0.5, & if\ x\ is\ a\ synonym\ of\ y \\ 0, & otherwise \end{cases}$$

Training and Testing Data Preparation

To classify the queries into their relevant service concept, the query classifier needs to be trained for recognizing a query classification pattern. Then, the performance of the proposed querying approach is tested by using the testing data. Consequently, the training and testing data set is required. Based on the structure of the query classifier, an input of the classifier is a query and an output is the most relevant service concept. Given a query *Q* and a domain-specific ontology *O,* the data set consists of pairs of *Q* and a relevant service concept *C* in O; (Q, C). To prepare the data set for single-label classification in service querying, descriptions in each service concept are applied. As mentioned about the structure of the service ontology, the ontology consists of service concepts and each concept comprises two main properties, namely a concept name and concept descriptions. Therefore, pairs of a concept description and its service concept are generated. For example, a service concept "Airline_Agent" comprises two descriptions; "Airline Agent" and "Flight Agent", so generated pairs of data are {"Airline Agent", "Airline_Agent"} and {"Flight Agent", "Airline_Agent"}.

In addition, new concept descriptions are also created by replacing terms in the description with their synonyms. Given a query term *QT*, its synonyms are looked up from WordNet 3.1 and only synonyms from proper senses of words are selected. For example, based on the WordNet 3.1 database, the term "Airline" consists of two senses of words; 1) a hose that carries air under pressure and 2) a commercial enterprise that provides scheduled flights for passengers. In this case, the second sense is selected and synonyms with that sense are "airline business" and "airway". Consequently, new data pairs; {"Airline Business Agent", "Airline_Agent"} and {"Airway Agent", "Airline_Agent"}, are generated.

Query Classifier Type

To classify a query into a relevant service concept, both neural network, and machine learning techniques are applied. To define a query classifier, this thesis applies two types of artificial neural network; multilayer feed-forward neural network and radial basis function network, and two types of machine learning algorithm; K-nearest neighbor and classification tree algorithm.

2) Multi-Label Classification-based Querying

Given a domain-specific service ontology, the multi-label classification based querying approach categorizes a query into multiple ontological relevant service concepts. The pseudo-code of this approach is displayed in Figure 6.17. Like the single-label based approach, first of all, the multi-label approach generates a set of concept terms (CT) that are extracted from service concept descriptions, creates a query classifier, and then trains the classifier by using the prepared training data set. After that, the query classifier is ready for querying relevant service concepts.

```
Function Multi-label Classification-based Querying (EQT: {eqt1, eqt2, …,
eqtn}, SC : {sc1, sc2, …, scm})
   For each scdi in Sc_desc
      Concept_terms = meaningful noun words in scdi
      Add all terms in Concept_terms in CT
   End for (CT: {ct1, ct2, …, ctk})
   MACHINE = Create m two-layer neural networks or m machines
with k input nodes and 2 output nodes
                ; MACHINE = {m1, m2, …, mm}
   For mi in MACHINE
      Train mi
   End for
   ----------------------------------------------------------------------
   For mi in MACHINE
   // Run the trained machine
      output = Run mi
      If output is the first class
         Add sci into a relevant concept set : RC
      End if
   End for
   Return RC
End Multi-label Classification-based Querying
```

Figure 6.17 Pseudo-code of the multi-label classification-based querying approach

The main part of this approach is to define and create a query classifier that classifies a query to the relevant concepts. Similar to the single-label approach, both neural network and machine learning techniques are applied for query classification. The difference is that the multi-label approach needs multiple machines, while the single-label approach requires only one machine to solve querying problem. The example of the neural network-based machines for multi-label querying classifier is shown in Figure 6.18. Each machine in the query classification is defined for each service concept and discovers whether a query belongs to that concept. Thus, the classifier contains *m* machines where *m* is the number of ontological service concepts and each machine comprises *k* input nodes and 2 output nodes, where *k* is the number of all concept terms. Because the classifier consists of multiple machines, each machine is trained separately with pairs of a query and a belonging value of a service class of that machine.

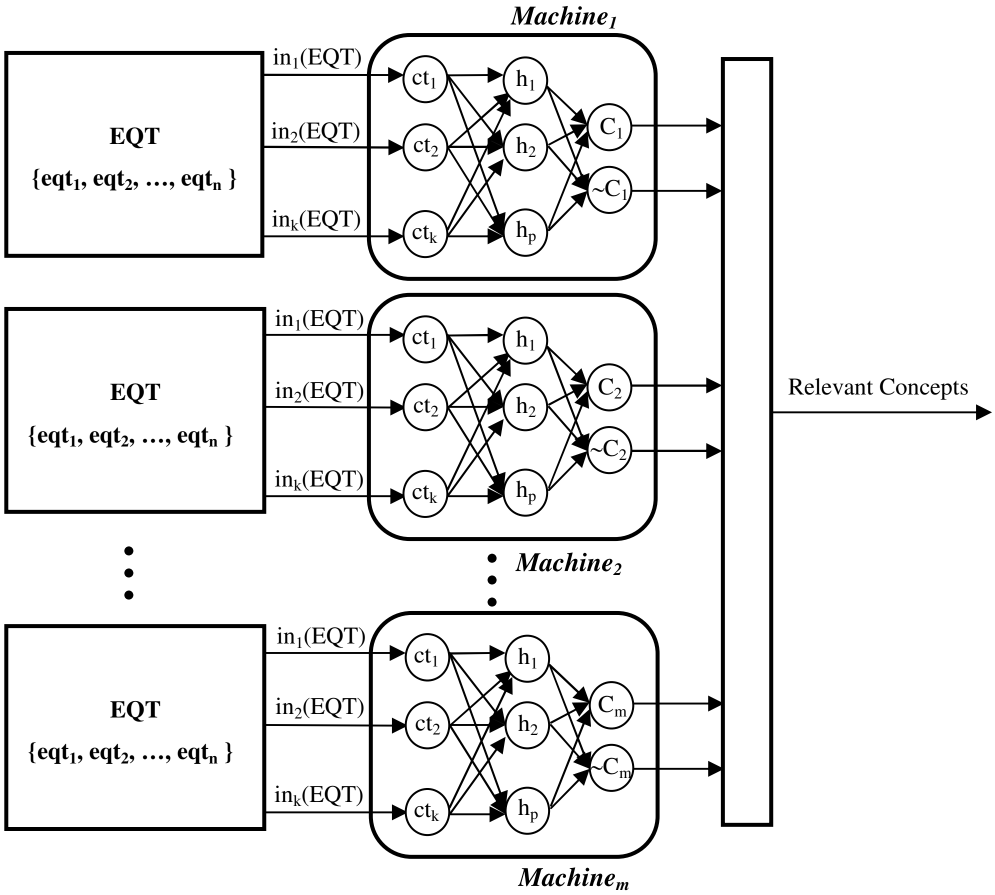


Figure 6.18 The structure of feed-forward neural network for the multi-label query classifier

Input Value of a query classifier

The structure in Figure 6.18 presents that the query classification contains *m* classification machines. A machine *i* clarifies that a query belongs to service concept *i* or not. Thus, given a query, all machines run separately and their results are combined at the end of the querying process. The input nodes and input values are defined as same as those in the single-label approach. That is, given a query *Q,* the input value of each input node is calculated from the number of term matching between expanded query terms of *Q* and a concept term of that input node. If a query term is exactly matched with a concept term, the matching score is added by 1. If a query term is a synonym of a concept term, the score is added by 0.5, otherwise 0.

Training and Testing Data Preparation

The training and testing data set for the multi-label based querying approach contains pairs of a query and its relevant service concepts. For Instance, a training data (Q, $\{SC_1, SC_2\}$) means a query Q is relevant to two service concepts-SC_1 and SC_2. To create the dataset for the multi-label classifier, the data set for each classification machine is separately generated. A training and test data of a machine i are represented by a pair of a query and an output class in machine i. The output classes of each machine are defined as class 1 and class 2. Given a query Q, if an output of machine i is class 1, it means that Q belongs to a service concept i. In contrast, if the output is class 2, it means that Q is not in a service concept i. For example, there exist 3 service concepts; SC_1, SC_2, and SC_3, so there are three classification machines; M_1, M_2, and M_3, for classifying SC_1, SC_2, and SC_3 respectively. Based on the previous example, the training data sets for M_1, M_2, and M_3 are (Q, 1), (Q, 1) and (Q, 2) respectively.

To automatically generate a dataset for training and testing the multi-label classifier, parent nodes of actual service concepts are considered. In this thesis, the transport service ontology is applied. The ontology consists of abstract service concepts and actual service concepts. The abstract concepts represent domains of transport services, while the actual ones relate to the real business services. Given a parent node of actual service concepts, its concept descriptions are assigned as queries, and the child nodes are assigned as relevant concepts of the given queries.

For example, an abstract concept "Baggage_Agent_Abstract" has a concept description; "Baggage Agent", and consists of two actual service concepts; "Baggage_Delivery" and "Baggage_Pick-up". Therefore, the generated training data is ("Baggage Agent", {"Baggage_Delivery", "Baggage_Pick-up"}). Based on the structure of the query classifier, the training data sets for M_1, M_2 and M_3 are ("Baggage Agent", 1), ("Baggage Agent", 1) and ("Baggage Agent", 0) respectively; in the case that machine M_1, M_2, and M_3 are applied to classify concept "Baggage_Delivery", "Baggage_Pick-up", and "Airline_Agent" respectively. Similar to the training data of the single-label approach, the data set is also created by replacing query terms with their synonyms. For instance, the word "Luggage" is a synonym of the word "Baggage"; hence ("Luggage Agent", 1), ("Luggage Agent", 1) and ("Luggage Agent", 0) are added to the data set of M_1, M_2 and M_3 respectively.

Besides categorizing a query into multiple concepts, the multi-label should be also able to correctly classify the query into one relevant concept. Consequently, the data set that is generated for the single-label classifier is also included in the data set of the multi-label classifier. That is, apart from using descriptions of parent abstract concepts, the concept descriptions of all actual concepts are considered for preparing the training and testing data set.

Query Classifier Type

Like the single-label classifier, both neural network and machine learning technique are employed for recognizing the service concepts. The neural network-based multi-label classifier is created from two kinds of neural network; multilayer feed-forward neural networks and radial basis function networks, while machine learning-based classifier is divided into three types; k-nearest neighbor-based, classification tree-based and support vector machine-based classifier.

3) Combine-Label Classification-based Querying

Like the multi-label approach, the combine-label querying approach receives a set of expanded query terms (*EQT*) of a query *Q* and then categorizes it into multiple relevant service concepts. The query classifier of the multi-label approach in Figure 6.18 shows that each classification machine separately receives a query and detects whether the query belongs to a service concept of that machine. Unfortunately, it is possible that a query is not classified to any service concept. That is, all classification machines classify the query into class 2, which represents that the query is irrelevant to service concepts of all machines. This means that the multi-label approach is unable to return results of several queries, so a combine-label classification-based querying approach is proposed to solve this problem.

The pseudo-code of the combine-label querying approach is presented in Figure 6.19. The result of this approach is a combination of the result of the single-label and multi-label approach. Firstly, the multi-label approach is applied to retrieve relevant concepts: RC-multi, of a query. If there is no relevant concept, the single-label approach is invoked and then returns a relevant concept; otherwise, the result from the multi-label approach is returned. Because the output nodes of the single-label approach are service concepts, it promises that a relevant concept is returned. That is, the combine-label approach is able to retrieve relevant service concepts for any queries.

```
Function Combine-label Classification-based Querying (EQT: {eqt1, eqt2, …,
eqtn}, SC : {sc1, sc2, …, scm})
   RC-multi = Multi-label Classification-based Querying (EQT, SC)
   if RC-multi is empty
      RC-single = Single-label Classification-based Querying (EQT, SC)
      return RC-single
   else if RC-multi is not empty
      return RC-multi
   end if
End Combine-label Classification-based Querying
```

Figure 6.19 Pseudo-code of the multi-label classification-based querying approach

Input Value of a query classifier

Query classifiers in the single-label and multi-label approach are trained separately, hence the input values in both approaches are computed in the same way as the formula in the previous sections.

Training and Testing Data Preparation

Given a query *Q,* a result of the combine-label approach is one or more relevant service concepts. Consequently, the same data set that is used in the multi-label approach is applied. That is, the data set is generated from concept descriptions of actual service concepts and their parent abstract concepts. Thus, this data set contains both pairs of a query and a relevant concept; (q, sc), and pairs of a query and multiple relevant concepts; (q, $\{sc_1, sc_2, ..., sc_i\}$), where i is the number of relevant concepts.

Query Classifier Type

Because the combine-label approach applies both the single-label and multi-label based approach to query relevant services, a query classifier for this approach needs to use the same type of classifier in both single-label and multi-label approach. Based on the single-label approach, to create the query classifier, the combine-label approach applies two types of artificial neural networks; multilayer feed-forward neural network and radial basis function network, and two kinds of machine learning algorithms; k-nearest neighbor and classification tree.

6.6 Hybrid Querying Approach

The focus of this chapter is on querying service concepts that relate to a query. Given a query *Q,* a set of expanded query terms of *Q* is generated by using the proposed query expansion approaches. Those expanded terms are then fed to the querying module to discover relevant concepts.

As mentioned in Section 6.4 , there are two types of the proposed query expansion approaches, namely, WordNet-based and ontology-based expansion. While the WordNet-based querying approach applies WordNet to expand query terms before sending them to the querying module in order to pick up relevant concepts, the ontology-based querying approach applies descriptions in ontological concepts.

In this section, the hybrid querying approach that combines results from the WordNet-based and the ontology-based querying approach is proposed. The workflow of the hybrid based approach for service querying is shown in Figure 6.20. Given a query *Q,* the query term extraction module separates *Q* into query terms *QT* and then sends them to both WordNet-based and ontology-based query expansion module. Expanded query terms (*EQT*) of each expansion method are sent to its own querying module in order to get relevance scores between the *EQT* and every service concept. The WordNet-based and the ontology-based querying modules apply the same querying approach in Section 6.4 and $\{rv_{w1}, rv_{w2}, \ldots, rv_{wk}\}$ and $\{rv_{o1}, rv_{o2}, \ldots, rv_{ok}\}$ are results of the WordNet-based and ontology-based approach respectively, where rv_{wi} and rv_{oi} are a relevance value between the concept i and terms that are expanded by using WordNet and ontology respectively, and k is the number of service concepts. Next, service concepts with relevance values from both methods are fed to the hybrid querying module in order to retrieve relevant service concepts.

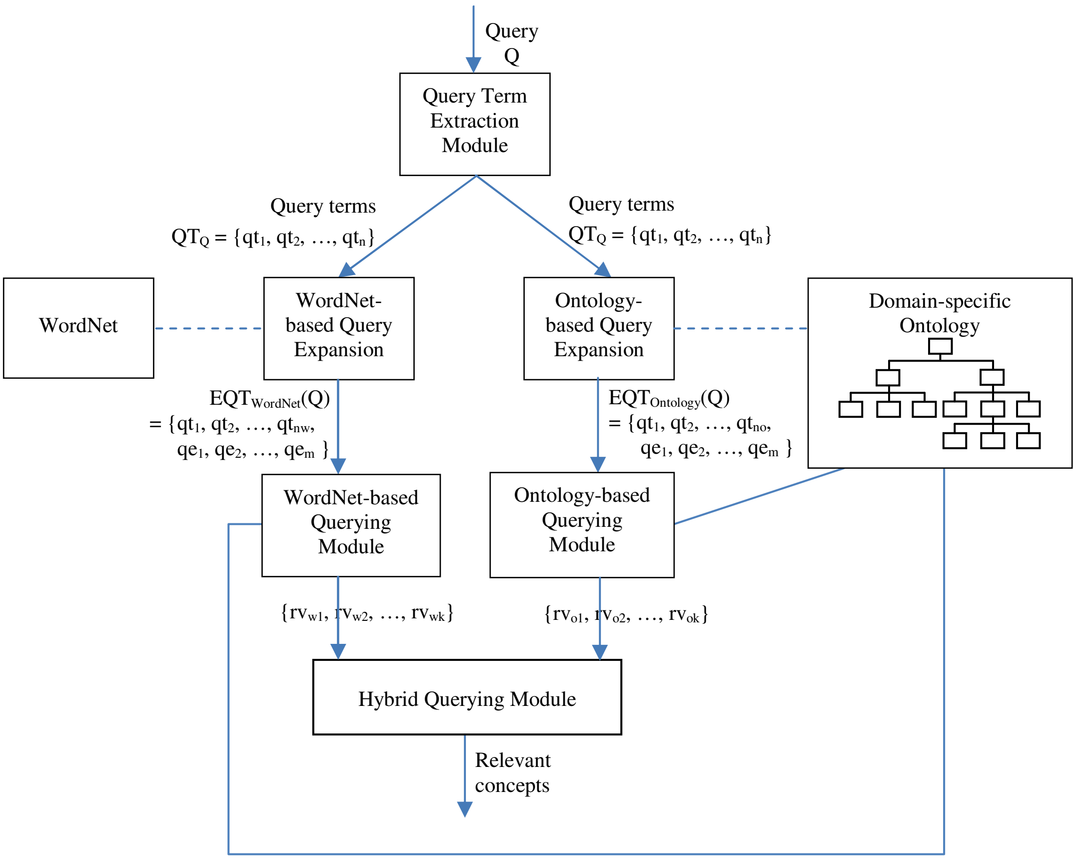


Figure 6.20 Workflow of the hybrid based approach for service querying

```
Function Hybrid Querying (RVW: {rvw1, rvw2, …, rvwk}, RVO: {rvo1, rvo2, …,
rvok}, SC : {sc1, sc2, …, sck})
   For each i from k
      hybrid_sci = calculate hybrid relevance value of sci by using (1)
      if hybrid_sci > query threshold
         Add sci into a relevant concept set : RC
      end if
   end for
   return RC
End Hybrid Querying
```

Figure 6.21 Pseudo-code of the hybrid querying module

6.6.1 The Hybrid Querying Module

The hybrid querying module receives results from both WordNet-based and ontology-based querying module and then combines them to get relevant concepts. The pseudo-code of this module is presented in Figure 6.21. Given a query Q, RV_W and RV_O are a set of relevance values between Q and every service concept based on

WordNet and ontology-based approach respectively. For each service concept *i,* the hybrid-based relevance value of the concept i is calculated as follows,

$$rv_{hi} = (w_w * rv_{wi}) + (w_o * rv_{oi}) \qquad (1)$$

, where rv_{hi}, rv_{wi} and rv_{oi} are the hybrid-based, WordNet-based and ontology-based relevance value between the query and the concept i respectively, and w_w and w_o is the weight of WordNet-based and ontology-based value. If rv_{hi} is greater than the defined query threshold *(QT)*, the service concept i is added into the relevant concept list.

6.7 Experiments

This section presents the experimental results of the proposed semantic service querying approaches, namely the ECBR, the vector-based and the classification-based approaches. Following the service querying types, the results of each approach are divided into three sub-categories; single-label, multi-label and combine-label querying. To evaluate the performance of the semantic service querying approaches, well-known performance measures in the area of information retrieval, such as the average precision, average recall, average f-measure and average fallout rate, are applied. Moreover, the querying rate (percentage of hitting query samples), and all-measures combination are also focused on. The value of the all-measures combination is calculated as follows:

$$\begin{aligned} All\_measures \\ &= (0.4 * avg\_precision) + (0.15 * avg\_recall) + (0.15 \\ &* avg\_fmeasure) + (0.15 * (1 - avg\_fallout)) + (0.15 \\ &* querying\_rate) \end{aligned}$$

As mentioned previously, the proposed query expansion methods expand query terms and then send those expanded terms to the querying approaches in order to get the results. Based on the query expansion methods in Section 6.4 , the performance of every service querying approach with every query expansion method is also tested. In this thesis, six query expansion methods are proposed; namely 1) WordNet-based approach with All-senses expansion technique (QE-1), 2) WordNet-based approach with Proper-sense expansion technique (QE-2), 3) Ontology-based approach with All-related terms expansion technique (QE-3), 4) Ontology-based approach with Most-related terms expansion technique (QE-4) 5) Hybrid-based approach with All-terms expansion technique (QE-5 : QE-1 + QE-3) and 6) Hybrid-based approach with Proper-term expansion technique (QE-6 : QE-2 + QE-4). In contrast, QE-0 is a non-query expansion method.

For every query expansion method, each service querying approach with settings querying variables is tested, hence there exist a large number of experimental results for semantic service querying. In this chapter, the querying performances by setting the querying variable as the value that returns the best results are presented, while the rest of the results are shown in Appendix A.

6.7.1 ECBR Querying Approach

To evaluate the ECBR based querying approach, for all querying types, performance was tested by setting the Querying Threshold *(QT)* from 0.1 to 0.9 with an increment of 0.1. Given a query *Q,* this approach returns a relevant service concept if its relevance score is greater than *QT*. In this section, the performance of the ECBR querying approach with a setting *QT* as 0.7 is presented.

Single-Label based Approach

The performances of the ECBR approach for a single-label service querying are shown in Table 6.1. The ECBR approach with QE-5 gave the best performance with 54.90% in precision and recall, and it was able to get the results from 99.96% of all query samples. That is, QE-5 improved the precision and recall value around 10% when comparing with the approach with no query expansion method (QE-0). On the other hand, the querying approach with QE-3 returned the worst performance with 38.68% in precision and recall, which was less than the approach with QE-0. It means that terms expanded from the ontology may be irrelevant to the query.

Table 6.1 The experiments of ECBR approach for a single-label service querying, *QT*=0.7

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	44.79%	44.79%	44.79%	0.21%	99.25%	61.21%
QE-1	51.09%	51.09%	51.09%	0.19%	99.96%	65.73%
QE-2	49.83%	49.83%	49.83%	0.19%	99.79%	64.82%
QE-3	38.68%	38.68%	38.68%	0.24%	99.25%	56.92%
QE-4	41.33%	41.33%	41.33%	0.23%	99.25%	58.79%
QE-5	54.90%	54.90%	54.90%	0.17%	99.96%	68.40%
QE-6	51.26%	51.26%	51.26%	0.19%	99.79%	65.82%

Multi-Label based Approach

The results of the ECBR approach for multi-label service querying are displayed in Table 6.2. Similar to the single-label based ECBR approach, the querying approach with QE-5 returned the best result with 52.63% in precision and 72.19% in recall, and it was able to query the samples 61.78%. In contrast, the ECBR approach with no query expansion gave the worst performance with 42.71% in precision and 50.13% in recall. That is, all proposed query expansion methods are able to improve the performance of the ECBR approach for multi-label service querying. Moreover, it is clear that the ontology-based query expansion methods (QE-3 and QE-4) assisted the ECBR multi-label querying approach to considerably increase the querying rate.

Table 6.2 The experiments of ECBR approach for a multi-label service querying, *QT*=0.7

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	42.71%	50.13%	42.51%	0.29%	27.32%	50.03%
QE-1	48.17%	63.66%	50.27%	0.37%	35.74%	56.66%
QE-2	46.74%	59.22%	47.97%	0.35%	32.75%	54.63%
QE-3	44.11%	77.29%	50.68%	0.61%	75.16%	63.02%
QE-4	48.28%	69.78%	52.99%	0.37%	64.02%	62.28%
QE-5	52.63%	72.19%	56.58%	0.35%	61.78%	64.58%
QE-6	45.97%	61.10%	48.51%	0.35%	50.48%	57.35%

Combine-Label based Approach

Table 6.3 shows the results of the ECBR approach for combine-label service querying with a setting *QT* as 0.7. For each query sample, the multi-label based approach is applied to get a result. If the result is empty, the single-label based approach is then applied. Consequently, the querying rates in all query expansion methods were much greater than those in the multi-label approach. Like the ECBR multi-label querying approach, the ECBR combine-label approach with QE-5 returned the best performance with 57.42% in precision, while the approach with QE-0 gave the worst result with 47.09% in precision. Furthermore, the query expansion methods that apply expanding terms from the ontology (QE-3, QE-4, QE-5 and QE-6) improved the recall values; 59.12-67.61%, of the ECBR querying approach.

Table 6.3 The experiments of ECBR approach for a combine-label service querying, *QT*=0.7

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	47.09%	46.66%	44.81%	0.22%	99.16%	62.40%
QE-1	53.86%	56.87%	52.32%	0.24%	99.96%	67.88%
QE-2	51.67%	53.27%	49.82%	0.24%	99.80%	66.07%
QE-3	44.43%	68.74%	48.61%	0.51%	99.16%	65.17%
QE-4	48.01%	60.48%	49.75%	0.31%	99.16%	65.57%
QE-5	57.42%	67.61%	58.10%	0.26%	99.96%	71.78%
QE-6	53.46%	59.12%	52.92%	0.25%	99.80%	68.12%

6.7.2 Vector-based Querying Approach

To evaluate the vector-based approach for single-label, multi-label and combine-label service querying, performance was tested by setting the Querying Threshold *(QT)* from 0.1 to 0.9 with an increment of 0.1. Given a query *Q,* this approach returns a relevant service concept if its relevance score is greater than *QT*. As mentioned in Section 6.5.2 , based on the vector representation, the vector-based querying approach is categorized into two types; VSM-based and EVSM-based querying approach. In this section, the performance of the vector-based querying approach with a setting *QT* as 0.9 is selected and presented as follows. The *QT* was selected as 0.9 since the best results were obtained at this threshold.

Single-Label based Approach

1) VSM-based Approach

The performance of the VSM-based approach for single-label service querying is presented in Table 6.4. The performance measures, precision, recall and f-measure, of the querying approach with all query expansion methods were quite high; greater than 60%. The approach with QE-5 returned the best result with 77.54% in precision and recall, while one with no query expansion method (QE-0) showed the worst performance with 61.59% with precision and recall. Moreover, the querying rate in every query expansion method was very high with more than 97% of all queries.

Table 6.4 The experiments of VSM-based approach for a single-label service querying, *QT*=0.9

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	61.59%	61.59%	61.59%	0.15%	97.87%	72.77%
QE-1	73.09%	73.09%	73.09%	0.10%	99.54%	81.08%
QE-2	69.59%	69.59%	69.59%	0.12%	98.70%	78.50%
QE-3	62.29%	62.29%	62.29%	0.15%	98.24%	73.32%
QE-4	66.89%	66.89%	66.89%	0.13%	98.24%	76.54%
QE-5	77.54%	77.54%	77.54%	0.09%	98.41%	84.03%
QE-6	70.67%	70.67%	70.67%	0.11%	98.33%	79.20%

2) EVSM-based Approach

The performance of the EVSM-based approach for single-label service querying is shown in Table 6.5. Like the VSM-based single-label approach, the performances of the approach with all expansion methods were greater than 60% in precision, recall and f-measure. The querying rates of this approach were greater than 98%, which were a little better than those in the VSM-based approach. The best performance of this approach came with using QE-5 to expand query terms; 77.33% in precision and recall, while using QE-0 gave the worst result with 63.14%. The EVSM-based approach with QE-5 and QE-6 required more processing time than the other expansion methods because the QE-5 and QE-6 had to expand queries by using both WordNet and the service ontology.

Table 6.5 The experiments of EVSM-based approach for a single-label service querying, QT=0.9

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	63.14%	63.14%	63.14%	0.14%	98.12%	73.89%
QE-1	71.57%	71.57%	71.57%	0.11%	99.54%	80.02%
QE-2	69.21%	69.21%	69.21%	0.12%	98.83%	78.25%
QE-3	64.57%	64.57%	64.57%	0.14%	98.41%	74.94%
QE-4	63.55%	63.55%	63.55%	0.14%	98.41%	74.22%
QE-5	77.33%	77.33%	77.33%	0.09%	98.41%	83.88%
QE-6	72.56%	72.56%	72.56%	0.11%	98.41%	80.54%

Multi-Label based Approach

1) VSM-based Approach

The performance of the VSM-based approach for multi-label service querying is displayed in Table 6.6. The performance measures of the VSM-based multi-label approach with all query expansion methods were very good and higher than around 65% in all precision, recall and f-measure. The results of the approach with both QE-5 and QE-6 were around 90% in all measures, largely outperforming those of the approach with other expansion methods, at around 70% in all measures. In contrast, the results in Table 6.6 clearly demonstrated that the querying rates of the VSM-based multi-label approach were too low with less than 22% of all query samples. This is because QT was set as 0.9 and the approach returned service concepts that their relevance values were more than QT. As a result, few service concepts were returned.

Table 6.6 The experiments of VSM-based approach for a multi-label service querying, QT=0.9

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	66.61%	66.48%	64.92%	0.20%	11.38%	63.03%
QE-1	77.10%	75.24%	74.06%	0.13%	4.27%	68.86%
QE-2	72.82%	70.23%	69.91%	0.15%	5.95%	66.02%
QE-3	69.46%	67.38%	65.79%	0.17%	10.94%	64.38%
QE-4	73.06%	71.04%	70.09%	0.13%	21.96%	68.67%
QE-5	95.00%	92.57%	91.66%	0.03%	2.00%	80.93%
QE-6	91.33%	87.43%	87.27%	0.05%	3.91%	78.31%

2) EVSM-based Approach

Table 6.7 shows the performance of the EVSM-based approach for multi-label service querying. The EVSM-based approach with QE-5 performed very well in all performance measures; 95.28% in precision, 91.24% in recall and 90.50% in f-measure, while the approach with no query expansion gave the worst performance with 58.55%, 55.38% and 55.26% in precision, recall, and f-measure respectively. Similar to the VSM-based multi-label approach, with a setting of *QT* as 0.9, the EVSM approach with all query expansion methods was able to query just a small number of query samples and this led the querying rates to be lower than 17%.

Table 6.7 The experiments of EVSM-based approach for a multi-label service querying, *QT*=0.9

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	58.55%	55.38%	55.26%	0.18%	4.67%	55.69%
QE-1	79.82%	85.57%	79.95%	0.14%	16.37%	74.19%
QE-2	67.85%	73.34%	67.24%	0.22%	10.10%	64.71%
QE-3	67.62%	60.62%	61.12%	0.14%	4.87%	61.02%
QE-4	71.68%	66.57%	67.00%	0.12%	7.83%	64.86%
QE-5	95.28%	91.24%	90.50%	0.03%	2.12%	80.69%
QE-6	87.74%	83.69%	82.95%	0.08%	2.12%	75.40%

Combine-Label based Approach

1) VSM-based Approach

The performance of the VSM-based approach for combine-label service querying is presented in Table 6.8. The performance of the VSM-based approach with no query expansion was worse than other query expansion methods; 62.58% in precision and 59.10% in recall, while the approach with QE-5 gave the best result with 78.11% in precision and 73.80% in recall. By comparing with the VSM-based multi-label approach, the VSM-based combine-label approach was able to increase very considerably the querying rate from less than 20% to around 99%. Although the performances of QE-5 and QE-6 dropped from approximate 90% in multi-label approach to around 70% in combine-label approach, the performances of other expansion methods in both multi-label and combine-label were not much different.

Table 6.8 The experiments of VSM-based approach for a combine-label service querying, *QT*=0.9

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	62.58%	59.10%	59.33%	0.15%	97.56%	72.41%
QE-1	73.66%	69.58%	69.98%	0.10%	99.52%	80.31%
QE-2	69.72%	65.60%	66.03%	0.12%	98.60%	77.40%
QE-3	63.51%	59.88%	60.11%	0.15%	97.92%	73.07%
QE-4	67.57%	64.24%	64.38%	0.13%	97.92%	75.99%
QE-5	78.11%	73.80%	74.30%	0.08%	98.24%	83.18%
QE-6	71.46%	67.14%	67.63%	0.11%	98.16%	78.51%

2) EVSM-based Approach

The results of the EVSM-based approach for combine-label service querying are shown in Table 6.9. Like the single-label based and multi-label based querying approach, the EVSM with QE-5 returned the best performance with 77.83% in precision and 73.51% in recall, while the EVSM with no query expansion gave the worst performance with 63.66% in precision and 59.97% in recall. Applying the combine-label querying type largely increased the querying rate from less than 20% in the multi-label EVSM approach to around 98%.

Table 6.9 The experiments of EVSM-based approach for a combine-label service querying, *QT*=0.9

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	63.66%	59.97%	60.36%	0.14%	97.96%	73.19%
QE-1	71.88%	69.35%	68.82%	0.12%	99.52%	79.39%
QE-2	69.76%	66.54%	66.34%	0.13%	98.68%	77.62%
QE-3	65.06%	61.25%	61.69%	0.14%	98.24%	74.18%
QE-4	63.88%	60.18%	60.58%	0.14%	98.24%	73.38%
QE-5	77.83%	73.51%	74.00%	0.09%	98.24%	82.98%
QE-6	73.19%	68.86%	69.35%	0.10%	98.24%	79.73%

6.7.3 Classification-based Approach

Query terms, which are extracted from a query *Q,* are expanded by using WordNet and the service ontology. Then, all query terms are sent to the query classifier in order to categorize the query into relevant service concepts. As mentioned in Section 6.5.3 , this thesis applies artificial neural networks and machine learning algorithms for creating the service classifier.

In this thesis, similar to the semantic service annotation in Chapter 5, the experiments apply the transport service ontology (Dong, Hussain & Chang 2011) which consists of 261 actual service concepts. To evaluate the performance of the proposed service querying approaches, the querying dataset is generated from all descriptions of the service concepts in the service ontology. Based on the service querying type, including single-label, multi-label querying type, the dataset is divided into single-label and multi-label querying dataset. The single-label querying dataset is applied for testing the performance of the classification-based approaches for single-label service querying, on the other hand, the multi-label querying dataset is applied for testing the proposed approaches for multi-label and combine-label service querying.

The single-label querying dataset consists of 2389 query samples, while the multi-label querying dataset comprises 2504 query samples. To classify the queries into relevant service concepts, supervised learning algorithms in artificial neural networks and machine learning are applied and they require the input and target data for training the machine. In this experiment, the query samples are divided into three sub-dataset; 70% for training, 15% for validation and 15% for testing. That is, 2389 samples of the single-label dataset are divided into 1672, 358 and 359 samples, while 2504 samples of the multi-label dataset are divided into 1753, 375 and 376 samples for training, validation, and testing respectively.

To implement the neural networks and the machine learning algorithms for service querying, similar to the service annotation, the Neural Network Toolbox in Matlab is applied to model a multilayer feed-forward neural network (FF) and a radial basis function network (RBF). In addition, the common classification algorithms, namely, k-nearest neighbor (KNN), classification tree (CT) and support vector machine (SVM), in Matlab are also applied to solve the service querying problem.

The classification-based querying approach is divided into five categories; FF-based, RBF-based, KNN-based, CT-based and SVM-based querying approach. All approaches are applied for the multi-label service querying, while only FF-based, RFN-based, KNN-based and CT-based are applied for the single-label and combine-label service querying. To evaluate the classification-based approach for single-label, multi-label and combine-label service querying, the performance was tested by setting the number of hidden neurons of the FF-based approach from 10 to 90 with an increment of 10, the spread value of the RBF-based approach from 0.1 to 0.9 with an increment of 0.1, and the number of *k* of KNN-based approach from 1 to 10 with an increment of 1. In this section, the best variable value for each classification-based approach is selected and the performances of the classification-based querying approaches are presented as follows.

Single-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network

Based on setting the number of hidden neurons as 30, the performance of the FF-based approach for single-label service querying is presented in Table 6.10. The results showed that the FF-based approach with all query expansion methods was able to query all query samples. While the FF-based approach with QE-6 returned the best performance with 80.78% in precision and recall, the approach with no query expansion gave the worst performance with 59.33% in precision and recall.

Table 6.10 The experiments of FF-based approach for a single-label service querying, N=30

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	59.33%	59.33%	59.33%	0.16%	100.00%	71.51%
QE-1	77.99%	77.99%	77.99%	0.08%	100.00%	84.58%
QE-2	77.72%	77.72%	77.72%	0.09%	100.00%	84.39%
QE-3	69.64%	69.64%	69.64%	0.12%	100.00%	78.73%
QE-4	66.57%	66.57%	66.57%	0.13%	100.00%	76.58%
QE-5	80.50%	80.50%	80.50%	0.07%	100.00%	86.34%
QE-6	80.78%	80.78%	80.78%	0.07%	100.00%	86.53%

2) Neural Network-based Approach - Radial Basis Function Network

Based on setting the spread value as 0.4, the performance of the RBF-based approach for single-label service querying is shown in Table 6.11. Similar to the FF-based approach, the RBF-based approach with QE-5 and QE-6 returned better results than the approach with other query expansion methods, while the approach with no query expansion method still returned the worst performance with 45.96% in precision and recall. However, the best precision and recall value of the RBF-based approach was around 4% less than the FF-based approach. Like the FF-based approach, the RBF-based approach was able to query all query samples.

Table 6.11 The experiments of RBF-based approach for a single-label service querying, S=0.4

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	45.96%	45.96%	45.96%	0.21%	100.00%	62.14%
QE-1	64.90%	64.90%	64.90%	0.13%	100.00%	75.41%
QE-2	57.94%	57.94%	57.94%	0.16%	100.00%	70.53%
QE-3	66.02%	66.02%	66.02%	0.13%	100.00%	76.19%
QE-4	66.30%	66.30%	66.30%	0.13%	100.00%	76.39%
QE-5	75.77%	75.77%	75.77%	0.09%	100.00%	83.02%
QE-6	76.88%	76.88%	76.88%	0.09%	100.00%	83.80%

3) Machine Learning-based Approach - K-Nearest Neighbor Algorithm

Based on setting the number of K as 1, the performance of the KNN-based approach for single-label service querying is displayed in Table 6.12. Like the FF-based and the RBF-based approach, the KNN-based approach with QE-0 returned the worst performance with 42.06% in precision and recall, while the KNN-based approach with QE-5 and QE-6 gave the best result with more than 72% in precision and recall. By comparing with the Neural Network (NN)-based approaches; both FF-based and RBF-based approach, the best precision and recall value of the KNN-based approach were 3-7% less than those of the NN-based approach.

Table 6.12 The experiments of KNN-based approach for a single-label service querying, K=1

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	42.06%	42.06%	42.06%	0.22%	100.00%	59.41%
QE-1	64.07%	64.07%	64.07%	0.14%	100.00%	74.83%
QE-2	56.55%	56.55%	56.55%	0.17%	100.00%	69.56%
QE-3	65.18%	65.18%	65.18%	0.13%	100.00%	75.61%
QE-4	62.95%	62.95%	62.95%	0.14%	100.00%	74.05%
QE-5	72.98%	72.98%	72.98%	0.10%	100.00%	81.07%
QE-6	73.82%	73.82%	73.82%	0.10%	100.00%	81.66%

4) Machine Learning-based Approach - Classification Tree Algorithm

The performance of the CT-based approach for the single-label service querying is shown in Table 6.13. The precision and recall values of the CT-based approach with QE-5 and QE-6 were better than those of the approach with other expansion methods. By comparing with other classification-based approaches for single-label service querying, the best performance with 72.98% in precision and recall of this approach was the worst. However, the precision and recall of the CT-based approach with QE-0 were approximately 10% greater than the RBF-based and the KNN-based approach with QE-0.

Table 6.13 The experiments of CT-based approach for a single-label service querying

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	51.53%	51.53%	51.53%	0.19%	100.00%	66.04%
QE-1	61.00%	61.00%	61.00%	0.15%	100.00%	72.68%
QE-2	59.33%	59.33%	59.33%	0.16%	100.00%	71.51%
QE-3	64.90%	64.90%	64.90%	0.13%	100.00%	75.41%
QE-4	69.36%	69.36%	69.36%	0.12%	100.00%	78.53%
QE-5	71.59%	71.59%	71.59%	0.11%	100.00%	80.10%
QE-6	72.98%	72.98%	72.98%	0.10%	100.00%	81.07%

Multi-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network

Based on setting the number of hidden neurons as 30, the performance of the FF-based approach for multi-label service querying is presented in Table 6.14. The overall performances were high and greater than 85% in both precision and recall. The FF-based approach with QE-1 performed the best with 95.22% in precision and 94.12% in recall; however, the querying rate was less than 50%. Although applying QE-6 returned a slightly lesser performance with 93.89% in precision and 93.27% in recall, the approach was able to query more than 60% query samples.

Table 6.14 The experiments of FF-based approach for a multi-label service querying, N=30

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	86.46%	85.54%	84.59%	0.06%	38.30%	80.84%
QE-1	95.22%	94.12%	93.43%	0.03%	45.21%	88.00%
QE-2	93.80%	93.91%	92.41%	0.06%	45.74%	87.32%
QE-3	94.02%	91.96%	91.63%	0.04%	52.66%	88.04%
QE-4	91.12%	89.47%	88.98%	0.05%	54.79%	86.43%
QE-5	93.44%	92.52%	91.86%	0.04%	58.78%	88.84%
QE-6	93.89%	93.27%	91.99%	0.06%	61.17%	89.51%

2) Neural Network-based Approach - Radial Basis Function Network

Based on setting the spread value as 0.9, the performance of the RBF-based approach for multi-label service querying is shown in Table 6.15. The RBF-based approach with QE-0 was unable to query any query sample term, while the approach with QE-5 was able to query around 50% of query samples. Furthermore, the approach with QE-5 returned a very good performance with 96.30% in precision and 92.22% in recall. Although the RBF-based approach with QE-1, QE-2 and QE-4 gave 100% in both precision and recall, the querying rates were too low and less than 12%. By comparing with the FF-based approach, the precision and recall values of the RBF-based approach were quite a lot better, but the querying rate were much worse.

Table 6.15 The experiments of RBF-based approach for a multi-label service querying, S=0.9

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	-	-	-	-	0.00%	-
QE-1	100.00%	100.00%	100.00%	0.00%	0.27%	85.04%
QE-2	100.00%	100.00%	100.00%	0.00%	0.27%	85.04%
QE-3	93.06%	88.83%	89.13%	0.06%	38.03%	84.61%
QE-4	100.00%	100.00%	100.00%	0.00%	11.70%	86.76%
QE-5	96.30%	92.22%	92.99%	0.01%	50.27%	88.84%
QE-6	98.46%	97.69%	97.95%	0.01%	17.29%	86.32%

3) Machine Learning-based Approach - K-Nearest Neighbor

Table 6.16 The experiments of KNN-based approach for a multi-label service querying, K=9

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	71.43%	67.78%	68.30%	0.12%	16.76%	66.48%
QE-1	74.03%	76.16%	73.78%	0.12%	31.91%	71.87%
QE-2	74.53%	77.48%	73.34%	0.22%	24.73%	71.11%
QE-3	80.54%	81.54%	79.26%	0.12%	45.21%	78.10%
QE-4	85.31%	84.26%	83.38%	0.09%	48.14%	81.48%
QE-5	86.55%	88.31%	85.55%	0.09%	50.27%	83.22%
QE-6	88.45%	87.45%	86.28%	0.08%	61.17%	85.61%

Based on setting the number of K as 9, the performance of the KNN-based approach for multi-label service querying is presented in Table 6.16. The KNN-based approach with QE-6 returned the best performance with 88.45% in precision, 87.45% in recall and 61.17% in querying rate, while the approach with QE-0 gave the worst result with 71.43% in precision, 67.78% in recall and 16.76% in querying rate. Furthermore, the performance in precision and recall of the KNN-based approach was 5-10% less than the best result of the NN-based querying approaches.

4) Machine Learning-based Approach - Classification Tree

The performance of the CT-based approach for multi-label service querying is displayed in Table 6.17. The CT-based approach with QE-5 returned the best performance in all factors with 87.21% in precision, 90.99% in recall, and 83.51% in querying rate, while the results of the approach with QE-0 were the worst. By comparing with the best case of the KNN-based approach, although the precision of the CT-based approach with QE-5 was a little less than the KNN-based approach with QE-6, the performance in recall, and querying rate was quite a lot better. That is, the CT-based approach increased by around 20% of the querying rate from the KNN-based approach.

Table 6.17 The experiments of CT-based approach for a multi-label service querying

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	72.47%	77.56%	73.37%	0.16%	62.23%	75.94%
QE-1	79.37%	83.33%	80.14%	0.11%	76.60%	82.74%
QE-2	74.64%	81.05%	75.84%	0.17%	71.28%	79.06%
QE-3	82.84%	85.14%	82.72%	0.11%	76.86%	84.83%
QE-4	83.54%	84.47%	82.73%	0.10%	75.80%	84.85%
QE-5	87.21%	90.99%	87.85%	0.09%	83.51%	89.22%
QE-6	86.67%	88.95%	86.50%	0.10%	82.98%	88.42%

5) Machine Learning-based Approach - Support Vector Machine

The performance of the SVM-based approach for multi-label service querying is presented in Table 6.18. The SVM-based approach with QE-0 and QE-5 were quite similar with around 83% in precision, 89% in recall and 69% in querying rate. Although the SVM-based approach with QE-0 was able to query all query samples, its performance was quite poor with 19% in precision and 47% in recall. By comparing with the best case of other ML-based approaches, the precision value of the SVM-based approach was a bit less than the KNN-based and CT-based approach, while the recall value of the SVM-based approach was slightly better than the KNN-based approach. The querying rate of the SVM-based approach was approximately 10% less than the CT-based approach and 10% greater than the KNN-based approach.

Table 6.18 The experiments of SVM-based approach for a multi-label service querying

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	19.39%	46.64%	25.21%	3.99%	100.00%	47.94%
QE-1	83.43%	89.34%	84.10%	0.38%	68.35%	84.58%
QE-2	19.08%	56.63%	27.60%	1.19%	100.00%	50.09%
QE-3	69.91%	88.73%	71.84%	4.48%	81.65%	78.63%
QE-4	70.80%	90.46%	73.16%	4.10%	82.18%	79.58%
QE-5	84.15%	89.53%	84.54%	0.45%	69.95%	85.19%
QE-6	32.01%	68.09%	42.47%	0.83%	100.00%	59.26%

Combine-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network

Based on setting the number of hidden neurons as 30, the performance of the FF-based approach for combine-label service querying is shown in Table 6.19. The FF-based approach with every query expansion method was able to query all query samples. The performance of the approach with QE-0 was less than those of using other query expansion methods. This demonstrated that applying the query expansion methods assisted in improving the querying performances. The approach with QE-1 and QE5 returned similar results with 82% in precision and 80% in recall. That means, in this case, combining the results from WordNet and the ontology could not improve the performance of the service querying. However, the performance of the approach with QE-6 also gave a good result with 81% in precision and 79% in recall, while the approach with QE-2 and QE-4 returned approximately 70% in precision and recall. That is, combining the results by considering only the proper terms from WordNet and the service ontology was able to improve the performance of this approach.

Table 6.19 The experiments of FF-based approach for a combine-label service querying

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	67.69%	65.47%	65.28%	0.13%	100.00%	76.67%
QE-1	82.68%	80.38%	80.28%	0.07%	100.00%	87.16%
QE-2	75.89%	74.14%	73.66%	0.11%	100.00%	82.51%
QE-3	72.92%	70.73%	70.70%	0.11%	100.00%	80.36%
QE-4	71.73%	69.48%	69.36%	0.12%	100.00%	79.50%
QE-5	82.31%	80.92%	80.66%	0.08%	100.00%	87.15%
QE-6	81.10%	79.43%	78.82%	0.09%	100.00%	86.17%

2) Neural Network-based Approach - Radial Basis Function Network

Based on setting the spread value as 0.9, the performance of the RBF-based approach for combine-label service querying is presented in Table 6.20. The RBF-based approach with QE-5 returned the best performance with 75.80% with precision and 72.60% in recall, while the approach with QE-0 returned only approximately 30% in precision and recall. That is, expanding the query terms from

WordNet and the ontology much improved the performance of the RBF-based querying approach. However, by comparing with the FF-based approach for combine-label service querying, the overall results of the RBF-based approach were certainly lower. Similar to the FF-based approach, the RBF-based approach was also able to query all query samples.

Table 6.20 The experiments of RBF-based approach for a combine-label service querying

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	32.45%	30.60%	30.88%	0.26%	100.00%	52.16%
QE-1	56.91%	54.29%	54.67%	0.17%	100.00%	69.09%
QE-2	47.61%	45.00%	45.39%	0.20%	100.00%	62.57%
QE-3	63.32%	60.04%	60.31%	0.15%	100.00%	73.36%
QE-4	64.63%	61.15%	61.67%	0.14%	100.00%	74.25%
QE-5	75.80%	72.60%	73.11%	0.09%	100.00%	82.16%
QE-6	72.34%	68.62%	69.15%	0.11%	100.00%	79.59%

3) Machine Learning-based Approach - K-Nearest Neighbor

Based on setting the number of k as 9, the performance of the KNN-based approach for combine-label service querying is displayed in Table 6.21. The KNN-based approach with QE-6 returned the best performance with 66.34% in precision and 64.51% in recall, while the approach with QE-0 returned the worst performance with 36.97% in precision and 34.22% in recall. The performances of the KNN-based approach with other query expansion methods were in the range of 40%-60% in precision and recall. The approach was able to query all query samples. By comparing with the NN-based approach, the best performances of both FF-based and RBF-based approaches were at least 8% better than the performance of the KNN-based approach.

Table 6.21 The experiments of KNN-based approach for a combine-label service querying

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	36.97%	34.22%	34.57%	0.24%	100.00%	55.07%
QE-1	44.64%	43.96%	43.33%	0.22%	100.00%	60.92%
QE-2	40.24%	39.26%	38.46%	0.26%	100.00%	57.72%
QE-3	52.37%	51.42%	50.52%	0.21%	100.00%	66.21%
QE-4	56.49%	54.35%	54.09%	0.18%	100.00%	68.84%
QE-5	58.66%	58.14%	56.89%	0.18%	100.00%	70.69%
QE-6	66.34%	64.51%	63.88%	0.15%	100.00%	75.77%

4) Machine Learning-based Approach - Classification Tree

The performance of the CT-based approach for combine-label service querying is presented in Table 6.22. The CT-based approach with QE-5 returned the best result with 78.68% in precision and 81.38% in recall, while the approach with QE-0 gave the worst performance with 57.05% in precision and 58.40% in recall. The performance of the approach with other query expansion methods was in the range of 62%-77% in precision and 66%-80% in recall. This demonstrated that the query expansion methods assisted the CT-based querying approach to improve the performance by 20% maximum. Furthermore, the best performance of the CT-based approach was better than the RBF-based and KNN-based approach, but a little less than the performance of the FF-based approach. Similar to other classification-based approaches, the querying rates of the CT-based approach in all cases were 100%.

Table 6.22 The experiments of CT-based approach for a combine-label service querying

QE-Method	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
QE-0	57.07%	58.40%	55.99%	0.20%	100.00%	69.96%
QE-1	69.30%	71.88%	69.48%	0.15%	100.00%	78.90%
QE-2	62.51%	66.23%	62.64%	0.19%	100.00%	74.30%
QE-3	70.06%	71.11%	69.31%	0.15%	100.00%	79.06%
QE-4	71.30%	71.29%	70.04%	0.14%	100.00%	79.70%
QE-5	78.68%	81.38%	78.80%	0.11%	100.00%	85.48%
QE-6	77.77%	79.20%	77.21%	0.13%	100.00%	84.55%

6.8 Experiment Summary

Based on the experimental results in Section 6.7 , the best results of the proposed semantic based approaches for single-label, multi-label and combine-label service annotation are presented in Table 6.23, Table 6.24 and Table 6.25 respectively. Based on the experimental results, it can be summarized that:

1) Based on the results in all-measures, FF-based approach outperforms other proposed approaches in single-label, multi-label, and combine-label service querying. In contrast, the performance of the ECBR is worse than the others.
2) The performances of the vector-based and the classification-based approaches are much better than the performances of the ECBR approach in

all single-label, multi-label, and combine-label service querying. Moreover, the overall all-measures values of the classification-based approaches are a little better than the vector-based approaches. Although the precision, recall, and f-measure values of the vector-based approaches are quite high and close to the values of the classification-based approaches, the querying rates of the vector-based approaches are lower, especially in the multi-label querying. This demonstrates that the classification-based approaches are suitable for all single-label, multi-label, and combine-label service querying because they perform well in all performance measures.

3) Using the query expansion methods assists in the improvement of service querying performance because the best experimental results are returned from the approaches with a query expansion method.
4) Most of the best performances of the proposed approaches, except the FF-based approach for combine-label querying, expand queries by using the hybrid query expansion methods (QE-5 and QE-6). Consequently, it can be concluded that expanding the query by using both synonyms in a lexical dictionary and related terms from domain-specific ontology assists in improvement of service querying performance.

Table 6.23 The best results of semantic based approaches for single-label service querying

Approach	QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Querying Rate	All-measures
ECBR	QE-5	54.90%	54.90%	54.90%	0.17%	99.96%	68.40%
VSM	QE-5	77.54%	77.54%	77.54%	0.09%	98.41%	84.03%
EVSM	QE-5	77.33%	77.33%	77.33%	0.09%	98.41%	83.88%
FF	QE-6	80.78%	80.78%	80.78%	0.07%	100.00%	86.53%
RBF	QE-6	76.88%	76.88%	76.88%	0.09%	100.00%	83.80%
KNN	QE-6	73.82%	73.82%	73.82%	0.10%	100.00%	81.66%
CT	QE-6	72.98%	72.98%	72.98%	0.10%	100.00%	81.07%

Table 6.24 The best results of semantic based approaches for multi-label service querying

Approach	QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Querying Rate	All-measures
ECBR	QE-5	52.63%	72.19%	56.58%	0.35%	61.78%	64.58%
VSM	QE-5	95.00%	92.57%	91.66%	0.03%	2.00%	80.93%
EVSM	QE-5	95.28%	91.24%	90.50%	0.03%	2.12%	80.69%
FF	QE-6	93.89%	93.27%	91.99%	0.06%	61.17%	89.51%
RBF	QE-5	96.30%	92.22%	92.99%	0.01%	50.27%	88.84%
KNN	QE-6	88.45%	87.45%	86.28%	0.08%	61.17%	85.61%
CT	QE-5	87.21%	90.99%	87.85%	0.09%	83.51%	89.22%
SVM	QE-5	84.15%	89.53%	84.54%	0.45%	69.95%	85.19%

Table 6.25 The best results of semantic based approaches for combine-label service querying

Approach	QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Querying Rate	All-measures
ECBR	QE-5	57.42%	67.61%	58.10%	0.26%	99.96%	71.78%
VSM	QE-5	78.11%	73.80%	74.30%	0.08%	98.24%	83.18%
EVSM	QE-5	77.83%	73.51%	74.00%	0.09%	98.24%	82.98%
FF	QE-1	82.68%	80.38%	80.28%	0.07%	100.00%	87.16%
RBF	QE-5	75.80%	72.60%	73.11%	0.09%	100.00%	82.16%
KNN	QE-6	66.34%	64.51%	63.88%	0.15%	100.00%	75.77%
CT	QE-5	78.68%	81.38%	78.80%	0.11%	100.00%	85.48%

6.9 Conclusion

This chapter presents a semantic-based approach for querying services. Given a query *Q,* the querying approach returns a service concept or service concepts that are relevant to *Q*. Then, those relevant service concepts are used for retrieving services in the next step. Similar to the semantic service annotation in Chapter 5, three semantic querying approaches; ECBR, Vector-based and Classification-based querying approaches, are proposed. The principal workflows of those approaches are divided into three steps; term extraction, query expansion, and querying module. Given a query *Q,* the term extraction module extracts *Q* into query terms and feeds them to the query expansion module. Then, the query terms are enlarged by using either WordNet or the service ontology. The querying module receives those expanded query terms and then calculates relevance scores between the query terms and concept terms of every ontological service concept. Service concepts with high relevance score are then returned.

In addition, three querying types, including single-label, multi-label and combine-label querying, are proposed. The single-label querying returns only the most relevant service concept, while the multi-label querying is able to return multiple service concepts. The combine-label querying normally returns the same results as the multi-label approach, but it returns only the most relevant concept in the case that the result of multi-label approach is empty. Furthermore, a hybrid querying approach which combines the results from typical querying approaches with both WordNet-based and ontology-based query expansion methods is also proposed.

To evaluate the performance of the proposed service querying approaches, the precision, recall, f-measure, fallout rate, querying rate, and all-measures combination are applied. The results demonstrate that the classification-based querying approach performs better than other ECBR and Vector-based approaches. In addition, expanding query terms is able to improve the performance of the querying approaches.

Chapter 7 Semantic Service Retrieval

7.1 Introduction

In this chapter, semantic approaches are proposed for retrieving the business services based on the results of the service annotation and the service querying in Chapter 5 and Chapter 6 respectively. Given a query *Q,* relevant service metadata (SDEs) are semantically returned. For example, the retrieval approach receives a query "Vessel agent consultant" and then it retrieves a service of the provider "Ben Lexcen Marine Brokers" with the service description "Trawlers, Fishing Vessels, Charter, Power Boats and Cruises". The main process of the proposed retrieval approaches is to query the relevant service concepts and then retrieve services that are annotated to those concepts. To retrieve the relevant business services, the service annotation approaches in Chapter 5 are employed to annotate or link the service metadata to relating service concepts, while the service querying approaches in Chapter 6 are applied to obtain the relevant service concepts based on a given query.

The service retrieval approaches in this chapter are divided into two groups, namely Non-Fuzzy based approaches and Fuzzy-based approaches. The former approach basically returns all services that are annotated to the queried concepts, while the latter approach retrieves the services based on the fuzzy relatedness values between 1) a service metadata and a service concept, and 2) a service concept and a query. Similar to Chapter 5 and Chapter 6, both Non-Fuzzy and Fuzzy based retrieval approaches are further classified into three types; the ECBR, the vector-based and the classification-based approaches. The workflows of those three approaches are identical, but they apply different service annotations and querying concepts based on the technique being used. That is, the ECBR retrieval approach applies the results from the ECBR annotation and ECBR querying approaches. On the other hand, the vector-based retrieval approach employs the vector-based annotation and vector-based querying approaches, while the classification-based retrieval approach uses the classification-based annotation and classification-based querying approaches.

In the next section, the main workflow of the semantic service retrieval methodology is presented. Then, each step of the methodology is explained in details in the rest of the chapter.

7.2 Semantic Service Retrieval Methodology

In this section, the workflow of the semantic service retrieval methodology is displayed in Figure 7.1. Given a query *Q,* the semantic service retrieval approach is able to retrieve a service or services that are relevant to *Q*. The steps involved in the semantic service retrieval methodology are as follows:

1. **The service annotation module** annotates service metadata to relevant ontological service concepts by considering their provider name and descriptions. The detailed working of this module was described in Chapter 5.

2. **The service querying module** queries ontological service concepts that are relevant to a query *Q*. The explanation of this module was presented in Chapter 6.

3. **The service retrieval module** retrieves service metadata that is relevant to the query *Q* by considering the service annotation dataset from step 1 and the relevant concepts from step 2.

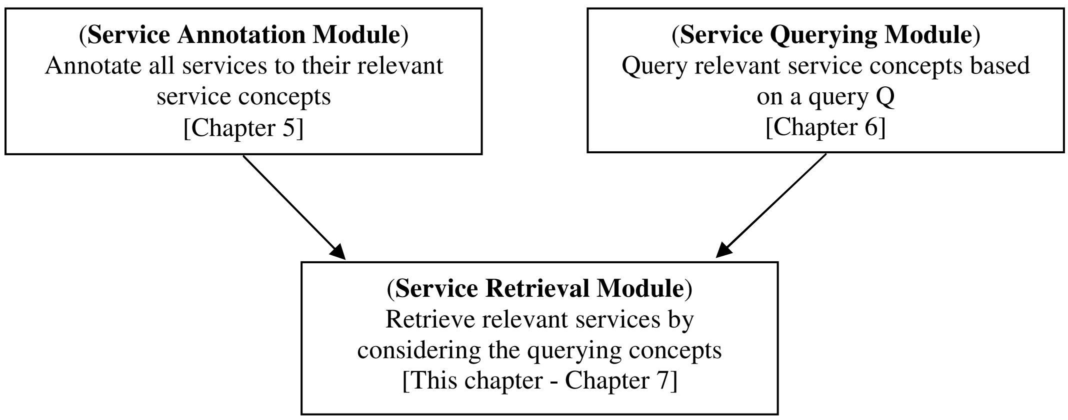


Figure 7.1 Flow Chart of the Semantic Service Retrieval Methodology

The semantic service retrieval methodology comprises three sub-modules, namely 1) *service annotation module*, 2) *service querying module* and 3) *service retrieval module*. The descriptions of the first two modules have been explained in Chapter 5 and Chapter 6. The output of the service annotation module is a set of the relations between the service metadata and the service concepts. For example, the service concept "Airline_Booking" is annotated to the service "Skywest Airlines", the service "Skytrans Airlines", the service "Escape Travel", and the service "GoStay". On the other hand, the output of the service querying module is a set of ontological concepts that relate to a given query. For instance, the query "Flight agent" is related to the concept "Airline_Agent" and the concept "Airline_Booking". Then, both the service annotation dataset and the queried concepts are sent to the service retrieval module for discovering the relevant services.

In this research, two main types of service retrieval approaches are proposed; 1) Non-Fuzzy based retrieval approach 2) Fuzzy based retrieval approach. The non-Fuzzy based approach simply returns all services that are linked to the queried concepts. That is, based on the previous example, at least four services; "Skywest Airlines", "Skytrans Airlines", "Escape Travel", and "GoStay", are retrieved from the query "Flight agent". In contrast, the Fuzzy based approach considers degrees of the annotation relevance and the querying relevance in order to compute the relevance value between the query and each service. That is, if the relevance score between the concept "Airline_Booking" and the service "Skywest Airlines" is greater than the score between "Airline_Booking" and "Skytrans Airlines", the service "Skywest Airlines" then has more possibility to be retrieved. The additional descriptions of the Non-Fuzzy based and Fuzzy based service retrieval approaches are mentioned in Section 7.3 and Section 7.4 respectively. Then, the experimental results of the proposed semantic service retrieval approaches are shown in Section 7.5 and Section 7.6 . The chapter is finally concluded in Section 7.7 .

7.3 Non-Fuzzy based Service Retrieval Approach

To discover relevant business services, the non-Fuzzy based service retrieval approach is proposed. The process of the non-Fuzzy based approach is shown in Figure 7.2. The approach receives the service annotation dataset (*SA*) and the queried service concepts (*QC*) from the service annotation module and the query module respectively. The *SA* consists of pairs of a service concept and a set of its relevant services, while the QC_Q is a set of service concepts that are relevant to the query Q. From Figure 7.2, the service ontology contains m service concepts (sc_1-sc_m). Each service concept sc_i is annotated to n_i services, where n_i is a number of annotated services of sc_i, and $s_{(i,j)}$ is the j-th service that relates to sc_i. Regarding the QC_Q, there exist k queried service concepts based on the query Q, where $sc_{(Q,i)}$ is the i-th service concepts that are queried from Q.

```
Program Non-Fuzzy based service retrieval approach
(SA : [sc1 : {s(1,1), s(1,2), ..., s(1,n1)},
       sc2 : {s(2,1), s(2,2), ..., s(2,n2)}, ...,
       scm : {s(m,1), s(1m,2), ..., s(m,nm)}],
 QCQ : {sc(Q,1), sc(Q,2), ..., sc(Q,k)},
 AT : Annotation threshold,
 QT : Query threshold)
   RSall = {}
   For each sc(Q,i) in QCQ
      If relevance(Q,sc(Q,i)) > QT
         RS(Q,i) = Get the annotated services; {s((Q,i),1), s((Q,i),2), ...,
s((Q,i),n(Q,i))},
                 of the service concept sc(Q,i) from SA
         For each sj in RS(Q,i)
            If relevance(sc(Q,i),sj) > AT
               Add sj in RSall
            End If
         End For
      End If
   End For
   Return RSall
End Non-Fuzzy based service retrieval approach
```

Figure 7.2 Pseudo-code of the Non-Fuzzy based service retrieval approach

Based on the service annotation and querying approaches in Chapter 5 and Chapter 6, the relevance values are applied to represent the relatedness among services, service concepts and queries. The relevance values in the service annotation are computed from the similarity between a service and a concept, while the service querying focuses on the relationships between a query and a concept. The relevance values in both cases are in the range of 0 and 1. To define whether a service is relevant to a service concept, *the annotation threshold* (*AT*) is applied, while *the querying threshold (QT)* is applied for defining the query-concept relationship. Given *SA* and QC_Q, the approach focuses on only the service concepts in which their relevance scores are greater than *QT*. Then, for each selected concept, the approach retrieves all annotated services that have relevance scores more than *AT*.

The example of the proposed non-Fuzzy based approach is shown in Figure 7.3. The query Q is "Flight agent", QC_Q is {"Airline_Agent", "Airline_Booking"}, and SA is ["Airline_Booking" : {"Skywest Airlines", "Skytrans Airlines", "Escape Travel", "GoStay"}, "Airline_Booking" : {"Skywest Airlines", "Skytrans Airlines", "Escape Travel", "GoStay"}]. The relevance values for service annotations and service querying are presented on their relationships. For instance, the querying relevance scores between Q and "Airline_Agent", and Q and "Airline_Booking" are 0.8 and 0.5 respectively. The approach retrieves the relevant services based on the parametric setting of AT and QT. That is, if AT and QT are set as 0.6, the approach will focus on only the concept "Airline_Agent" and then retrieves "Skywest Airlines" and "Skytrans Airlines" services. In contrast, if AT and QT are set as 0.45, the service "Skywest Airlines", "Skytrans Airlines", and "Escape Travel" are then discovered.

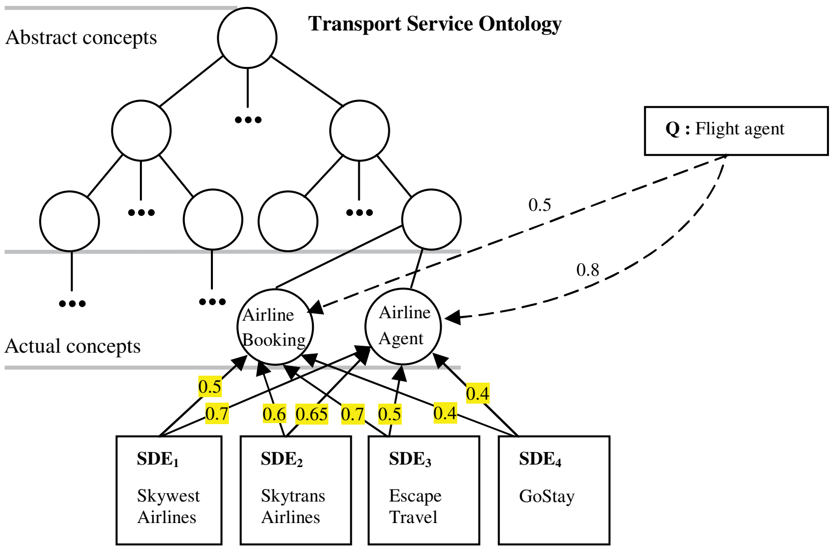


Figure 7.3 The example of the non-Fuzzy based service retrieval approach

7.4 Fuzzy based Service Retrieval Approach

Apart from the non-Fuzzy based approach for service retrieval, the Fuzzy-based approach is also proposed to retrieve business services. The process of the Fuzzy-based service retrieval approach is presented in Figure 7.4. Given Q, SA, and QC_Q, for each queried concept ($sc_{(Q,i)}$) and each annotated service (s_j), the approach gets the querying relevance score between Q and $sc_{(Q,i)}$, and the annotation relevance value between $sc_{(Q,i)}$ and s_j. While the non-Fuzzy approach defines the relevance scores as crisp values, the Fuzzy-based approach assigns them by using Fuzzy variables and membership functions. Then, the Fuzzy inference machine fires the Fuzzy rules to calculate the retrieval scores from the previous annotation and querying relevance scores. The service s_j will be retrieved if its retrieval score is greater than the retrieval threshold *(RT)*. The details of the Fuzzy retrieval variables and rules are described in Section 7.4.1 and Section 7.4.2 respectively.

```
Program Fuzzy based service retrieval approach
(SA : [sc1 : {s(1,1), s(1,2), ..., s(1,n1)},
       sc2 : {s(2,1), s(2,2), ..., s(2,n2)}, ...,
       scm : {s(m,1), s(1m,2), ..., s(m,nm)}],
 QCQ : {sc(Q,1), sc(Q,2), ..., sc(Q,k)},
 RT : Retrieval threshold
  RSall = {}
  For each sc(Q,i) in QCQ
    RS(Q,i) = Get the annotated services; {s((Q,i),1), s((Q,i),2), ...,
s((Q,i),n(Q,i))},
           of the service concept sc(Q,i) from SA
    For each sj in RS(Q,i)
      Retrieval_score(Q,sj) =
fuzzy_inference(relevance(Q,sc(Q,i)),relevance(sc(Q,i),sj))
      If Retrieval_score(Q,sj) > RT
        Add sj in RSall
      End If
    End For
  End For
  Return RSall
End Fuzzy based service retrieval approach
```

Figure 7.4 Pseudo-code of the Fuzzy based service retrieval approach

7.4.1 Fuzzy Service Retrieval Variables

The main idea of the Fuzzy based retrieval approach is to assign the relevance scores by using fuzzy logic theory (Klir & Yuan 1995; Ross 2009; Zadeh 1965). In this section, the annotation relevance scores, the querying relevance scores, and the retrieval scores are defined with three fuzzy variables; namely *annotaion_relevance*, *query_relevance*, and *retrieval_score*. Charts of those variables are displayed in Figure 7.5, Figure 7.6 and Figure 7.7 respectively. The values of each fuzzy variable are defined by several Gaussian-shaped membership functions. The *annotation_relevance* and *query_relevance* variables consist of three membership functions – low, medium and high, while the *retrieval_score* variable comprises five functions; low, low_medium, medium, medium_high and high. The x-axis of Figure 7.5, Figure 7.6 and Figure 7.7 represents the annotation relevance values, the querying relevance values, and the service retrieval scores respectively, while the y-axis shows the degrees of membership functions of the *annotation_relevance*, *query_relevance* and *retrieval_score* variables.

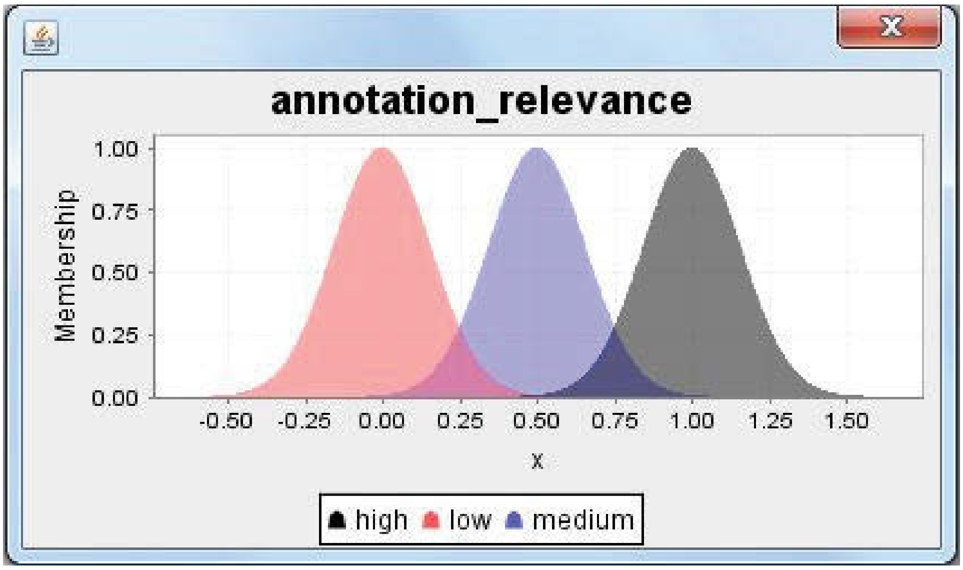


Figure 7.5 The membership functions of the *annotation_relevance* variable

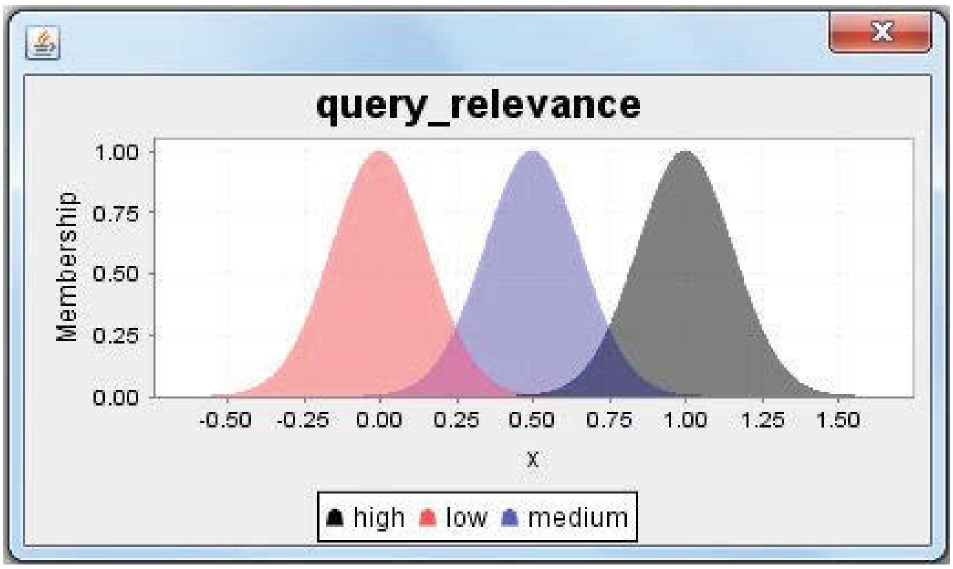


Figure 7.6 The membership functions of the *query_relevance* variable

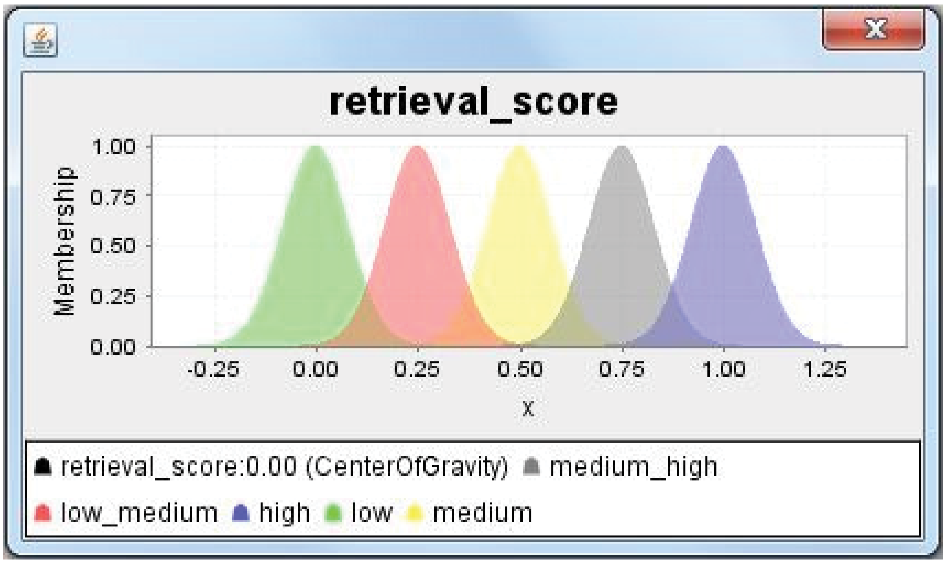


Figure 7.7 The membership functions of the *retrieval_score* variable

The fuzzy variable charts demonstrate that given values, i.e. the annotation relevance scores, are assigned to multiple fuzzy membership functions or fuzzy sets with different degrees. For example, the relevance score between a service and a concept is 0.3. Figure 7.5 shows that the score partially belongs to both low and medium fuzzy sets with the degrees of 0.25 and 0.5 respectively. In contrast, if the annotation relevance score is 0.9, it will totally belong to the high fuzzy set.

7.4.2 Fuzzy Service Retrieval Rules

Given the annotation value between the service *S* and the concept *C,* and querying value between the query *Q* and the concept *C,* the Fuzzy inference machine applies the fuzzy rules to calculate the retrieval score of the service *S.* If the retrieval score of *S* is greater than the retrieval threshold *(RT)*, the service *S* will be retrieved. In this work, nine fuzzy service retrieval rules are defined as follows.

RULE 1 : IF *annotation_relevance* IS low AND *query_relevance* IS low
THEN *retrieval_score* IS low_medium

RULE 2 : IF *annotation_relevance* IS low AND *query_relevance* IS medium
THEN *retrieval_score* IS medium

RULE 3 : IF *annotation_relevance* IS low AND *query_relevance* IS high

THEN *retrieval_score* is medium_high

RULE 4 : IF *annotation_relevance* IS medium AND *query_relevance* IS low

THEN *retrieval_score* IS medium

RULE 5 : IF *annotation_relevance* IS medium AND *query_relevance* IS medium

THEN *retrieval_score* IS medium_high

RULE 6 : IF *annotation_relevance* IS medium AND *query_relevance* IS high

THEN *retrieval_score* is high

RULE 7 : IF *annotation_relevance* IS high AND *query_relevance* IS low

THEN *retrieval_score* IS medium_high

RULE 8 : IF *annotation_relevance* IS high AND *query_relevance* IS medium

THEN *retrieval_score* IS high

RULE 9 : IF *annotation_relevance* IS high AND *query_relevance* IS high

THEN *retrieval_score* is high

While the input variables of the fuzzy rules are *annotation_relevance* and *query_relevance*, the output variable is *retrieval_score*. In this thesis, the *query_relevance* variable has more priority than the *annotation_relevance* variable. That is, services that relate to concepts with high query scores tend to be retrieved more than ones that are relevant to concepts with low query scores, although the annotation scores of those services are quite low. This is because the values of the annotation scores are generally less than one of the query scores. To calculate the relatedness, the length of the service descriptions and queries affects the relevance values. Given a service *S* or a query *Q,* the more content of *Q* or a service description of *S* semantically appears in concept descriptions of a concept *C,* the more *S* or *Q* are relevant to *C.* This leads the shorter-length queries to return higher relevance scores when comparing with the longer-length service descriptions.

7.4.3 Fuzzy Inference Example

The proposed approach applies the fuzzy inference machine to compute the retrieval score based on the fuzzy membership functions and fuzzy rules in Section 7.4.1 and Section 7.4.2 respectively. This section demonstrates how the fuzzy inference machine infers the retrieval score. In this thesis, the Mamdani-style fuzzy inference technique is applied. It consists of four primary steps – 1) Fuzzification, 2) Rule evaluation 3) Aggregation of the rule outputs and 4) Defuzzification (Negnevitsky 2005). Given an annotation relevance score and a querying relevance score, the machine firstly converts the given crisp inputs into the degrees of fuzzy membership functions (Step 1) and then evaluates the output of each fuzzy rule (Step 2). After that, the machine combines the fuzzified outputs of all rules and then converts them into the crisp service retrieval score (Step 3).

Based on the example in Figure 7.3, the retrieval scores of the service "Skywest Airlines" with the concept "Airline_Agent" and "Airline_Booking" are computed as shown in Figure 7.8 and Figure 7.9 respectively.

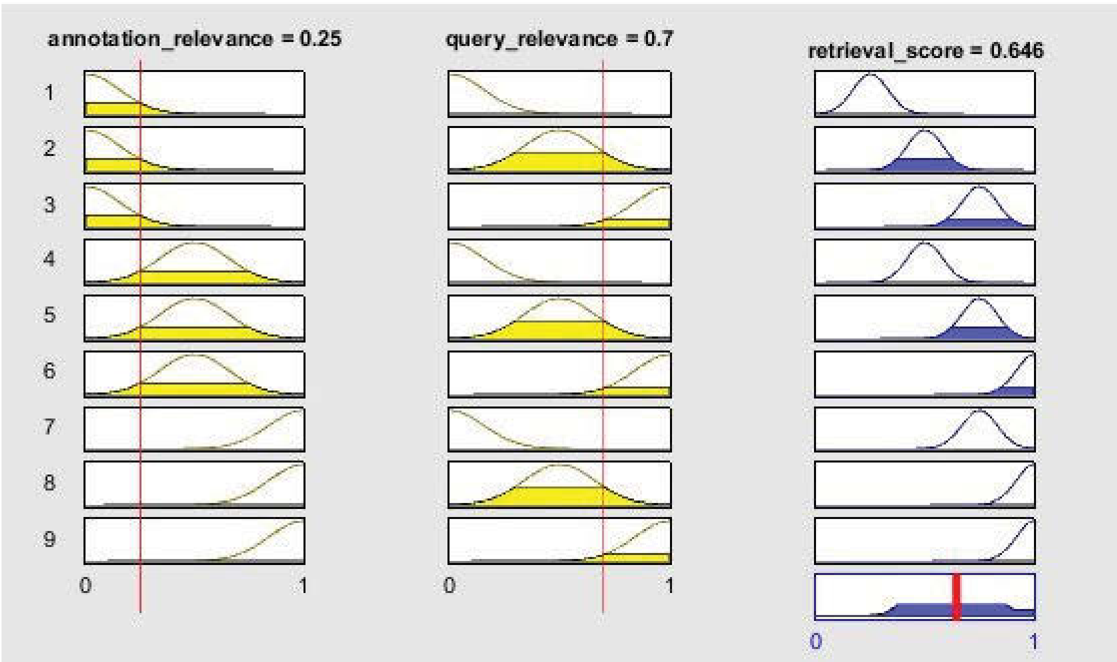


Figure 7.8 The retrieval score inference of the service “Skywest Airlines” with the concept “Airline_Agent”

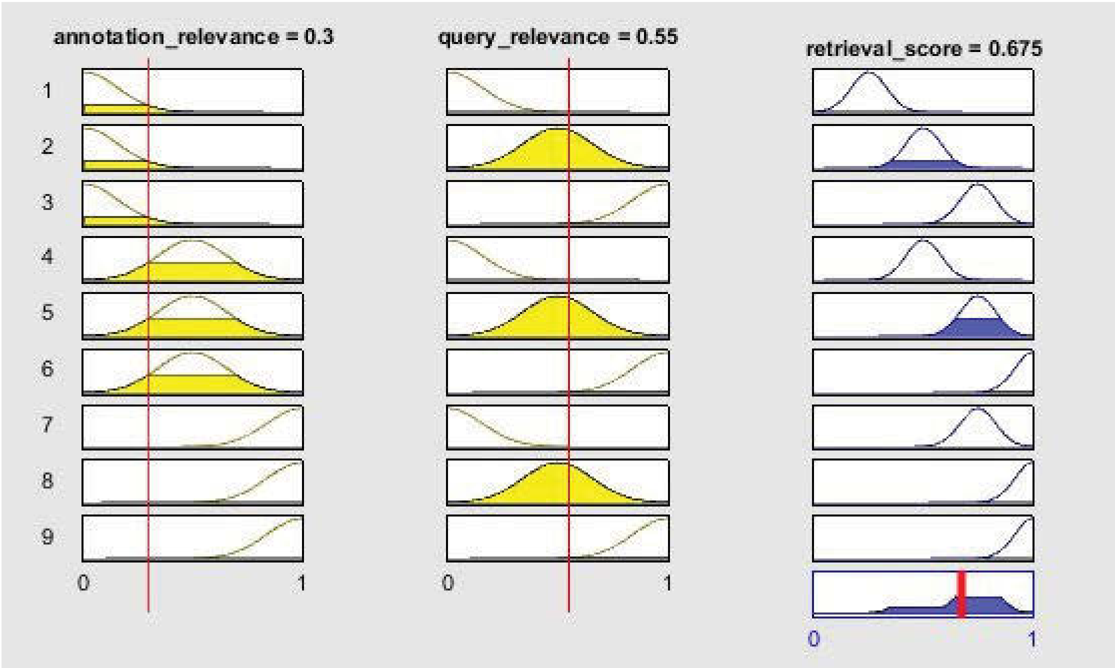


Figure 7.9 The retrieval score inference of the service “Skywest Airlines” with the concept “Airline_Booking”

7.5 Experiments

This section presents the experimental results of the proposed semantic service retrieval approaches; the non-fuzzy based and fuzzy based approaches, in Section 7.5.1 and Section 7.5.2 respectively. The service retrieval approaches apply the service annotation approach to link every service to relevant concepts, while the service querying approach is applied for discovering service concepts that are relevant to a given query. The annotation relevance scores and the query relevance scores are then applied for retrieving relevant services. The non-fuzzy based approaches consider that the annotation and query relevance scores are crisp, and retrieve all services that are annotated to the queried service concepts. In contrast, the fuzzy based approaches consider that the relevance scores are fuzzy, and retrieve services whose retrieval scores reach the retrieval threshold.

Based on the service annotation and service querying approaches, both non-fuzzy based and fuzzy based service retrieval approaches are divided into ECBR, Vector-based and Classification-based approaches. Each approach is categorized into three sub-categories, namely single-label, multi-label and combine-label retrieval. The single-label retrieval approaches apply the single-label annotation and single-label querying approaches, while the multi-label annotation and querying approaches and combine-label annotation and querying approaches are applied for the multi-label and combine-label retrieval approaches respectively.

To evaluate the performance of the proposed service retrieval approaches, similar to the service querying, the average precision, average recall, average f-measure, average fallout-rate, retrieval rate (percentage of hitting query samples), and all-measures combination, are also applied. The value of the all-measures combination is calculated as follows:

$$All\_measures = (0.4 * avg\_precision) + (0.15 * avg\_recall) + (0.15 * avg\_fmeasure) + (0.15 * (1 - avg\_fallout)) + (0.15 * retrieval\_rate)$$

In addition, the results are presented based on the proposed query expansion methods in Chapter 6. In this work, there are seven query expansion methods; 1) No query expansion method (QE-0), 2) WordNet-based method with All-senses expansion technique (QE-1), 3) WordNet-based method with Proper-sense expansion technique (QE-2), 4) Ontology-based method with All-related terms expansion technique (QE-3), 5) Ontology-based method with Most-related terms expansion technique (QE-4) 6) Hybrid-based method with All-terms expansion technique (QE-5 : QE-1 + QE-3) and 7) Hybrid-based method with Proper-term expansion technique (QE-6 : QE-2 + QE-4).

7.5.1 Non-Fuzzy based Retrieval Approach

For each query expansion method, the non-fuzzy service retrieval approaches with all settings of annotation and querying variables are tested. This leads the number of the experimental cases to be extremely large. This section demonstrates the experiments based on the best settings of the annotation and querying variables in Chapter 5 and Chapter 6. The annotation and querying variable settings of all proposed service retrieval approaches are shown in Table 7.1, where *AT* is the annotation threshold, *QT* is the querying threshold, *N* is a number of hidden neurons in the Feed-Forward Neural Network, *S* is the spread value of the Radial Function Network and *K* is a number of *K* in K-Nearest Neighboring algorithm. During each test, the *AT* and *QT* parameters are set to values that give optimum result.

Table 7.1 The parameter settings of the non-fuzzy based semantic service retrieval approaches

Service Retrieval Approaches		Annotation Variables	Querying Variables
ECBR	Single-label	*AT*=0.05	*QT*=0.5
	Multi-label	*AT*=0.2	*QT*=0.8
	Combine-label	*AT*=0.3	*QT*=0.1
Vector-based	Single-label	VSM : *AT*=0.7 EVSM : *AT*=0.7	VSM : *QT*=0.4 EVSM : *QT*=0.4
	Multi-label	VSM : *AT*=0.7 EVSM : *AT*=0.1	VSM : *QT*=0.7 EVSM : *QT*=0.9
	Combine-label	VSM : *AT*=0.1 EVSM : *AT*=0.1	VSM : *QT*=0.6 EVSM : *QT*=0.9
Classification-based	Single-label	FF : *N*=80 RBF : *S*=0.9 KNN : *K*=9	FF : *N*=10 RBF : *S*=0.9 KNN : *K*=6
	Multi-label	FF : *N*=10 RBF : *S*=0.5 KNN : *K*=6	FF : *N*=10 RBF : *S*=0.9 KNN : *K*=9
	Combine-label	FF : *N*=10 RBF : *S*=0.5 KNN : *K*=6	FF : *N*=90 RBF : *S*=0.9 KNN : *K*=9

ECBR Retrieval Approach

Single-Label based Approach (AT=0.05, QT=0.5)

To evaluate the ECBR approach for the single-label non-fuzzy service retrieval, the performance of the method was tested by setting the annotation threshold *(AT)* and the querying threshold *(QT)* as 0.05 and 0.5 respectively. The performances of this approach are presented in Table 7.2. The single-label ECBR approach with QE-5 and QE-6 gave a good performance with approximately 30% in precision, 30-35% in recall and 42-44% in retrieval rate. The approach with the ontology-based query expansion; QE-3 and QE-4, was able to increase the retrieval rate and outperformed the approach with WordNet-based query expansion; QE-1 and QE-2. Moreover, using WordNet decreased a little the performances of the single-label ECBR approach.

Table 7.2 The experiments of ECBR approach for a single-label non-fuzzy service retrieval (*AT*=0.05, *QT*=0.5)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	20.28%	24.13%	15.31%	0.29%	25.00%	32.73%
QE-1	17.98%	22.47%	14.01%	0.27%	29.17%	32.00%
QE-2	17.98%	22.47%	14.01%	0.29%	29.17%	31.99%
QE-3	21.51%	24.82%	14.77%	0.20%	43.75%	36.07%
QE-4	26.52%	33.53%	21.46%	0.18%	47.92%	41.02%
QE-5	31.33%	31.06%	23.51%	0.20%	41.67%	41.94%
QE-6	31.43%	34.34%	24.77%	0.19%	43.75%	42.97%

Multi-Label based Approach (AT=0.2, QT=0.8)

To evaluate the ECBR approach for the multi-label non-fuzzy service retrieval, the performance was tested by setting *AT* and *QT* as 0.2 and 0.8 respectively. The performances of this approach are presented in Table 7.3. The approach with hybrid query expansion gave the best precision value; 65.45%, which was much better than applying other expansion methods. In contrast, applying the ontology-based query expansion methods gave better recall values and retrieval rates than the others.

Table 7.3 The experiments of ECBR approach for a multi-label non-fuzzy service retrieval (*AT*=0.2, *QT*=0.8)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	42.42%	7.20%	11.27%	0.14%	4.84%	35.45%
QE-1	30.45%	10.98%	12.48%	0.13%	8.06%	31.89%
QE-2	31.82%	5.40%	8.45%	0.14%	6.45%	30.75%
QE-3	32.95%	15.76%	11.82%	0.12%	25.81%	36.17%
QE-4	38.00%	18.96%	13.79%	0.06%	20.97%	38.25%
QE-5	65.45%	8.53%	14.38%	0.08%	8.06%	45.82%
QE-6	65.45%	8.53%	14.38%	0.08%	8.06%	45.82%

Combine-Label based Approach (AT=0.3, QT=0.1)

To evaluate the ECBR approach for the combine-label non-fuzzy service retrieval, the performance was tested by setting *AT* and *QT* as 0.3 and 0.1 respectively. The performances of this approach are presented in Table 7.4. The overall precision values of this approach with every query expansion method were quite close to each other. The approach with QE-5 gave the best performance with 15.67% in precision, 51.70% in recall and 98.39% in retrieval rate. In addition, using query expansion methods; QE-1 to QE-6, were able to assist in recall and retrieval rate improvement. Overall, applying the query expansion methods assisted in improving the performance of the retrieval approach when comparing with applying a non-query expansion method.

Table 7.4 The experiments of ECBR approach for a combine-label non-fuzzy service retrieval (*AT*=0.3, *QT*=0.1)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	16.17%	39.41%	17.02%	0.68%	88.71%	43.14%
QE-1	15.16%	40.83%	16.51%	0.86%	96.77%	44.05%
QE-2	15.35%	40.63%	16.55%	0.82%	90.32%	43.14%
QE-3	15.03%	51.35%	16.99%	1.24%	93.55%	45.11%
QE-4	17.08%	48.85%	18.38%	0.90%	93.55%	45.81%
QE-5	15.67%	51.70%	17.81%	1.31%	98.39%	46.26%
QE-6	17.08%	51.09%	18.92%	0.97%	91.94%	45.98%

Vector-based Retrieval Approach

Single-Label based Approach (AT=0.7, QT=0.4)

To evaluate the Vector-based approach for the single-label non-fuzzy service retrieval, the performance was tested by setting *AT* and *QT* as 0.7 and 0.4 respectively. The performances of this approach by using VSM and EVSM vectors are presented in Table 7.5 and Table 7.6 respectively.

Table 7.5 The experiments of Vector (VSM) approach for a single-label non-fuzzy service retrieval (*AT*=0.7, *QT*=0.4)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
QE-1	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
QE-2	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
QE-3	33.33%	3.51%	6.35%	0.02%	6.25%	30.75%
QE-4	45.00%	12.87%	11.96%	0.03%	10.42%	38.28%
QE-5	41.67%	20.18%	17.46%	0.05%	6.25%	38.24%
QE-6	56.25%	16.09%	14.95%	0.03%	8.33%	43.40%

The performances of the non-fuzzy VSM-based approach in Table 7.5 show that the approach with all query expansion methods could not improve the performance of single-label service retrieval. Although QE-1 to QE-6 increased the retrieval rate values, the precision and recall values were decreased. In addition, the approach without query expansion gave 62.5% in precision and 30.26% in recall, but it was able to retrieve services only around 4%. In this case, it can conclude that the non-fuzzy single-label VSM based approach is unsuitable for retrieving services.

The performances of the non-fuzzy EVSM-based approach in Table 7.6 were quite similar with 62.50%, 30.26% and 4.17% in precision, recall, and retrieval rates respectively. Only the approach with QE-3 gave 100% in precision; however, recall and retrieval rate were reduced to only 10.53% and 2.08%. Comparing with the VSM-based approach, the performances of EVSM-based approach with query expansion methods were better than VSM-based approach with query expansion.

Table 7.6 The experiments of Vector (EVSM) approach for single-label non-fuzzy service retrieval (*AT*=0.7, *QT*=0.4)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
QE-1	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
QE-2	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
QE-3	100.00%	10.53%	19.05%	0.00%	2.08%	59.75%
QE-4	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
QE-5	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
QE-6	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%

Multi-Label based Approach (LT=0.5, QT=0.9)

To evaluate the Vector-based approach for the multi-label non-fuzzy service retrieval, the performance was tested by setting *AT* and *QT* as 0.7 and 0.7 respectively. The performances of this approach by using VSM and EVSM vectors are presented in Table 7.7 and Table 7.8 respectively.

The experimental results of VSM-based approach in Table 7.7 demonstrate that the approach without query expansion outperforms using most of the query expansion methods. Only QE-1 performed 100% in precision; however, its recall and retrieval rate values were low with 10.53% and 1.61% respectively. In addition, overall retrieval rate values of this approach were low; 1-18% and QE-5 was unable to retrieve any service.

Table 7.7 The experiments of Vector (VSM) approach for a multi-label non-fuzzy service retrieval (*AT*=0.7, *QT*=0.7)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	56.25%	17.76%	17.86%	0.03%	6.45%	43.81%
QE-1	100.00%	10.53%	19.05%	0.00%	1.61%	59.68%
QE-2	50.00%	5.26%	9.52%	0.02%	3.23%	37.70%
QE-3	11.11%	33.33%	16.67%	0.11%	4.84%	27.65%
QE-4	35.61%	24.99%	16.26%	0.05%	17.74%	38.08%
QE-5	-	-	-	-	0.00%	-
QE-6	41.67%	20.18%	17.46%	0.05%	4.84%	38.03%

The experimental results of EVSM-based approach in Table 7.8 showed that the approach with hybrid-based query expansion gave the best performance with 82.14% in precision and 52.27% in recall; however, retrieval rate was just 1.61%. Furthermore, using WordNet-based query expansion methods (QE-1 and QE-2) also assisted in much recall value improvement. Like VSM-based approach, retrieval rates of EVSM-based approach were low with 1-21%.

Table 7.8 The experiments of Vector (EVSM) approach for a multi-label non-fuzzy service retrieval (*AT*=0.1, *QT*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	53.57%	32.39%	40.28%	0.24%	3.23%	47.78%
QE-1	14.72%	69.50%	17.53%	1.35%	20.97%	36.89%
QE-2	22.40%	88.07%	19.59%	1.53%	6.45%	40.85%
QE-3	41.07%	26.14%	31.94%	0.19%	3.23%	40.60%
QE-4	40.51%	48.53%	41.39%	0.60%	9.68%	46.05%
QE-5	82.14%	52.27%	63.89%	0.17%	1.61%	65.50%
QE-6	82.14%	52.27%	63.89%	0.17%	1.61%	65.50%

Combine-Label based Approach (LT=0.6, QT=0.9)

To evaluate the Vector-based approach for the combine-label non-fuzzy service retrieval, the performance was tested by setting *AT* and *QT* as 0.1 and 0.6 respectively. The performances of this approach by using VSM and EVSM vectors are presented in Table 7.9 and Table 7.10 respectively.

The overall performances of VSM-based approaches were quite close to each other. The approach with QE-6 gave a little better than the others with 23.13%, 58.66% and 95.16% in precision, recall, and retrieval rate respectively. Like VSM-based approach, applying both WordNet and ontology-based query expansion (QE-5) gave the best performance in EVSM-based approach. It seems that recall and retrieval rate values tend to be increased by the EVSM-based approach.

Table 7.9 The experiments of Vector (VSM) approach for a combine-label non-fuzzy service retrieval (*AT*=0.1, *QT*=0.6)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	18.43%	58.15%	21.24%	1.02%	96.77%	48.64%
QE-1	20.33%	56.70%	23.75%	0.76%	96.77%	49.60%
QE-2	20.46%	57.66%	23.45%	0.84%	95.16%	49.50%
QE-3	21.73%	55.59%	22.97%	0.94%	96.77%	49.85%
QE-4	18.81%	56.87%	21.28%	1.06%	95.16%	48.36%
QE-5	20.92%	54.84%	23.82%	0.85%	96.77%	49.56%
QE-6	23.13%	58.66%	25.02%	0.90%	95.16%	50.94%

Table 7.10 The experiments of Vector (EVSM) approach for a combine-label non-fuzzy service retrieval (*AT*=0.1, *QT*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	20.28%	54.87%	23.00%	0.87%	91.94%	48.45%
QE-1	18.47%	57.39%	21.75%	0.88%	96.77%	48.64%
QE-2	21.69%	59.71%	23.77%	0.87%	93.55%	50.10%
QE-3	21.05%	56.53%	23.96%	0.91%	95.16%	49.63%
QE-4	20.70%	55.93%	23.43%	0.86%	91.94%	48.84%
QE-5	22.18%	60.10%	25.31%	0.87%	96.77%	51.07%
QE-6	20.62%	58.26%	23.46%	0.93%	93.55%	49.40%

Classification-based Retrieval Approach

The proposed classification-based retrieval approaches apply learning algorithms for neural networks and machine learning in order to classify queries and services into existing service concepts. Thus, the performances of the approaches are demonstrated based on not only label types (single-label, multi-label and combine-label), but also the classification techniques; namely, the feed-forward neural network (FF), the radial basis function network (RBF), the K-nearest neighbor (KNN) and the classification tree (CT).

Single-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network (*AN*=80, *QN*=10)

To evaluate the FF-based approach for the single-label non-fuzzy service retrieval, the performance was tested by setting the annotating hidden neurons (*AN*) and the querying hidden neurons (*QN*) as 80 and 10 respectively.

The performances of this approach in Table 7.11 demonstrated that the approach without query expansion outperformed the approach with query expansion. In this case, it was obvious that precision and recall values of this approach were high, 60-80%; meanwhile, retrieval rate values were low, less than 10%.

Table 7.11 The experiments of FF-based approach for a single-label non-fuzzy service retrieval (*AN*=80, *QN*=10)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	79.17%	50.00%	61.29%	0.17%	2.08%	63.65%
QE-1	61.81%	40.79%	49.11%	0.34%	4.17%	53.78%
QE-2	60.74%	48.99%	53.34%	0.37%	6.25%	55.53%
QE-3	61.81%	40.79%	49.11%	0.34%	4.17%	53.78%
QE-4	61.81%	40.79%	49.11%	0.34%	4.17%	53.78%
QE-5	61.81%	40.79%	49.11%	0.34%	4.17%	53.78%
QE-6	61.81%	40.79%	49.11%	0.34%	4.17%	53.78%

2) Neural Network-based Approach - Radial Basis Function Network (*AS*=0.6, *QS*=0.4)

To evaluate the RBF-based approach for the single-label non-fuzzy service retrieval, the performance was tested by setting the annotating spread value (*AS*) and the querying spread value (*QS*) as 0.9 and 0.9 respectively. The performances of this approach are presented in Table 7.12. The approach with QE-3 to QE-6 gave good precision values, meanwhile applying QE-0, QE-1 and QE-2 could not retrieve any testing service. Although precision values of this approach reached 93.75%, it could retrieve only 2.08% of testing services.

Table 7.12 The experiments of RBF-based approach for a single-label non-fuzzy service retrieval (*AS*=0.9, *QS*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	-	-	-	-	0.00%	-
QE-1	-	-	-	-	0.00%	-
QE-2	-	-	-	-	0.00%	-
QE-3	64.58%	17.32%	25.93%	0.03%	6.25%	48.25%
QE-4	93.75%	39.47%	55.56%	0.03%	2.08%	67.06%
QE-5	73.44%	16.28%	25.26%	0.03%	8.33%	51.85%
QE-6	93.75%	39.47%	55.56%	0.03%	2.08%	67.06%

3) Machine Learning-based Approach - K-Nearest Neighbor Algorithm (*AK*=9, *QK*=6)

To evaluate the KNN-based approach for the single-label non-fuzzy service retrieval, the performance was tested by setting the annotating *k* value (*AK*) and the querying *k* value (*QK*) as 9 and 6 respectively. The performances of this approach are presented in Table 7.13. The approach with QE-6 returned the best performance with 63.61% in precision, 37.85% in recall and 14.58% in retrieval rate. Similar to the single-label RBF-based approach, expanding the queries resulted in much improved performances of the approach.

Table 7.13 The experiments of KNN-based approach for a single-label non-fuzzy service retrieval (*AK*=9, *QK*=6)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	16.27%	31.58%	14.35%	0.24%	14.58%	30.55%
QE-1	27.41%	35.32%	16.36%	0.45%	16.67%	36.15%
QE-2	8.23%	40.00%	13.56%	0.26%	10.42%	27.85%
QE-3	33.12%	28.85%	21.26%	0.32%	12.50%	37.59%
QE-4	38.36%	26.88%	24.53%	0.26%	18.75%	40.83%
QE-5	44.53%	26.50%	25.82%	0.34%	20.83%	43.73%
QE-6	63.61%	37.85%	36.89%	0.20%	14.58%	53.81%

4) Machine Learning-based Approach - Classification Tree Algorithm

The performances of the CT-based approach for the single-label non-fuzzy service retrieval are presented in Table 7.14. Comparing with single-label RBF-based and KNN-based approaches, the CT-based approaches returned lower performances with 7%-45% in precision and 2%-30% in recall. The approach with QE-3 gave the best performance with 44.65% in precision, 20.14% in recall and 14.58% in retrieval rate. In addition, applying hybrid query expansion (QE-5 and QE-6) was able to increase recall and retrieval rate values; however, their precision values were much less than QE-3. Like other single-label non-fuzzy classification-based approaches, the performances of the CT-based approach with using query expansion methods were much better than the approach without any query expansion.

Table 7.14 The experiments of CT-based approach for a single-label non-fuzzy service retrieval

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	7.29%	2.30%	3.50%	0.17%	16.67%	21.26%
QE-1	17.48%	8.03%	10.83%	0.39%	16.67%	27.26%
QE-2	11.67%	3.68%	5.60%	0.18%	10.42%	22.59%
QE-3	44.65%	20.14%	27.35%	0.14%	14.58%	42.15%
QE-4	34.73%	15.66%	21.27%	0.13%	18.75%	37.22%
QE-5	29.19%	27.97%	18.86%	0.49%	22.92%	37.07%
QE-6	32.11%	30.76%	20.75%	0.33%	20.83%	38.65%

Multi-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network (*AN*=10, *QN*=10)

To evaluate the FF-based approach for the multi-label non-fuzzy service retrieval, the performance was tested by setting the annotating hidden neurons (*AN*) and the querying hidden neurons (*QN*) as 10 and 10 respectively. The performances of this approach are presented in Table 7.15. In general, the precision and recall values of this approach with any query expansion method were high. Most of them were around 92% in precision and 78% in recall. The approach with QE-5 returned better performance with 96.88% and 81.58% in precision and recall respectively. In addition, there was no difference between the approach with and without the query expansion method. Although precision values of this approach were high, retrieval rate were less than 10%.

Table 7.15 The experiments of FF-based approach for a multi-label non-fuzzy service retrieval (*AN*=10, *QN*=10)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	92.61%	78.36%	84.89%	0.19%	8.06%	77.71%
QE-1	91.54%	77.55%	83.97%	0.23%	6.45%	76.78%
QE-2	91.54%	77.55%	83.97%	0.23%	6.45%	76.78%
QE-3	91.54%	77.55%	83.97%	0.23%	6.45%	76.78%
QE-4	91.54%	77.55%	83.97%	0.23%	6.45%	76.78%
QE-5	96.88%	81.58%	88.57%	0.03%	3.23%	79.75%
QE-6	92.61%	78.36%	84.89%	0.19%	8.06%	77.71%

2) Neural Network-based Approach - Radial Basis Function Network (*AS*=0.5, *QS*=0.9)

To evaluate the RBF-based approach for the multi-label non-fuzzy service retrieval, the performance was tested by setting the annotating spread value (*AS*) and the querying spread value (*QS*) as 0.5 and 0.9 respectively. The performances of this approach are presented in Table 7.16.

The approach with QE-6 gave the best performance with 100% in precision and 84.62% in recall. However, it was able to retrieve only 1.61% of testing services. In addition, applying QE-0, QE-1, QE-2 and QE-4 were unable to retrieve any testing service.

Table 7.16 The experiments of RBF-based approach for a multi-label non-fuzzy service retrieval (*AS*=0.5, *QS*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	-	-	-	-	0.00%	-
QE-1	-	-	-	-	0.00%	-
QE-2	-	-	-	-	0.00%	-
QE-3	80.89%	68.26%	73.77%	0.09%	12.90%	70.58%
QE-4	-	-	-	-	0.00%	-
QE-5	75.16%	60.89%	66.95%	0.11%	19.35%	67.13%
QE-6	100.00%	84.62%	91.67%	0.00%	1.61%	81.68%

3) Machine Learning-based Approach - K-Nearest Neighbor Algorithm (*AK*=6, *QK*=9)

To evaluate the KNN-based approach for the multi-label non-fuzzy service retrieval, the performance was tested by setting the annotating *k* value (*AK*) and the querying *k* value (*QK*) as 6 and 9 respectively. The performances of this approach are presented in Table 7.17. The precision values of this approach were in the range of 57% to 87%. All query expansion methods were able to improve the performance of the multi-label non-fuzzy KNN-based approach. The approach with QE-1 returned the highest precision and recall values; however, the retrieval rate was less than 10%. Given queries, the KNN-based approach was able to retrieve more services than the FF-based and RBF-based approaches.

Table 7.17 The experiments of KNN-based approach for a multi-label non-fuzzy service retrieval (*AK*=6, *QK*=9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	57.55%	64.12%	56.71%	0.16%	6.45%	57.09%
QE-1	86.99%	81.53%	79.79%	0.14%	9.68%	75.43%
QE-2	70.88%	60.94%	65.18%	0.17%	6.45%	63.21%
QE-3	67.26%	61.69%	60.76%	0.28%	19.35%	63.13%
QE-4	58.93%	55.92%	53.94%	0.29%	19.35%	57.91%
QE-5	61.23%	55.21%	54.58%	0.31%	22.58%	59.30%
QE-6	62.94%	64.18%	60.49%	0.28%	16.13%	61.25%

4) Machine Learning-based Approach - Classification Tree Algorithm

Table 7.18 The experiments of CT-based approach for a multi-label non-fuzzy service retrieval

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	68.29%	70.66%	68.76%	0.38%	17.74%	65.83%
QE-1	67.75%	65.69%	64.70%	0.34%	27.42%	65.72%
QE-2	64.99%	64.08%	62.36%	0.38%	24.19%	63.54%
QE-3	67.77%	71.61%	69.06%	0.36%	24.19%	66.78%
QE-4	69.52%	72.42%	70.35%	0.34%	25.81%	68.04%
QE-5	62.30%	63.56%	62.35%	0.33%	32.26%	63.60%
QE-6	64.44%	66.72%	65.01%	0.34%	27.42%	64.60%

The performances of the CT-based approach for the multi-label non-fuzzy service retrieval are presented in Table 7.18. All precision and recall values of this approach were greater than 50%, and retrieval rates were in the range of 17% to 33%. The approach with QE-4 gave the best performance with 69.52% in precision and 72.42% in recall, while the approach with QE-5 returned the lowest performance with 62.30% and 63.56% in precision and recall respectively. Applying the query expansion methods tended to increase the retrieval rates around 7-15%.

5) Machine Learning-based Approach – Support Vector Machine

The performances of the SVM-based approach for the multi-label non-fuzzy service retrieval are presented in Table 7.19. Overall precision values of this approach were quite low which were in the range of 3-17%; meanwhile, recall values and retrieval rates were quite high with 45-100% and 35-100% respectively. Query expansion methods were able to increase precision values a little; however, recall and retrieval rates were sharply decreased. By considering all performance measures, the approach without query expansion outperformed other expansion methods.

Table 7.19 The experiments of SVM-based approach for a multi-label non-fuzzy service retrieval

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	3.26%	100.00%	5.87%	14.57%	100.00%	45.00%
QE-1	16.51%	45.73%	16.79%	4.95%	41.94%	36.53%
QE-2	14.02%	52.34%	14.35%	6.39%	35.48%	34.98%
QE-3	8.52%	57.74%	11.70%	7.49%	40.32%	33.75%
QE-4	12.40%	62.29%	14.89%	7.76%	35.48%	35.70%
QE-5	14.05%	48.17%	14.34%	5.65%	37.10%	34.71%
QE-6	13.78%	54.41%	14.42%	6.72%	37.10%	35.39%

Combine-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network (*AN*=10, *QN*=90)

To evaluate the FF-based approach for the combine-label non-fuzzy service retrieval, the performance was tested by setting the annotating hidden neurons (*AN*) and the querying hidden neurons (*QN*) as 10 and 90 respectively. The performances of this approach are presented in Table 7.20. Overall, precision and recall values of this approach were high, more than 70%. In contrast, retrieval rates were quite low and less than 10%. In addition, the approach with QE-6 gave the best result with 96.88% and 81.58% in precision and recall respectively.

Table 7.20 The experiments of FF-based approach for a combine-label non-fuzzy service retrieval (*AN*=10, *QN*=90)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	92.61%	78.36%	84.89%	0.19%	8.06%	77.71%
QE-1	73.45%	71.60%	67.90%	2.63%	6.45%	65.88%
QE-2	92.61%	78.36%	84.89%	0.19%	8.06%	77.71%
QE-3	92.61%	78.36%	84.89%	0.19%	8.06%	77.71%
QE-4	92.61%	78.36%	84.89%	0.19%	8.06%	77.71%
QE-5	92.61%	78.36%	84.89%	0.19%	8.06%	77.71%
QE-6	96.88%	81.58%	88.57%	0.03%	4.84%	79.99%

2) Neural Network-based Approach - Radial Basis Function Network (*AS*=0.5, *QS*=0.9)

To evaluate the RBF-based approach for the combine-label non-fuzzy service retrieval, the performance was tested by setting the annotating spread value (*AS*) and the querying spread value (*QS*) as 0.5 and 0.9 respectively. The performances of this approach are presented in Table 7.21. The approach without query expansion gave the best performance with 91.29% in precision and 82.60% in recall; however it was able to retrieve only around 7% of services. Applying the query expansion methods assisted in increasing retrieval rates to around 32%. Regarding query expansion methods, the approach with QE-4 returned the lowest performance with approximately 57% and 53% in precision and recall, while other expansion methods gave more than 60% in precision.

Table 7.21 The experiments of RBF-based approach for a combine-label non-fuzzy service retrieval (*AS*=0.5, *QS*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	91.29%	82.60%	86.19%	0.17%	6.45%	77.78%
QE-1	61.61%	54.23%	56.93%	0.15%	6.45%	57.26%
QE-2	78.93%	69.16%	73.37%	0.10%	11.29%	69.63%
QE-3	72.83%	62.20%	66.33%	0.18%	22.58%	66.77%
QE-4	57.22%	52.85%	54.41%	0.17%	19.35%	56.85%
QE-5	69.10%	55.10%	59.05%	0.17%	32.26%	64.58%
QE-6	63.21%	52.48%	56.82%	0.14%	17.74%	59.32%

3) Machine Learning-based Approach - K-Nearest Neighbor Algorithm (*AK*=6, *QK*=9)

To evaluate the KNN-based approach for the combine-label non-fuzzy service retrieval, the performance was tested by setting the annotating *k* value (*AK*) and the querying *k* value (*QK*) as 6 and 9 respectively. The performances of this approach are presented in Table 7.22. The approach with QE-1 gave the best performance with 75.72%, 63.76% and 17.74% in precision, recall and retrieval rate. Moreover, applying hybrid query expansion methods was able to increase recall and retrieval values, whereas precision values were reduced and less than using other expansion methods.

Table 7.22 The experiments of KNN-based approach for a combine-label non-fuzzy service retrieval (*AK*=6, *QK*=9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	47.47%	47.31%	44.42%	0.32%	24.19%	51.33%
QE-1	75.72%	63.76%	63.00%	0.27%	17.74%	66.92%
QE-2	49.81%	51.84%	42.07%	0.39%	27.42%	53.06%
QE-3	55.76%	63.65%	52.25%	0.31%	24.19%	58.27%
QE-4	47.69%	46.21%	41.72%	0.30%	30.65%	51.82%
QE-5	44.81%	56.53%	41.11%	0.59%	38.71%	53.29%
QE-6	42.10%	49.83%	40.34%	0.42%	32.26%	50.14%

4) Machine Learning-based Approach - Classification Tree Algorithm

Table 7.23 The experiments of CT-based approach for a combine-label non-fuzzy service retrieval

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	53.08%	61.15%	54.87%	0.35%	24.19%	57.21%
QE-1	62.09%	64.90%	60.53%	0.34%	32.26%	63.44%
QE-2	56.05%	60.13%	55.05%	0.35%	30.65%	59.24%
QE-3	65.04%	71.43%	66.09%	0.34%	27.42%	65.70%
QE-4	58.82%	64.61%	59.25%	0.34%	29.03%	61.41%
QE-5	52.94%	62.32%	53.85%	0.39%	35.48%	58.87%
QE-6	58.12%	63.56%	59.02%	0.34%	32.26%	61.42%

The performances of the CT-based approach for the combine-label non-fuzzy service retrieval are presented in Table 7.23. The approach with QE-3 gave the best performance with 65.04% in precision and 71.43% in recall, while the approach without the query expansion returned the worst performance with 53.08% in precision and 61.15% in recall. In addition, using the query expansion was able to improve the performances of the approach. The retrieval rates were in the range of 25-35%.

7.5.2 Fuzzy based Retrieval Approach

This section demonstrates the experimental results of the fuzzy-based service retrieval approaches. Similar the non-fuzzy based approach, some settings of the annotation and querying variables are selected for testing the approaches because of the experimental time issue. Settings of the experimental variables of the RBF proposed service retrieval approaches are presented in Table 7.24.

To evaluate the ECBR and the Vector-based approaches, both the annotation threshold *(AT)* and the querying threshold *(QT)* are set as 0 because all service concepts that relate to the services and query need to be considered. In terms of the Classification-based approach, the hidden neurons, the spread value and the number of *k* are set the same as the variables defined for evaluating the non-fuzzy based approaches. In this experiment, the fuzzy variables, their Gaussian-shaped membership functions, and the fuzzy rules are defined and presented in Section 7.4.1 and Section 7.4.2 In addition, the service retrieval approaches are also tested by using the Trapezoidal-shaped membership functions. The experimental results are demonstrated in Appendix B.

Table 7.24 The parameter settings of the fuzzy-based semantic service retrieval approaches

Service Retrieval Approaches		**Annotation Variables**	**Querying Variables**	**Retrieval Thresholds**
ECBR	Single-label	AT=0	QT=0	RT=0.25
	Multi-label	AT=0	QT=0	RT=0.55
	Combine-label	AT=0	QT=0	RT=0.55
Vector-based	Single-label	AT=0	QT=0	VSM : RT=0.65 EVSM : RT=0.95
	Multi-label	AT=0	QT=0	VSM : RT=0.95 EVSM : RT=0.95
	Combine-label	AT=0	QT=0	VSM : RT=0.95 EVSM : RT=0.95
Classification-based	Single-label	FF : N=80 RBF : S=0.9 KNN : K=9	FF : N=10 RBF : S=0.9 KNN : K=6	RT=0.75
	Multi-label	FF : N=10 RBF : S=0.5 KNN : K=6	FF : N=10 RBF : S=0.9 KNN : K=9	RT=0.95
	Combine-label	FF : N=10 RBF : S=0.5 KNN : K=6	FF : N=90 RBF : S=0.9 KNN : K=9	RT=0.95

ECBR Retrieval Approach

Single-Label based Approach (AT=0, QT=0, RT=0.25)

To evaluate the performance of the fuzzy ECBR approach for the single-label service retrieval, the annotation threshold *(AT)*, the querying threshold and the retrieval threshold *(RT)* are set as 0, 0 and 0.25 respectively. The performance of this approach is displayed in Table 7.25. The performances of this approach were quite close and retrieval rates of all expansion methods were 95.83%. Applying QE-4 gave the best performance with 13.14% and 42.55% in precision and recall. Moreover, the approach with all expansion methods increased recall values when comparing with non-expansion method. Unfortunately, overall precision values of this approach were low and less than 14%.

Table 7.25 The experiments of ECBR approach for a single-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.25)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	11.85%	36.35%	13.83%	0.74%	95.83%	41.53%
QE-1	9.63%	36.98%	11.73%	0.88%	95.83%	40.40%
QE-2	9.76%	36.98%	11.88%	0.84%	95.83%	40.48%
QE-3	9.50%	46.06%	12.00%	1.28%	95.83%	41.69%
QE-4	13.14%	42.55%	15.30%	0.87%	95.83%	43.18%
QE-5	9.33%	46.69%	11.80%	1.42%	95.83%	41.67%
QE-6	11.12%	43.18%	13.43%	0.98%	95.83%	42.17%

Multi-Label based Approach (AT=0, QT=0, RT=0.55)

The performance of the fuzzy ECBR approach for the multi-label service retrieval is presented in Table 7.26. To test the approach, *AT*, *QT*, and *RT* variables are set as 0, 0 and 0.55 respectively. The approach with QE-3 gave the best performance with 18.72% in precision, 64.45% in recall and 91.94% in retrieval rate. Furthermore, expanding the query by using ontology-based methods was able to much improve retrieval rate and all performance measures.

Table 7.26 The experiments of ECBR approach for a multi-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.55)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	18.49%	57.08%	20.20%	1.11%	58.06%	42.53%
QE-1	16.03%	57.10%	18.69%	1.37%	59.68%	41.53%
QE-2	16.80%	58.10%	19.34%	1.29%	58.06%	41.85%
QE-3	18.72%	64.45%	23.13%	1.53%	91.94%	49.19%
QE-4	16.36%	57.12%	18.82%	1.18%	87.10%	45.82%
QE-5	16.08%	56.45%	20.33%	1.18%	87.10%	45.84%
QE-6	17.78%	62.32%	22.20%	1.26%	70.97%	45.25%

Combine-Label based Approach (AT=0, QT=0, RT=0.55)

Table 7.27 The experiments of ECBR approach for a combine-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.55)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	18.49%	57.08%	20.20%	1.11%	58.06%	42.53%
QE-1	16.03%	57.10%	18.69%	1.37%	59.68%	41.53%
QE-2	16.80%	58.10%	19.34%	1.29%	58.06%	41.85%
QE-3	18.72%	64.45%	23.13%	1.53%	91.94%	49.19%
QE-4	16.36%	57.12%	18.82%	1.18%	87.10%	45.82%
QE-5	16.08%	56.45%	20.33%	1.18%	87.10%	45.84%
QE-6	17.78%	62.32%	22.20%	1.26%	70.97%	45.25%

To evaluate the fuzzy ECBR approach for combine-label service retrieval, the performance was tested by setting *AT*, *QT* and *RT* variables as 0, 0 and 0.55 respectively. The experimental results of this approach are shown in Table 7.27. It was clear to say that the results of the fuzzy combine-label based approach were the same as the ones of the multi-label based approach. That is, in this case, using the fuzzy single-label based approach was unable to improve and was ineffective when compared with the performance of the fuzzy combine-label based approach.

Vector-based Retrieval Approach

Single-Label based Approach (AT=0, QT=0, RT=0.9)

To evaluate the performance of the fuzzy Vector-based approach for the single-label service retrieval, *AT*, *QT* and *RT* variables are set as 0, 0 and 0.9 respectively. As mentioned in Chapter 4, the Vector-based approaches are categorized into VSM-based and EVSM-based approaches. The performances of the fuzzy VSM-based and the fuzzy EVSM-based service retrieval approaches are shown in Table 7.28 and Table 7.29 respectively.

The performance of the fuzzy single-label VSM-based approach in Table 7.28 showed that the approach with QE-3 gave the best performance with 15.66% in precision, 36.52% in recall and 95.83% in retrieval rate. Overall, precision, recall and retrieval rate values of this approach were around 13-16%, 31-37%, and 80-96% respectively. In addition, it seemed that using query expansion methods might not assist the approach to improve the performance.

Table 7.28 The experiments of Vector (VSM) approach for a single-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	15.62%	34.58%	18.14%	0.43%	83.33%	41.59%
QE-1	15.46%	31.69%	17.19%	0.34%	81.25%	40.65%
QE-2	15.95%	33.51%	17.78%	0.41%	81.25%	41.20%
QE-3	15.66%	36.52%	16.89%	0.54%	95.83%	43.57%
QE-4	13.59%	35.82%	16.33%	0.49%	87.50%	41.31%
QE-5	15.72%	33.10%	17.82%	0.40%	85.42%	41.68%
QE-6	15.29%	33.73%	17.80%	0.42%	83.33%	41.28%

The experimental results of the fuzzy EVSM-based approach for the single-label service retrieval in Table 7.29 demonstrated that the approach with QE-1 returned the best performance with around 47.46% in precision and 36% recall, and was able to retrieve services from 14.58% of query samples. Although using QE-4 gave a little higher precision value, 50%, the retrieval rate was only 6.25%. Furthermore, using QE-0, QE-3 and QE-5 was unable to retrieve any service from testing queries. By comparing with the VSM-based approach, precision values of EVSM-based approach were better, but retrieval rate was decreased.

Table 7.29 The experiments of Vector (EVSM) approach for a single-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	-	-	-	-	0.00%	-
QE-1	47.46%	36.00%	37.68%	0.24%	14.58%	47.19%
QE-2	33.06%	32.08%	31.36%	0.24%	8.33%	38.95%
QE-3	-	-	-	-	0.00%	-
QE-4	50.00%	34.21%	35.00%	0.08%	6.25%	46.31%
QE-5	-	-	-	-	0.00%	-
QE-6	0.00%	0.00%	0.00%	0.31%	4.17%	15.58%

Multi-Label based Approach (AT=0, QT=0, RT=0.95)

To evaluate the performance of the fuzzy Vector-based approach for the multi-label service retrieval, *AT*, *QT* and *RT* variables are set as 0, 0 and 0.95 respectively. The performances of the fuzzy VSM-based and the fuzzy EVSM-based service retrieval approaches are shown in Table 7.30 and Table 7.31 respectively.

Table 7.30 The experiments of Vector (VSM) approach for a multi-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.95)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	35.48%	48.43%	17.49%	0.36%	9.68%	40.48%
QE-1	-	-	-	-	0.00%	-
QE-2	58.33%	56.58%	25.91%	0.08%	3.23%	51.18%
QE-3	13.72%	60.00%	21.24%	0.33%	9.68%	34.08%
QE-4	33.61%	41.65%	28.73%	0.21%	25.81%	42.84%
QE-5	33.33%	100.00%	50.00%	0.07%	1.61%	51.07%
QE-6	29.44%	36.45%	21.73%	0.10%	9.68%	36.94%

The performance of the fuzzy multi-label VSM-based approach in Table 7.30 showed that the approach with QE-2 gave the best results with 58.33% in precision and 56.58% in recall. Overall retrieval rates of this approach were low and less than 10%. Although QE-4 could retrieve service from 25.81% of query samples, its precision value was less than non-expansion method. Moreover, QE-1 was unable to retrieve any service.

The results of the fuzzy multi-label EVSM-based approach in Table 7.31 presented that the approach with QE-5 gave the best performance with approximately 80% in precision and 40% in recall; meanwhile, the approach with QE-6 gave the worst performance with around 30% and 15% in precision and recall respectively. However, retrieval rate of QE-5 was only 1.61%. In addition, using query expansion methods tended to assist in recall values. Like VSM-based approach, overall retrieval rates of EVSM-based approach were low and less than 10%.

Table 7.31 The experiments of Vector (EVSM) approach for a multi-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.95)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	52.98%	25.57%	34.49%	0.22%	3.23%	45.65%
QE-1	41.37%	46.36%	32.65%	0.32%	22.58%	46.74%
QE-2	42.29%	54.61%	34.19%	0.29%	12.90%	47.13%
QE-3	38.10%	46.21%	34.10%	0.11%	4.84%	42.99%
QE-4	56.19%	40.42%	45.86%	0.21%	8.06%	51.60%
QE-5	80.95%	38.64%	52.31%	0.14%	1.61%	61.24%
QE-6	30.32%	14.73%	19.82%	0.19%	4.84%	33.01%

Combine-Label based Approach (AT=0, QT=0, RT=0.95)

To evaluate the performance of the fuzzy Vector-based approach for the combine-label service retrieval, *AT*, *QT* and *RT* variables are set as 0, 0 and 0.95 respectively. The performances of the fuzzy combine-label VSM-based and the fuzzy combine-label EVSM-based service retrieval approaches are shown in Table 7.32 and Table 7.33 respectively.

The performances of the fuzzy combine-label approach for VSM-based service retrieval and EVSM-based service retrieval in Table 7.32 and Table 7.33 were the same as the performances of the fuzzy multi-label approach in Table 7.30 and Table 7.31 respectively. That is, in this case, combining the single-label approach into the multi-label approach was unable to improve the retrieval performance. This was because the results of testing queries might relate to service concepts that both multi-label and single-label approaches were able to query. That is, the multi-label approach was capable of querying all concepts that the single-label approach could.

Table 7.32 The experiments of Vector (VSM) approach for a combine-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.95)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	35.48%	48.43%	17.49%	0.36%	9.68%	40.48%
QE-1	-	-	-	-	0.00%	-
QE-2	58.33%	56.58%	25.91%	0.08%	3.23%	51.18%
QE-3	13.72%	60.00%	21.24%	0.33%	9.68%	34.08%
QE-4	33.61%	41.65%	28.73%	0.21%	25.81%	42.84%
QE-5	33.33%	100.00%	50.00%	0.07%	1.61%	51.07%
QE-6	29.44%	36.45%	21.73%	0.10%	9.68%	36.94%

Table 7.33 The experiments of Vector (EVSM) approach for a combine-label fuzzy service retrieval (*AT*=0, *QT*=0, *RT*=0.95)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	52.98%	25.57%	34.49%	0.22%	3.23%	45.65%
QE-1	41.37%	46.36%	32.65%	0.32%	22.58%	46.74%
QE-2	42.29%	54.61%	34.19%	0.29%	12.90%	47.13%
QE-3	38.10%	46.21%	34.10%	0.11%	4.84%	42.99%
QE-4	56.19%	40.42%	45.86%	0.21%	8.06%	51.60%
QE-5	80.95%	38.64%	52.31%	0.14%	1.61%	61.24%
QE-6	30.32%	14.73%	19.82%	0.19%	4.84%	33.01%

Classification-based Retrieval Approach

Based on the classification techniques, the experimental results of the fuzzy Classification-based retrieval approaches are divided into FF-based, RBF-based, KNN-based, CT-based and SVM-based approaches. All classification techniques are applied in the multi-label service retrieval; meanwhile only the first four techniques are used in the single-label and combine-label service retrieval.

Single-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network (*AN*=80, *QN*=10)

To evaluate the fuzzy FF-based approach for the single-label service retrieval, the number of annotating hidden neurons (*AN*) and querying hidden neurons *(QT)* are set as 80 and 10 respectively. The performances of this approach are shown in Table 7.34. The experimental results show that the approach with QE-3 gave the best performance with around 96% in precision and 68% in recall, but it was able to retrieve services from only 2.08% of query samples. Although overall precision and recall values of this approach were quite high, retrieval rate values were too low; less than 10%, and needed to be improved. Moreover, the results showed that query expansion assisted the approach to improve the performance.

Table 7.34 The experiments of FF-based approach for a single-label fuzzy service retrieval (*AN*=80, *QN*=10)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	0.00%	0.00%	0.00%	0.34%	2.08%	15.26%
QE-1	33.58%	50.88%	38.38%	0.92%	6.25%	42.62%
QE-2	78.47%	68.42%	72.90%	0.26%	4.17%	68.17%
QE-3	96.30%	68.42%	80.00%	0.03%	2.08%	76.09%
QE-4	77.73%	64.47%	70.05%	0.26%	4.17%	66.86%
QE-5	66.48%	68.42%	66.85%	0.50%	4.17%	62.43%
QE-6	39.16%	29.61%	33.17%	0.99%	8.33%	41.18%

2) Neural Network-based Approach - Radial Basis Function Network (*AS*=0.9, *QS*=0.9)

To evaluate the fuzzy RBF-based approach for the single-label service retrieval, the annotating spread value (*AS*) and the querying spread value (*QS*) are set as 0.9

and 0.9 respectively. The performances of this approach are displayed in Table 7.35. The results showed that the approach with QE-0, QE-1 and QE-2 could not retrieve any service from the query samples. In contrast, QE-4 and QE-6 gave the best precision value with approximate 94% and 40% in recall. Unfortunately, overall retrieval rates of this approach were low and less than 10%.

Table 7.35 The experiments of RBF-based approach for a single-label fuzzy service retrieval (*AS*=0.9, *QS*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	-	-	-	-	0.00%	-
QE-1	-	-	-	-	0.00%	-
QE-2	-	-	-	-	0.00%	-
QE-3	64.58%	17.32%	25.93%	0.03%	6.25%	48.25%
QE-4	93.75%	39.47%	55.56%	0.03%	2.08%	67.06%
QE-5	73.44%	16.28%	25.26%	0.03%	8.33%	51.85%
QE-6	93.75%	39.47%	55.56%	0.03%	2.08%	67.06%

3) Machine Learning-based Approach - K-Nearest Neighbor Algorithm (*AK*=9, *QK*=6)

To test the performance of the fuzzy KNN-based approach for the single-label service retrieval, the numbers of *K* for annotation (*AK*) and querying (*QK*) were defined as 9 and 6 respectively. The experimental results of this approach in Table 7.36 demonstrated that the approach with QE-6 gave the best performance with approximately 64% in precision and 38% in recall. Overall, most performances of the approach with the query expansion were around 25-65% in precision, and outperformed the approach without the query expansion. However, retrieval rates were in the range of 10-20%.

Table 7.36 The experiments of KNN-based approach for a single-label fuzzy service retrieval (*AK*=9, *QK*=6)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	16.27%	31.58%	14.35%	0.24%	14.58%	30.55%
QE-1	27.41%	35.32%	16.36%	0.45%	16.67%	36.15%
QE-2	8.23%	40.00%	13.56%	0.26%	10.42%	27.85%
QE-3	33.12%	28.85%	21.26%	0.32%	12.50%	37.59%
QE-4	38.36%	26.88%	24.53%	0.26%	18.75%	40.83%
QE-5	44.53%	26.50%	25.82%	0.34%	20.83%	43.73%
QE-6	63.61%	37.85%	36.89%	0.20%	14.58%	53.81%

4) Machine Learning-based Approach - Classification Tree Algorithm

The performances of the fuzzy CT-based approach for the single-label service retrieval are shown in Table 7.37. Overall, the precision and recall values of the approach with query expansion methods were much greater than the approach without the query expansion; approximate 4-37% increase and 1-30% increase in precision and recall respectively. However, this approach was able to retrieve services from less than 25% of all query samples.

Table 7.37 The experiments of CT-based approach for a single-label fuzzy service retrieval

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	7.29%	2.30%	3.50%	0.17%	16.67%	21.26%
QE-1	17.48%	8.03%	10.83%	0.39%	16.67%	27.26%
QE-2	11.67%	3.68%	5.60%	0.18%	10.42%	22.59%
QE-3	44.65%	20.14%	27.35%	0.14%	14.58%	42.15%
QE-4	34.73%	15.66%	21.27%	0.13%	18.75%	37.22%
QE-5	29.19%	27.97%	18.86%	0.49%	22.92%	37.07%
QE-6	32.11%	30.76%	20.75%	0.33%	20.83%	38.65%

Multi-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network (*AN*=10, *QN*=10)

To evaluate the fuzzy FF-based approach for the multi-label service retrieval, the number of annotating hidden neurons (*AN*) and querying hidden neurons *(QT)* are set as 10 and 10 respectively. The performances of this approach are shown in Table 7.38. The experimental results demonstrated that the overall precision and recall values of this approach were quite high, 90-98% in precision and 40-82% in recall; however, retrieval rates were less than 10%. In addition, performances of the approach with hybrid-based query expansion methods outperformed the approach without the query expansion.

Table 7.38 The experiments of FF-based approach for a multi-label fuzzy service retrieval (*AN*=10, *QN*=10)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	91.11%	63.98%	74.53%	0.19%	8.06%	73.40%
QE-1	97.88%	53.23%	64.57%	0.02%	8.06%	73.03%
QE-2	98.13%	46.14%	58.58%	0.02%	6.45%	70.93%
QE-3	93.60%	54.51%	65.34%	0.07%	6.45%	71.37%
QE-4	95.15%	40.71%	54.88%	0.04%	6.45%	68.36%
QE-5	96.88%	81.58%	88.57%	0.03%	3.23%	79.75%
QE-6	90.46%	75.19%	81.61%	0.27%	8.06%	75.87%

2) Neural Network-based Approach - Radial Basis Function Network (*AS*=0.5, *QS*=0.9)

To evaluate the fuzzy RBF-based approach for the multi-label service retrieval, the annotating spread value (*AS*) and the querying spread value (*QS*) are set as 0.5 and 0.9 respectively. The performances of this approach are displayed in Table 7.39. Although the approach with QE-0, QE-1, QE-2 and QE-4 was unable to retrieve any service, the precision values of the approach with other query expansion methods were more than 75%. Expanding the query by using QE-6 gave the best performance in both precision and recall. Unfortunately, the overall retrieval rates of this approach were less than 20%.

Table 7.39 The experiments of RBF-based approach for a multi-label fuzzy service retrieval (*AS*=0.5, *QS*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	-	-	-	-	0.00%	-
QE-1	-	-	-	-	0.00%	-
QE-2	-	-	-	-	0.00%	-
QE-3	80.89%	68.26%	73.77%	0.09%	12.90%	70.58%
QE-4	-	-	-	-	0.00%	-
QE-5	75.16%	60.89%	66.95%	0.11%	19.35%	67.13%
QE-6	100.00%	84.62%	91.67%	0.00%	1.61%	81.68%

3) Machine Learning-based Approach - K-Nearest Neighbor Algorithm (*AK*=6, *QK*=9)

To test the performance of the fuzzy KNN-based approach for the multi-label service retrieval, the numbers of *K* for annotation (*AK*) and querying (*QK*) were defined as 6 and 9 respectively. The experimental results in Table 7.40 demonstrated that the approach with QE-3 gave the best performance with approximately 63%, 54% and 24% in precision, recall and retrieval rate. Comparing with the approach without the query expansion, all query expansion methods (QE-1 to QE-6) gave much better precision and recall values. Applying the ontology-based and hybrid-based query expansion tended to increase the number of retrieved queries.

Table 7.40 The experiments of KNN-based approach for a multi-label fuzzy service retrieval (*AK*=6, *QK*=9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	42.90%	39.35%	36.40%	0.34%	8.06%	44.68%
QE-1	60.50%	47.75%	48.79%	0.27%	14.52%	55.82%
QE-2	58.75%	56.52%	57.60%	0.43%	6.45%	56.52%
QE-3	62.59%	53.57%	54.85%	0.25%	24.19%	59.89%
QE-4	58.75%	55.09%	55.39%	0.28%	20.97%	58.18%
QE-5	58.56%	56.33%	54.32%	0.31%	27.42%	59.09%
QE-6	60.81%	57.97%	57.87%	0.25%	20.97%	59.81%

4) Machine Learning-based Approach - Classification Tree Algorithm

The performances of the fuzzy CT-based approach for the multi-label service retrieval are shown in Table 7.41. Overall, the performances of this approach with any query expansion methods were high with 59-66% in precision, 66-75% in recall and 17-32% in retrieval rate. In this case, applying the ontology-based query expansion (QE-3 and QE-4) gave better performance in both precision and recall than retrieving services without the query expansion.

Table 7.41 The experiments of CT-based approach for a multi-label fuzzy service retrieval

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	65.18%	72.30%	68.16%	0.38%	17.74%	64.74%
QE-1	65.37%	69.11%	65.17%	0.34%	27.42%	65.35%
QE-2	63.09%	66.84%	62.63%	0.37%	24.19%	63.23%
QE-3	65.62%	74.71%	69.29%	0.36%	24.19%	66.42%
QE-4	66.10%	75.33%	69.87%	0.36%	25.81%	67.04%
QE-5	59.29%	67.40%	62.65%	0.35%	32.26%	63.01%
QE-6	61.45%	69.16%	64.59%	0.35%	27.42%	63.70%

5) Machine Learning-based Approach – Support Vector Machine Algorithm

The performances of the fuzzy SVM-based approach for the multi-label service retrieval are shown in Table 7.42. Overall, the precision values of this approach with any query expansion methods were low and in the range of 3-22%; on the other hand, the recall values were quite high and in the range of 47-90%. In addition, the retrieval rates were also quite high, 37-100%. The precision values of the SVM-based approach with the query expansion methods were 10-18% better than the approach without the query expansion; meanwhile, the recall values were 25-40% decreased. In this case, using query expansion methods could not improve the overall performance of non-query expansion based approach.

Table 7.42 The experiments of SVM-based approach for a multi-label fuzzy service retrieval

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	3.12%	88.97%	5.51%	12.58%	100.00%	43.53%
QE-1	20.18%	47.90%	21.37%	2.97%	45.16%	39.79%
QE-2	19.02%	51.59%	19.67%	3.83%	40.32%	38.77%
QE-3	13.90%	62.37%	15.81%	6.95%	41.94%	37.53%
QE-4	17.83%	62.86%	19.98%	5.81%	37.10%	39.25%
QE-5	21.58%	52.48%	22.53%	3.62%	38.71%	40.15%
QE-6	21.65%	61.84%	23.34%	4.46%	38.71%	41.57%

Combine-Label based Approach

1) Neural Network-based Approach - Feed-Forward Neural Network (*AN*=10, *QN*=90)

To evaluate the fuzzy FF-based approach for the combine-label service retrieval, the number of annotating hidden neurons (*AN*) and querying hidden neurons *(QT)* are set as 10 and 90 respectively. The performances of this approach are shown in Table 7.43. The results demonstrated that this approach performed well with high precision and recall values, but retrieval rates were less than 10%. In this case, expanding query by using QE-6 slightly improved performance of the approach without query expansion.

Table 7.43 The experiments of FF-based approach for a combine-label fuzzy service retrieval (*AN*=10, *QN*=90)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	90.70%	79.93%	84.90%	0.27%	8.06%	77.17%
QE-1	78.76%	58.39%	56.43%	2.11%	8.06%	64.62%
QE-2	90.70%	79.93%	84.90%	0.27%	8.06%	77.17%
QE-3	90.70%	79.93%	84.90%	0.27%	8.06%	77.17%
QE-4	90.46%	75.19%	81.61%	0.27%	8.06%	75.87%
QE-5	91.84%	59.64%	67.03%	0.23%	9.68%	72.15%
QE-6	96.88%	81.58%	88.57%	0.03%	4.84%	79.99%

2) Neural Network-based Approach - Radial Basis Function Network (*AS*=0.5, *QS*=0.9)

To evaluate the fuzzy RBF-based approach for the combine-label service retrieval, the annotating spread value (*AS*) and the querying spread value (*QS*) are set as 0.5 and 0.9 respectively. The experimental results of this approach in Table 7.44 show that the precision and recall values were high and greater than 50%. In this case, the approach without the query expansion gave the best performance with approximate 91% in precision and 83% in recall; however, its retrieval rate was just around 6.5%. Regarding the query expansion methods, the approach with QE-2 outperformed other methods. Moreover, ontology-based and hybrid-based expansion methods were able to increase retrieval rate values.

Table 7.44 The experiments of RBF-based approach for a combine-label fuzzy service retrieval (*AS*=0.5, *QS*=0.9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	91.29%	82.60%	86.19%	0.17%	6.45%	77.78%
QE-1	61.61%	54.23%	56.93%	0.15%	6.45%	57.26%
QE-2	78.93%	69.16%	73.37%	0.10%	11.29%	69.63%
QE-3	72.83%	62.20%	66.33%	0.18%	22.58%	66.77%
QE-4	57.22%	52.85%	54.41%	0.17%	19.35%	56.85%
QE-5	69.10%	55.10%	59.05%	0.17%	32.26%	64.58%
QE-6	63.21%	52.48%	56.82%	0.14%	17.74%	59.32%

3) Machine Learning-based Approach - K-Nearest Neighbor Algorithm (*AK*=6, *QK*=9)

To test the performance of the fuzzy KNN-based approach for the combine-label service retrieval, the numbers of *K* for annotation (*AK*) and querying (*QK*) were defined as 6 and 9 respectively. The experimental results in Table 7.45 demonstrated that overall precision values, recall values, and retrieval rates of this approach were quite high and were in the range of 44-68%, 45-53%, and 27-45% respectively. Expanding query by using proper synonyms and related ontological terms (QE-1 and QE-3) gave the better performances than the others. In contrast, retrieval rates of the approach with the ontology-based and hybrid-based expansion methods were 1-13% increased when comparing with non-expansion method.

Table 7.45 The experiments of KNN-based approach for a combine-label fuzzy service retrieval (*AK*=6, *QK*=9)

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	53.71%	48.72%	46.26%	0.30%	32.26%	55.52%
QE-1	68.08%	53.11%	51.99%	0.28%	27.42%	62.07%
QE-2	50.81%	45.99%	37.73%	0.53%	30.65%	52.40%
QE-3	56.51%	51.74%	50.77%	0.27%	32.26%	57.78%
QE-4	51.48%	45.84%	46.86%	0.30%	33.87%	54.53%
QE-5	44.69%	49.76%	41.60%	0.54%	45.16%	53.27%
QE-6	50.79%	49.99%	44.10%	0.40%	40.32%	55.42%

4) Machine Learning-based Approach - Classification Tree Algorithm

The performances of the fuzzy CT-based approach for the combine-label service retrieval are shown in Table 7.46. Precision values, recall values, and retrieval rates of this approach were in the range of 54-66%, 62-75%, and 24-35% respectively. The approach with QE-3 gave the best performance with approximate 66% in precision and 75% in recall; meanwhile, the approach without the query expansion returned the worst performance with around 56%, 62% and 24% in precision, recall, and retrieval rate respectively. That is, in this case, all query expansion methods were able to improve the performances of the approach without query expansion.

Table 7.46 The experiments of CT-based approach for a combine-label fuzzy service retrieval

QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	55.57%	62.35%	58.22%	0.35%	24.19%	58.89%
QE-1	64.38%	68.74%	64.72%	0.34%	32.26%	65.56%
QE-2	59.08%	63.30%	59.26%	0.35%	30.65%	61.56%
QE-3	66.04%	75.15%	69.78%	0.35%	27.42%	67.21%
QE-4	59.68%	67.12%	62.37%	0.35%	29.03%	62.60%
QE-5	54.00%	65.82%	57.13%	0.41%	35.48%	60.30%
QE-6	59.29%	67.40%	62.65%	0.36%	32.26%	63.01%

7.6 Experiment Summary

Based on the experimental results in Section 7.5 , the best result of every proposed approach is selected and those results are compared in this section.

7.6.1 Non-Fuzzy based Service Retrieval Approach

The best results of the proposed non-fuzzy based approach for single-label, multi-label and combine-label service retrieval are presented in Table 7.47, Table 7.48 and Table 7.49 respectively. Based on the experimental results, it can be summarized that:

1) Non-fuzzy NN-based approaches, both FF-based and RBF-based approaches outperform other approaches in all single-label, multi-label and combine-label service retrieval.
2) Non-fuzzy NN-based approaches, like FF-based and RBF-based, give good precision values in all single-label, multi-label and combine-label retrieval. In addition, Vector-based approaches, VSM and EVSM, also perform well in precision for single-label and multi-label retrieval. Although precision values of ML-based approaches, KNN and CT, are also good, they are lower in precision values of NN-based approaches. In contrast, ECBR and SVM-based approaches return worse precision values than the others.
3) Recall values of non-fuzzy single-label service retrieval are lower than multi-label and combine-label service retrieval. This is because the real world services are able to relate to multiple service concepts, but the single-label based approach targets only one relevant service concept. Consequently, the

single-label based approach cannot retrieve all relevant services. EVSM-based and ECBR approaches return the worse recall values in single-label and multi-label service retrieval respectively. In contrast, recall values of all proposed approaches are good in combine-label service retrieval.

4) Although NN-based approaches perform well in precision, recall, f-measure and fallout rate, their retrieval rates are too low and less than 10%. In contrast, other proposed approaches are able to retrieve services from more testing queries; however, this leads those approaches to perform worse in other performance measures. In addition, combine-label Classification based approaches are able to increase retrieval rate values of multi-label Classification based approaches, but precision and recall values are slightly decreased.
5) Overall experimental results show that query expansion assists in performance improvement in service retrieval because most of the best results of non-fuzzy based service retrieval approaches apply query expansion methods. Based on the best performance of NN-based approaches, using ontology-based and hybrid-based query expansion methods most assist in performance improvement.

Table 7.47 The best results of non-fuzzy based approach for single-label service retrieval

Approach	**QE**	**Avg Precision**	**Avg Recall**	**Avg F-measure**	**Avg Fallout**	**Retrieval Rate**	**All-measures**
ECBR	QE-6	31.43%	34.34%	24.77%	0.19%	43.75%	42.97%
VSM	QE-0	62.50%	30.26%	26.19%	0.05%	4.17%	49.09%
EVSM	QE-3	100.00%	10.53%	19.05%	0.00%	2.08%	59.75%
FF	QE-0	79.17%	50.00%	61.29%	0.17%	2.08%	63.65%
RBF	QE-4	93.75%	39.47%	55.56%	0.03%	2.08%	67.06%
KNN	QE-6	63.61%	37.85%	36.89%	0.20%	14.58%	53.81%
CT	QE-3	44.65%	20.14%	27.35%	0.14%	14.58%	42.15%

Table 7.48 The best results of non-fuzzy based approach for multi-label service retrieval

Approach	QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
ECBR	QE-5	65.45%	8.53%	14.38%	0.08%	8.06%	45.82%
VSM	QE-1	100.00%	10.53%	19.05%	0.00%	1.61%	59.68%
EVSM	QE-5	82.14%	52.27%	63.89%	0.17%	1.61%	65.50%
FF	QE-5	96.88%	81.58%	88.57%	0.03%	3.23%	79.75%
RBF	QE-6	100.00%	84.62%	91.67%	0.00%	1.61%	81.68%
KNN	QE-1	86.99%	81.53%	79.79%	0.14%	9.68%	75.43%
CT	QE-4	69.52%	72.42%	70.35%	0.34%	25.81%	68.04%
SVM	QE-0	3.26%	100.00%	5.87%	14.57%	100.00%	45.00%

Table 7.49 The best results of non-fuzzy based approach for combine-label service retrieval

Approach	QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
ECBR	QE-5	15.67%	51.70%	17.81%	1.31%	98.39%	46.26%
VSM	QE-6	23.13%	58.66%	25.02%	0.90%	95.16%	50.94%
EVSM	QE-5	22.18%	60.10%	25.31%	0.87%	96.77%	51.07%
FF	QE-6	96.88%	81.58%	88.57%	0.03%	4.84%	79.99%
RBF	QE-0	91.29%	82.60%	86.19%	0.17%	6.45%	77.78%
KNN	QE-1	75.72%	63.76%	63.00%	0.27%	17.74%	66.92%
CT	QE-3	65.04%	71.43%	66.09%	0.34%	27.42%	65.70%

7.6.2 Fuzzy based Service Retrieval Approach

The best results of the proposed non-fuzzy based approach for single-label, multi-label and combine-label service retrieval are presented in Table 7.50, Table 7.51 and Table 7.52 respectively. Based on the experimental results, it can be summarized that:

1) Overall experimental results of fuzzy-based approaches for semantic service retrieval are similar to non-fuzzy based approaches. NN-based approaches obviously outperform other approaches in precision, recall, f-measure and fallout rate in all single-label, multi-label and combine-label service retrieval. However, low retrieval rate is a limitation of NN-based approaches.
2) Based on the performances of NN-based approaches, the fuzzy-based approach is able to much improve precision, recall and f-measure values of non-fuzzy based approach for single-label service retrieval. In contrast, performances of non-fuzzy based approach and fuzzy-based approach for

multi-label and combine-label service retrieval are identical. Regarding other approaches, it is obvious that performances of fuzzy ECBR approach are slightly improved in all single-label, multi-label and combine-label retrieval. However, for multi-label service retrieval, performances of fuzzy Vector-based and fuzzy ML-based approaches are slightly decreased.

Table 7.50 The best results of fuzzy based approach for single-label service retrieval

Approach	QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
ECBR	QE-4	13.14%	42.55%	15.30%	0.87%	95.83%	43.18%
VSM	QE-3	15.66%	36.52%	16.89%	0.54%	95.83%	43.57%
EVSM	QE-1	47.46%	36.00%	37.68%	0.24%	14.58%	47.19%
FF	QE-3	96.30%	68.42%	80.00%	0.03%	2.08%	76.09%
RBF	QE-4	93.75%	39.47%	55.56%	0.03%	2.08%	67.06%
KNN	QE-6	63.61%	37.85%	36.89%	0.20%	14.58%	53.81%
CT	QE-3	44.65%	20.14%	27.35%	0.14%	14.58%	42.15%

Table 7.51 The best results of fuzzy based approach for multi-label service retrieval

Approach	QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
ECBR	QE-3	18.72%	64.45%	23.13%	1.53%	91.94%	49.19%
VSM	QE-2	58.33%	56.58%	25.91%	0.08%	3.23%	51.18%
EVSM	QE-5	80.95%	38.64%	52.31%	0.14%	1.61%	61.24%
FF	QE-5	96.88%	81.58%	88.57%	0.03%	3.23%	79.75%
RBF	QE-6	100.00%	84.62%	91.67%	0.00%	1.61%	81.68%
KNN	QE-3	62.59%	53.57%	54.85%	0.25%	24.19%	59.89%
CT	QE-4	66.10%	75.33%	69.87%	0.36%	25.81%	67.04%
SVM	QE-0	3.12%	88.97%	5.51%	12.58%	100.00%	43.53%

Table 7.52 The best results of fuzzy based approach for combine-label service retrieval

Approach	QE	Avg Precision	Avg Recall	Avg F-measure	Avg Fallout	Retrieval Rate	All-measures
ECBR	QE-3	18.72%	64.45%	23.13%	1.53%	91.94%	49.19%
VSM	QE-2	58.33%	56.58%	25.91%	0.08%	3.23%	51.18%
EVSM	QE-5	80.95%	38.64%	52.31%	0.14%	1.61%	61.24%
FF	QE-6	96.88%	81.58%	88.57%	0.03%	4.84%	79.99%
RBF	QE-0	91.29%	82.60%	86.19%	0.17%	6.45%	77.78%
KNN	QE-1	68.08%	53.11%	51.99%	0.28%	27.42%	62.07%
CT	QE-3	66.04%	75.15%	69.78%	0.35%	27.42%	67.21%

7.7 Conclusion

This chapter presents the semantic-based approach for retrieving services. Given a query *Q,* the approach aims to return services that are relevant to *Q* by considering the service annotations and relevant service concepts from semantic service annotation module and the semantic service querying module respectively. Based on the semantic service annotation and service querying approaches, the proposed semantic service retrieval approaches are categorized into ECBR, Vector-based and Classification-based approaches. Moreover, the approaches are also divided into three types; namely single-label, multi-label and combine-label based approach. Each service retrieval approach applies the same type of service annotation and service querying approach in order to find out the relevant services. For instance, the single-label ECBR based service retrieval approach receives the input from the single-label ECBR based service annotation and service querying approaches.

In addition, every service retrieval approach is also divided into non-fuzzy based and fuzzy based approach. The non-fuzzy based approach applies the annotation threshold and the querying threshold to select the relevant concepts and the proper service annotations. Then, it retrieves services that are annotated to the service concepts being relevant to the given query. In contrast, the fuzzy based approach assumes that the annotation relevance and the querying relevance scores are fuzzy. The fuzzy variables, the fuzzy membership functions, and the fuzzy rules are defined in order to calculate the retrieval scores. Based on the fuzzy inference system, services are retrieved if their retrieval scores are greater than the retrieval threshold.

To evaluate the performance of the proposed service retrieval approaches, the precision, recall, f-measure, fallout rate, retrieval rate and all-measures are applied. The results demonstrate that most of the classification-based approaches outperform the ECBR and Vector-based approaches. In terms of the fuzzy inference, the fuzzy-based approaches need much runtime to retrieve the services; however, the performances of the fuzzy based and non-fuzzy based approaches are similar.

Chapter 8 System Implementation

8.1 Introduction

This chapter presents the software implementation of the semantic service retrieval methodology. The content of this chapter is divided into two main sub-topics; 1) tools and libraries that are used for implementing the retrieval system and 2) a developed prototype of the service retrieval system. Tools and libraries that are used for software implementation such as NetBeans IDE, Mathlab, WordNet API, Protégé OWL API, are presented in Section 8.2 . Those tools and libraries are applied for implementing the methodology and developing the service retrieval prototype in Transport service domain. As mentioned previously in Chapter 4, the proposed methodology consists of three primary parts, including the service annotation, the service querying, and the service retrieval parts. The service annotation part aims to automatically link services to relevant ontological service concepts. In contrast, the service querying and service retrieval parts are applied for retrieving services. Given a query, relevant service concepts are queried by the service querying part and relevant services are then retrieved by the service retrieval parts. Therefore, the proposed prototype in Section 8.3 consists of the service annotation prototype and the service retrieval prototype.

8.2 Tools and libraries

8.2.1 Tools

In this thesis, the service retrieval methodology is implemented by using Java and Matlab programming.

1) Java and NetBeans

As mentioned previously, three main tasks; the service annotation, the service querying, and the service retrieval, are considered. Almost all the processes of the system are programmed in Java language. In cases of the service annotation and the service querying, ECBR and Vector-based approaches are totally implemented by using Java; meanwhile, some parts of Classification-based approaches are implemented by using Matlab functions. Similarly, Java is used for implementing the whole service retrieval part. In this thesis, NetBeans IDE is used to assist in Java application development. The NetBeans IDE interface for the ECBR service annotation project is shown in Figure 8.1.

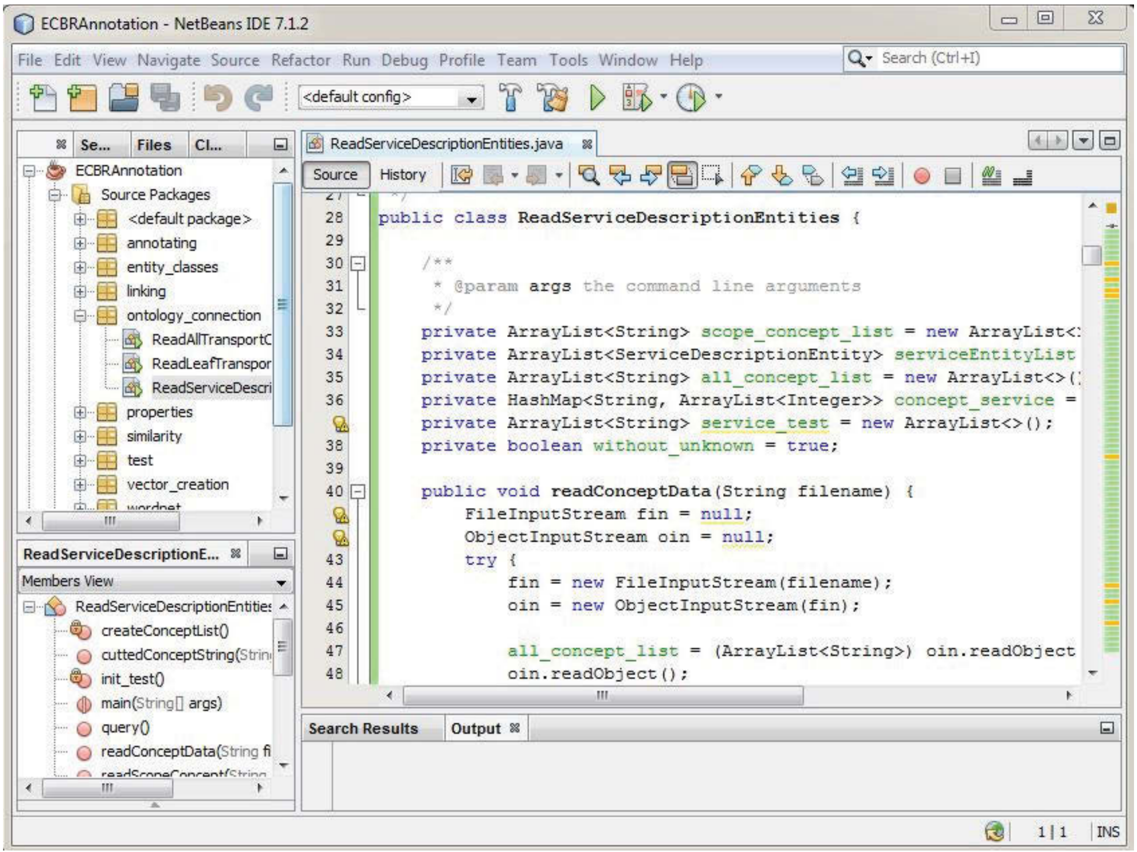


Figure 8.1 NetBeans IDE user interface for ECBR service annotation project

2) Matlab

Unlike the ECBR and Vector-based approaches, the Classification-based service annotation and service querying approaches need to apply classification methods, such as artificial neural networks and machine learning algorithms, provided in Matlab. Firstly, a Java program is developed for preprocessing and creating input, target and test datasets, which are used in service classification. Secondly, a Matlab classification program is computed and service concepts that are relevant to the input are returned. In this thesis, five classification algorithms are applied, namely Feed-Forward neural network, Radial Basis Function network, K-Nearest Neighbor, Classification Tree, and Support Vector Machine. To classify services into service concepts, for each classification method, functions for creating and training the service classifier are called. An example of Matlab code and Neural Network toolbox for training Feed-Forward based service classifiers is presented in Figure 8.2.

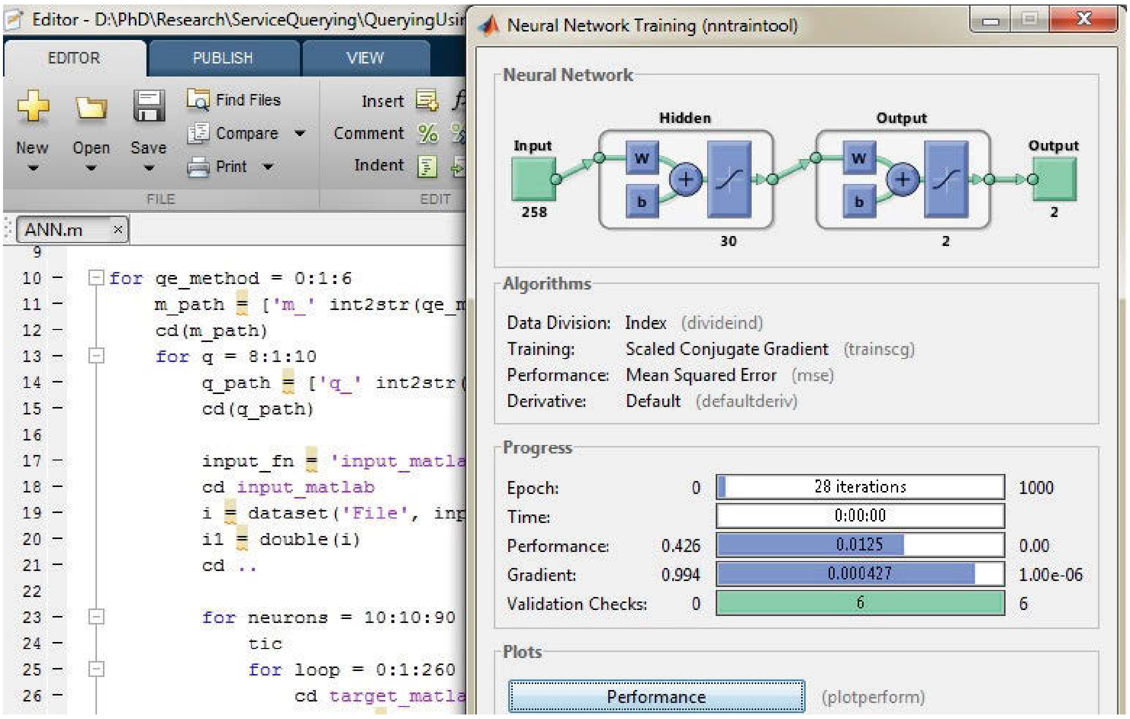


Figure 8.2 Matlab code and toolbox for creating Feed-Forward based service classifiers.

3) Protégé

This thesis proposed a semantic-based methodology for service retrieval. To make the methodology semantic, the service-domain ontology is applied for service annotation, service querying, and service retrieval. In terms of service annotation and service querying, services and queries are related to ontological service concepts. The service retrieval module then retrieves relevant services based on the relatedness between services and concepts, and between the query and concepts. To view and manage the ontology structure, an application called Protégé has been broadly used. The screen capture of the Protégé application is shown in Figure 8.3. The transport service ontology is opened and the service concept *"Airline_Airline_Agent"* is viewed.

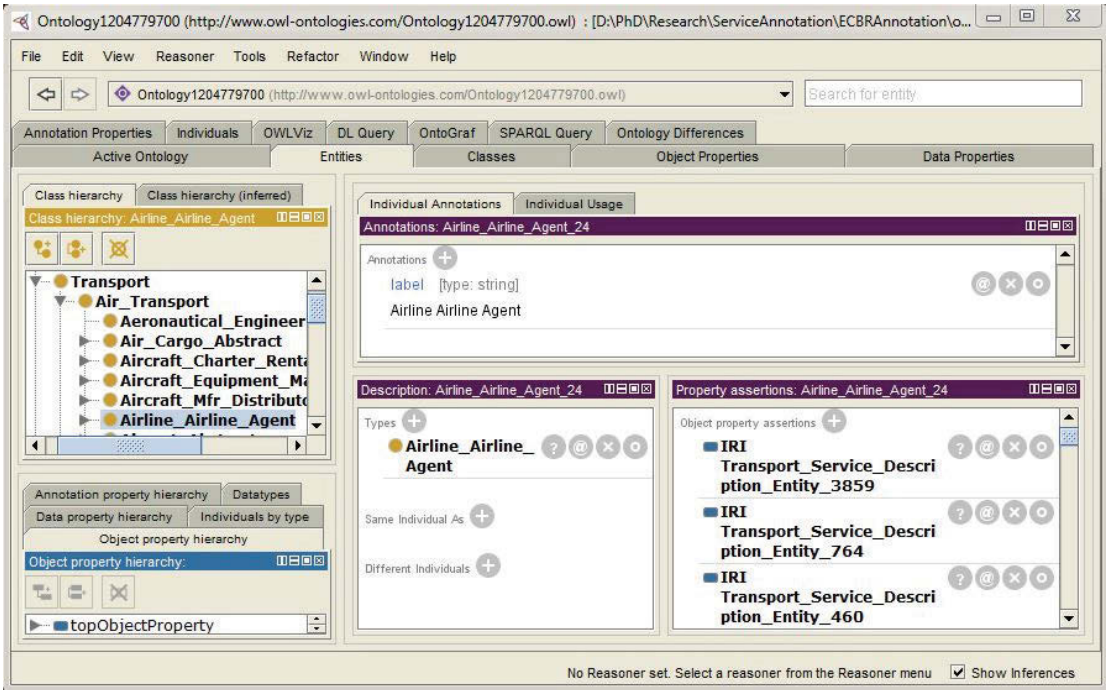


Figure 8.3 Transport service ontology in Protégé window.

8.2.2 Libraries

To accomplish the semantic service retrieval implementation, some Java libraries, such as Protégé OWL, WordNet, Word Similarity and Fuzzy Logic libraries are used.

1) Protégé OWL API

As mentioned in Chapter 3, the SDE metadata dataset and the transport service ontology provided by (Dong, Hussain & Chang 2011) are applied. Those datasets are in the form of W3C Web Ontology Language (OWL). To implement the semantic service annotation and the semantic service querying, SDE metadata and ontological service concepts need to be extracted from the datasets. Then, properties of those SDEs and concepts are applied for calculating the service-concept and the query-concept relatedness. In this thesis, the domain-specific ontology, like the transport service ontology, is connected by using Protégé OWL libraries. When the Protégé program is installed, the Protégé OWL libraries are already included in "\plugins\edu.stanford.smi.protegex.owl\" folder. The guideline of how to use Protégé OWL API is described in ProtégéWiki (Protege 2016).

2) WordNet API

To find the service-concept and query-concept relatedness, relevance scores between them are required. Although three different approaches are proposed; ECBR, Vector-based and Classification-based, for calculating the relevance scores, those proposed approaches have to consider synonyms of terms in services, concepts, and queries. In this thesis, WordNet 3.1 (Fellbaum 1998; Miller 1995; University 2016) is used as the thesaurus of the retrieval system. To access WordNet and collect the synonyms from it, a Java API called *"Java WordNet Library" (JWNL)* is applied (Walenz, Barton & Didion 2016).

3) Word Similarity Library

Based on the querying expansion module in Section 6.4 , a given query is firstly enlarged by using related terms. Then, the expanded query is matched to relevant concepts in the next step. In this thesis, WordNet-based and ontology-based query expansion approaches are proposed. In terms of the WordNet-based approach, the query is expanded by using synonyms from WordNet database. Typically, WordNet may provide synonyms of a word in various senses of words. As a result, the proposed WordNet-based query expansion approach is divided into two groups; All-senses expansion and Proper-sense expansion techniques. To expand queries by the latter technique, only synonyms of the most relevant word sense are applied for query expansion. We consider the similarity values between terms in the query in order to select the proper sense of the word. In this thesis, a Java API called *"WordNet Similarity for Java" (WS4J)* is applied to word similarity calculation (Shima 2016).

4) Fuzzy Logic Library

As described in Chapter 7, the semantic service retrieval part is categorized into Non-Fuzzy based and Fuzzy based approaches. Non-Fuzzy based approach retrieves all services that are annotated to relevant service concepts; meanwhile, Fuzzy-based approach considers that service-concept and query-concept relatedness is fuzzy and uses fuzzy rules to compute relevance scores between a query and services. In this thesis, a Java library called *"jFuzzyLogic"* is applied in order to define fuzzy-based information, such as fuzzy variables, membership functions, and fuzzy rules, and compute the service retrieval relevance scores (Cingolani 2016; Cingolani & Alcala-Fdez 2012; Cingolani & Alcalá-Fdez 2013).

8.3 Semantic service retrieval prototype

This section presents prototypes of the semantic service retrieval methodology, which comprise the prototype of semantic service annotation and the prototype of service querying and service retrieval.

8.3.1 Semantic service annotation prototype

The service annotation aims to semantically link SDE metadata to relevant ontological service concepts and is then used for retrieving relevant service metadata in the next step. In this section, a prototype which is developed for semantic service annotation is presented.

1) Prototype interface

The interface of the service annotation prototype is displayed in Figure 8.4. The prototype interface comprises three subtasks, namely, service ontology loader, service classifier loader, and service annotation tasks.

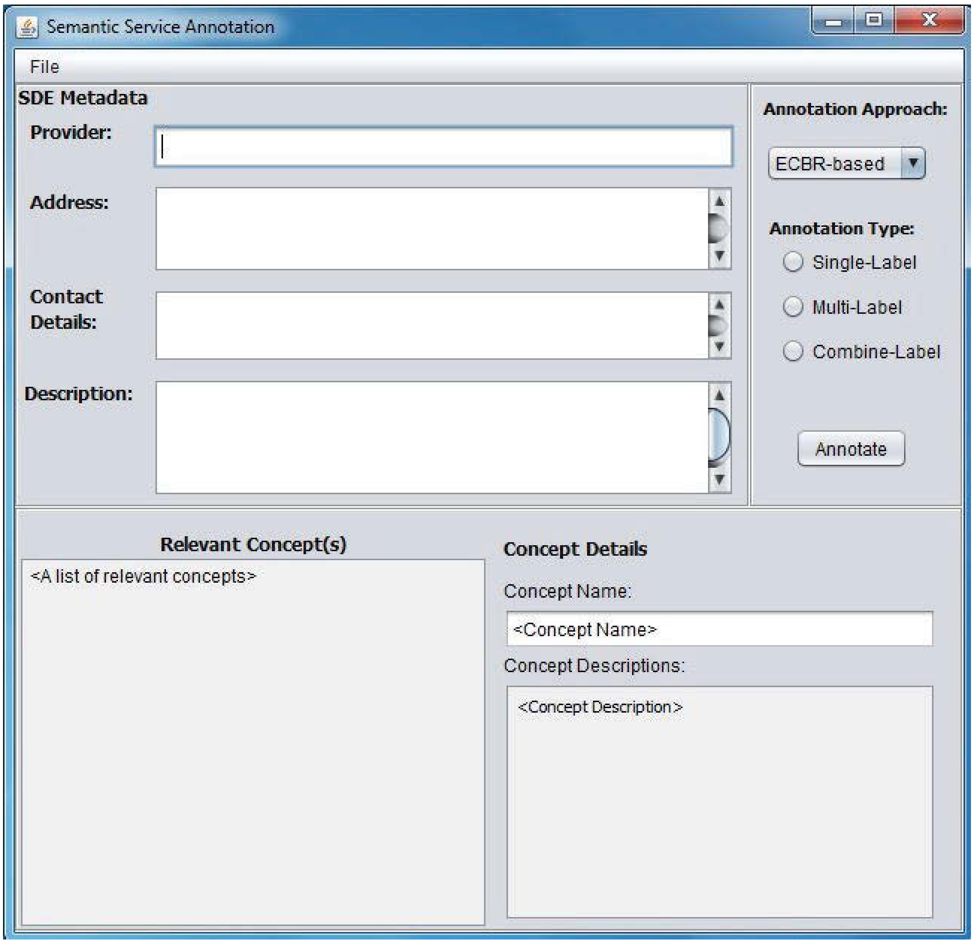


Figure 8.4 Semantic service annotation interface

2) Service ontology loader

The service ontology loader task aims to load the specific service ontology into the system. To load the ontology, the user has to select the menu *'File->Load'* and then clicks on the menu item *'Service Ontology'*. The interface of the service ontology loader menu is shown in Figure 8.5. Users need to define the location of domain-specific ontology, which is in the OWL format, to the annotation system. The ontology selection window is shown in Figure 8.6.

Figure 8.5 Service ontology loader menu

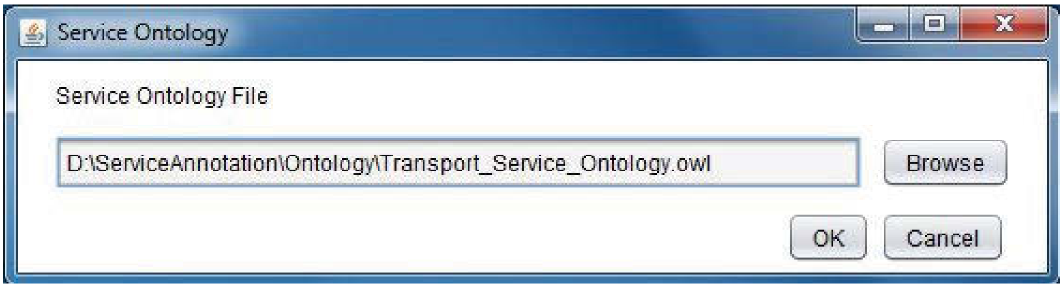


Figure 8.6 Ontology selection window

3) Service classifier loader

To annotate the services by using Classification-based approaches, the system firstly requires a user to define the trained service classifier. Similar to the service ontology loader, as shown in Figure 8.7, the user has to select menu *'File -> Load'* and then chooses *'Service Classifier'* menu item. Then, the service classifier selection window in Figure 8.8 is popped-up. The user firstly selects the type of the service annotation; single-label, multi-label or combine-label, and then defines the locations of service classifier files based on the classification algorithms; FF, RBF, KNN, CT, and SVM. Figure 8.8 shows the service classifier selection window of the multi-label based annotation approach. In the case of single-label and combine-label based approaches, only the locations of FF, RBF, KNN, and CT are provided. The service classifier files are in the mat format which is created by Matlab.

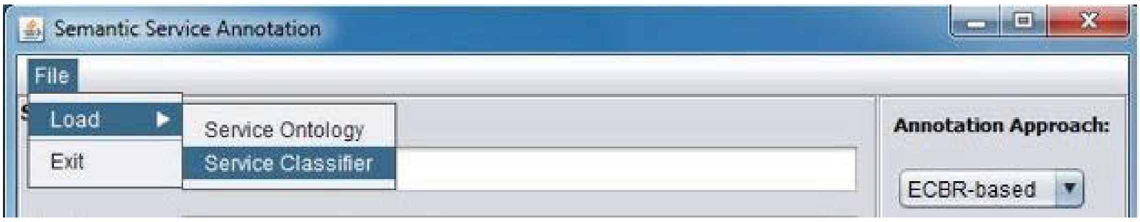


Figure 8.7 Service classifier loader menu

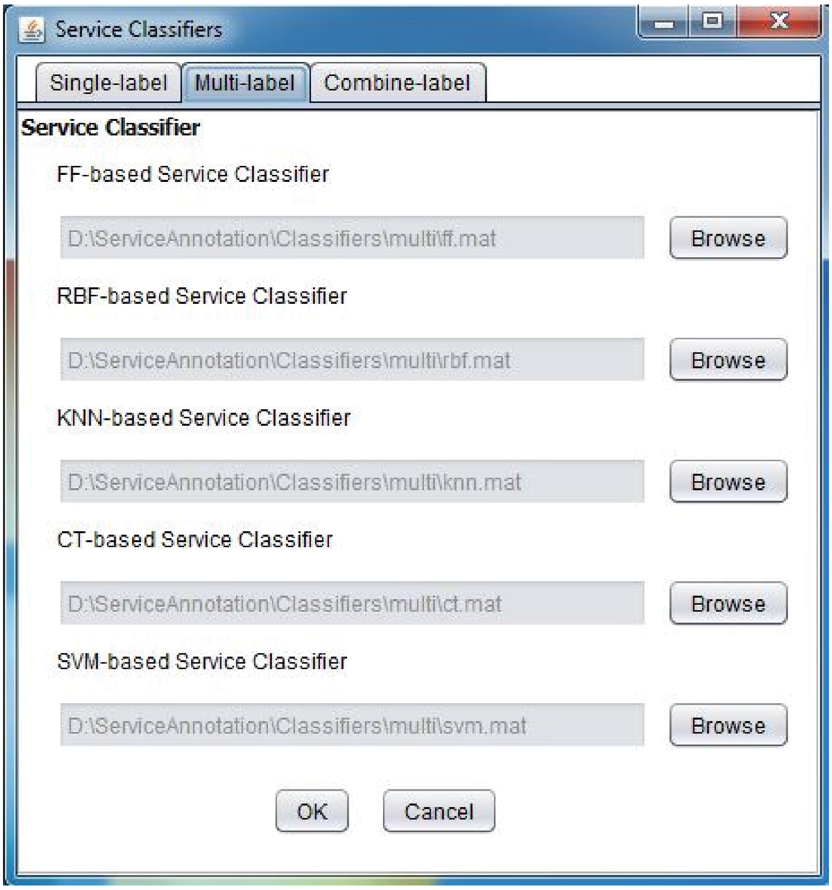


Figure 8.8 Service classifier selection window

4) Semantic service annotation

To annotate a service, a user has to put in the service information which consists of service name, service provider, service description, address and contact details. Then, the user selects the annotation approach (ECBR, VSM-based, EVSM-based, FF-based, RBF-based, KNN-based, CT-based or SVM-based approach) and annotation type (Single-label or multi-label connection type). Next, the user clicks on the button *'Annotate'* to get the service annotation results. Figure 8.9 shows the semantic service annotation interface with the information given from the user. The information of the service provider "Matthew's Labour Hire & Taxi Truck Services" is filled, and the FF-based annotation approach and the multi-label annotation type are selected.

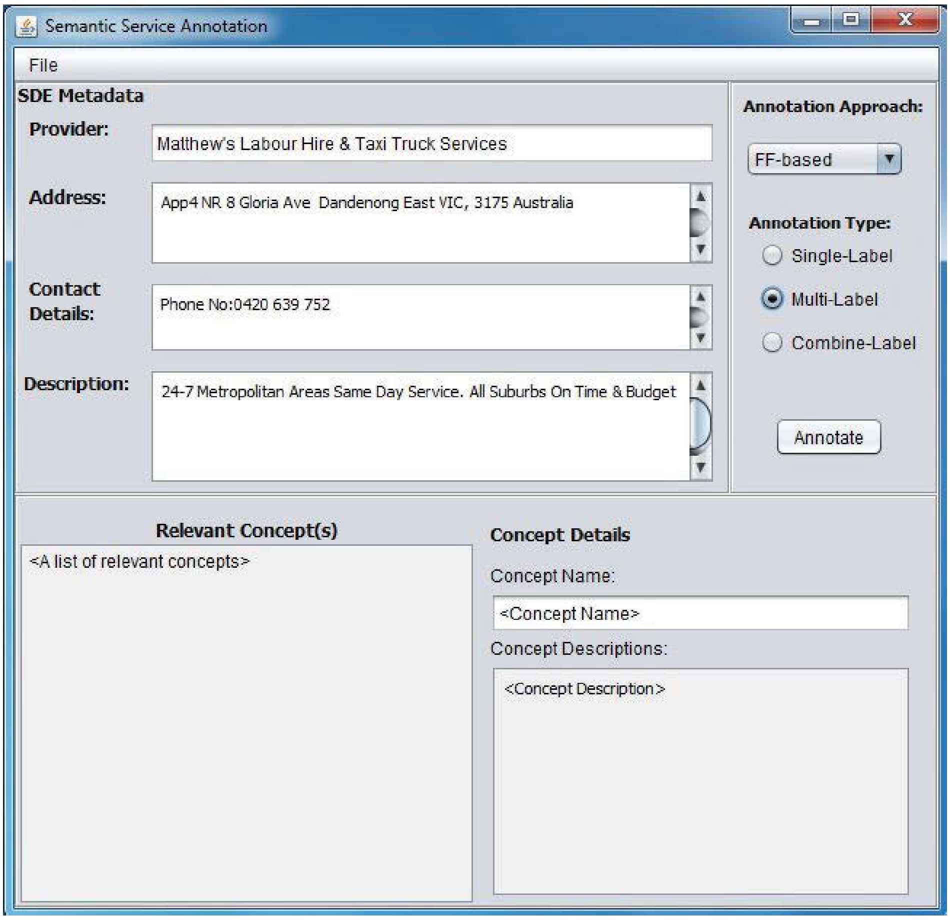


Figure 8.9 The semantic service annotation window with the given service information

Based on the chosen connection type, the output of the prototype system is a relevant service concept if the single-label type is selected. In contrast, with setting the connection type as multi-label or combine-label, the output is a set of relevant concepts. Moreover, the relevance score between the service and each relevant concept is also presented. When the user clicks on each service concept, the information of that concept, including concept name and concept descriptions, is shown in the right panel. Figure 8.10 displays the output of annotating the given service "Matthew's Labour Hire & Taxi Truck Services". In this example, the service is annotated to the concept "Taxi_Truck" and "Taxi_Pick-up", and descriptions of the concept "Taxi_Pick-up" are "Taxi Picking" and "Taxi Pick-up".

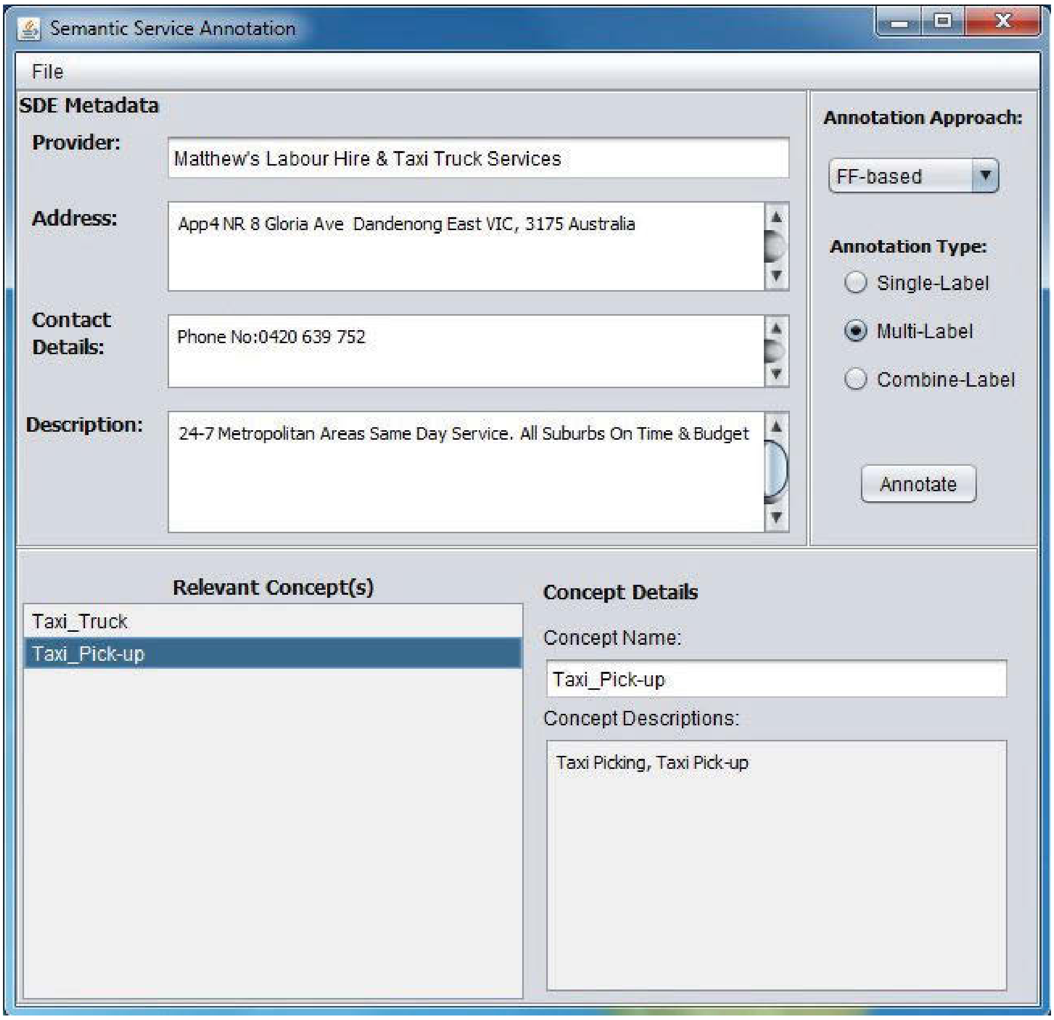
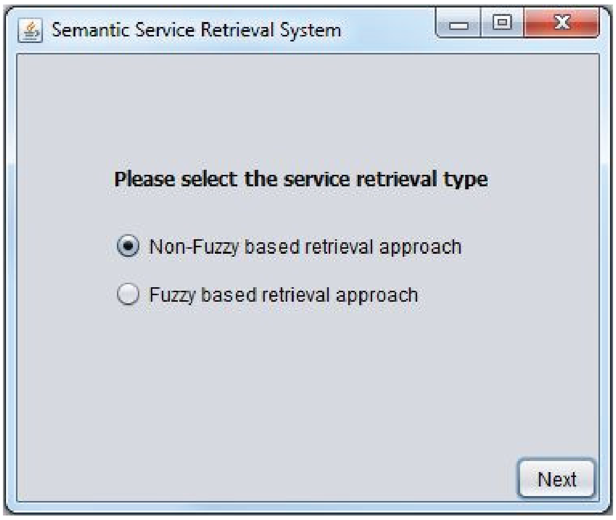


Figure 8.10 The output of annotating the provider "Matthew's Labour Hire & Taxi Truck Services"

8.3.2 Semantic service querying and service retrieval prototype

To retrieve relevant services, the service querying and the service retrieval parts are required. Given a query *Q*, the service querying part aims to get ontological concepts that are relevant to *Q*. Then, the service retrieval part returns relevant services based on the results of service annotation and service querying parts. In this case, a prototype that combines both parts together is developed because the service querying and the service retrieval are sub-tasks of semantic service retrieval.

Based on the proposed service retrieval approaches, a user firstly selects non-fuzzy based or fuzzy based approach, as shown in Figure 8.11.



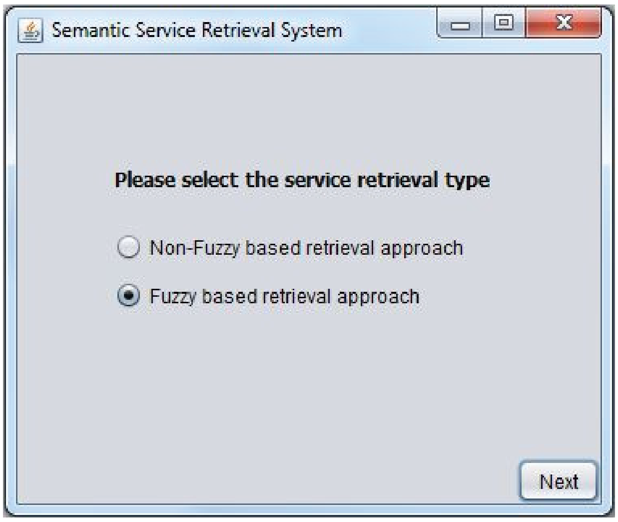


Figure 8.11 Service retrieval approach selection

1) Prototype interface

The interfaces of the non-fuzzy based and fuzzy based service retrieval prototype are shown in Figure 8.12 and Figure 8.13. Both non-fuzzy and fuzzy based approaches consist of four common tasks, namely, service ontology loader, annotation classifier loader, querying classifier loader and service retrieval tasks. In addition, the fuzzy based approach also has the fuzzy file loader task, and the fuzzy environment setting panel is added into the interface.

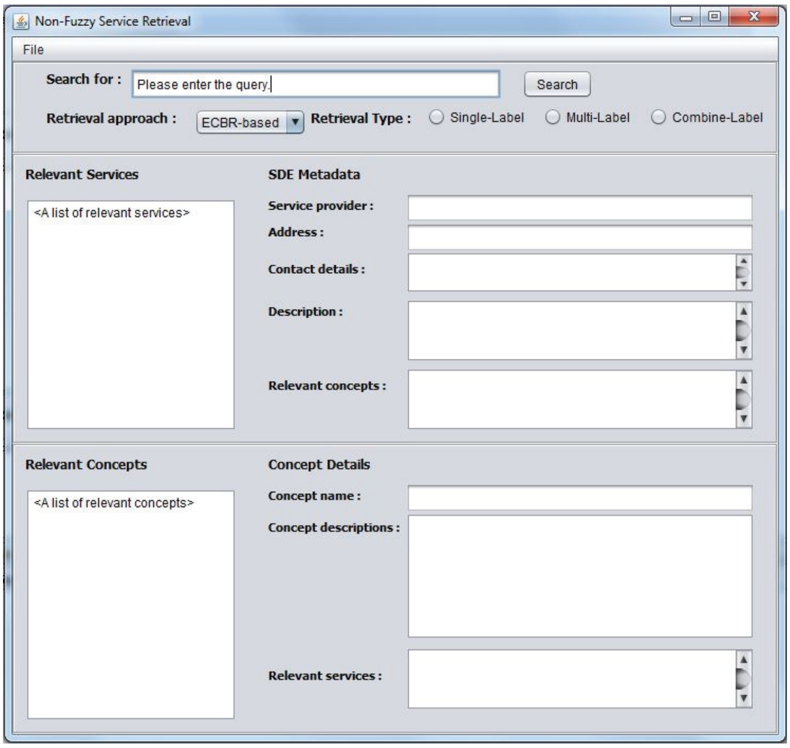


Figure 8.12 Non-fuzzy based service retrieval interface

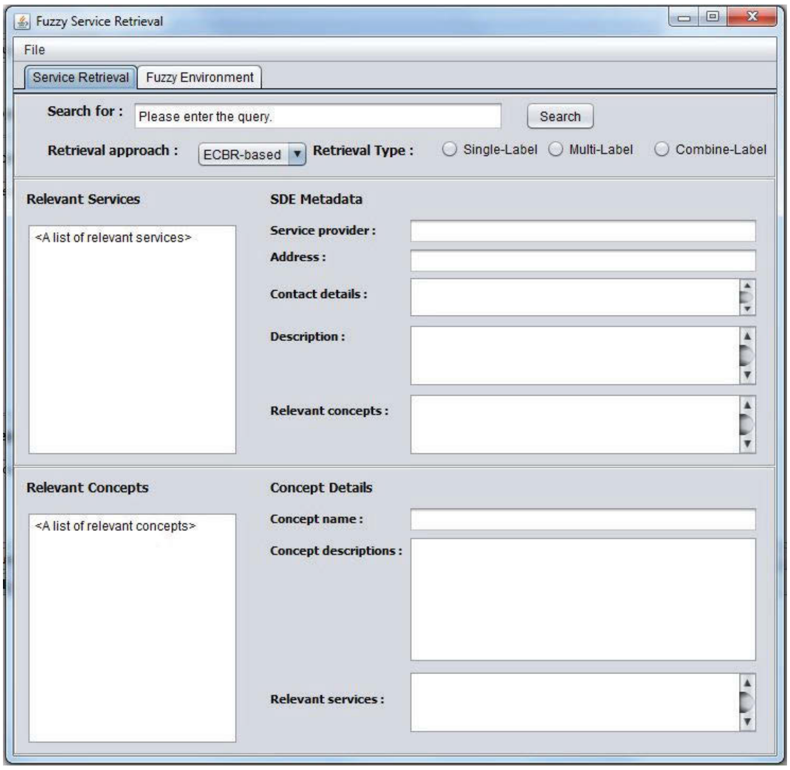


Figure 8.13 Fuzzy based service retrieval interface

2) Service ontology loader

Like the service ontology loader task in the service annotation, a user firstly requires defining the domain-specific service ontology by selecting the menu *"File -> Load -> Service Ontology"* and browsing the location of the ontology file. Both non-fuzzy and fuzzy based approaches follow the same steps to load the service ontology.

3) Classifier loader

To retrieving services that are relevant to a query, classifiers for service annotation and service querying are required for the Classification-based retrieval approach; FF, RBF, KNN, CT and SVM approaches. The service querying classifiers classify the query into relevant service concepts and the service annotation classifiers are used to get services that are relevant to those concepts. Both non-fuzzy and fuzzy based approaches follow the same steps to load the classifiers.

Regarding the service annotation classifiers, a user follows the menu *"File -> Load -> Classifiers -> Annotation Classifiers"* and then defines locations of the annotation classifiers. Likewise, to load the querying classifiers, the user has to follow the menu *"File -> Load -> Classifiers -> Querying Classifiers"* and browse the querying classifier files based on the classification-based methods. Examples of the classifier loader menu and the classifier selection window of the annotation classifiers and the querying classifiers are shown in Figure 8.14 - Figure 8.17 respectively.

Figure 8.14 Annotation classifier loader menu

Figure 8.15 Querying classifier loader menu

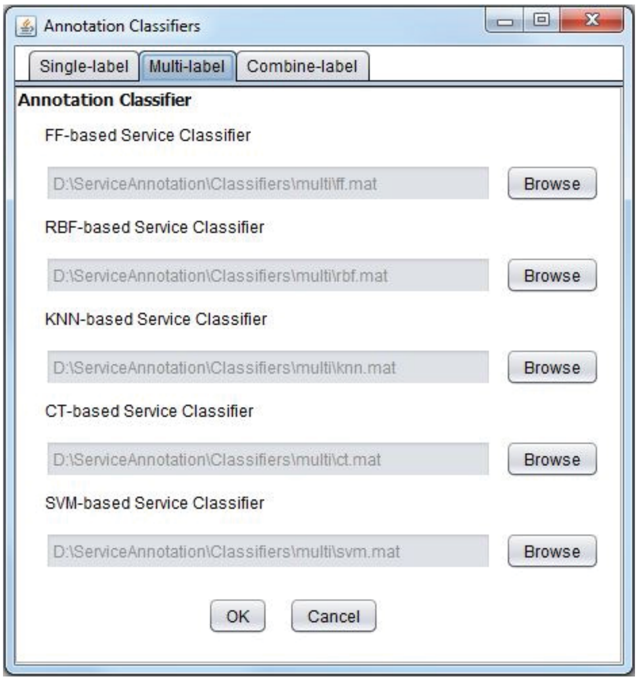


Figure 8.16 Annotation classifier selection window

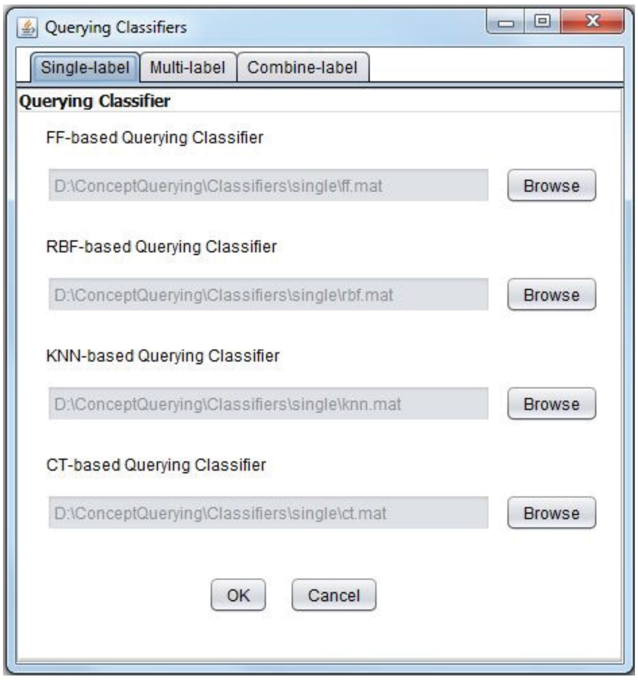


Figure 8.17 Querying classifier selection window

4) Fuzzy file loader

The fuzzy based retrieval approach applies fuzzy logic to calculate the relevance scores between a query and services. Fuzzy variables and fuzzy rules need to be defined in ".fcl" file which is used for supporting the "jFuzzyLogic" Java library.

The details of the fuzzy file creation are described in (jFuzzyLogic paper). In the case of the non-fuzzy based service retrieval, the fuzzy file loader task is not required.

To load a fuzzy file, a user follows the menu *"File -> Load -> Fuzzy file (.fcl)"* and then browses the location of the desired fuzzy file. The fuzzy file loader menu and fuzzy file selection window are shown in Figure 8.18 and Figure 8.19 respectively.

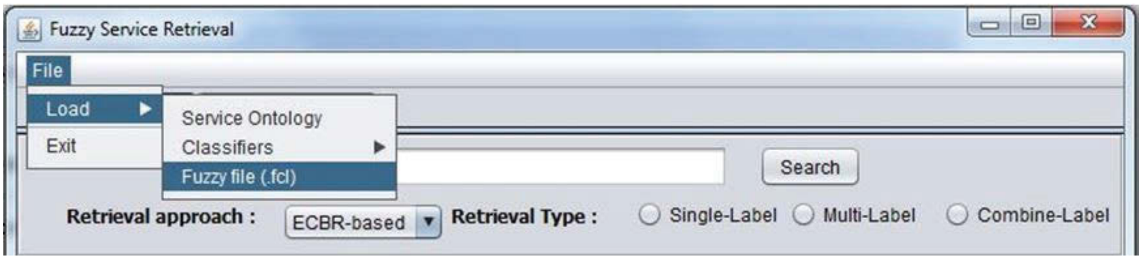


Figure 8.18 Fuzzy file loader menu

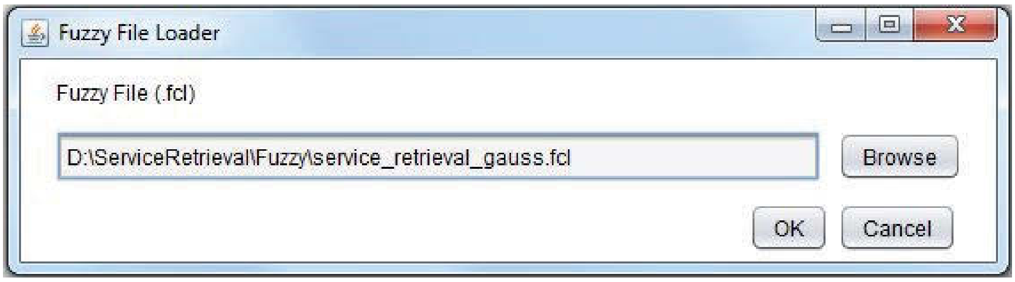


Figure 8.19 Fuzzy file selection window

5) Semantic service retrieval

To retrieve services, a user gives a query into "Search for" box and then selects the retrieval approach (ECBR, VSM-based, EVSM-based, FF-based, RBF-based, KNN-based, CT-based or SVM-based approach) and service querying type (Single-label or multi-label connection type). Next, the user clicks on the button *'Search'* to get services that are relevant to the query. Figure 8.20 and Figure 8.21 present the results of querying "Taxi cab" by using the non-fuzzy EVSM based and fuzzy EVSM based approaches for combine-label service retrieval.

Given the query "Taxi cab", the prototype firstly queries relevant service concepts and services that are relevant to those concepts are then retrieved. Regarding the non-fuzzy based approach (Figure 8.20), only the concept "Taxi" is queried. Seven services; such as "Yellow 13 CABS", "South Western Cabs" and so on, are retrieved because they are annotated to the concept “Taxi”. The details of each service and concept are shown in the right panel when a user clicks on a service and concept items in the left panel. In contrast, with the same query, the fuzzy-based approach queries the concept “Taxi”, “Taxi_Truck” and “Taxi_Pick-up” and eleven services are retrieved (Figure 8.21).

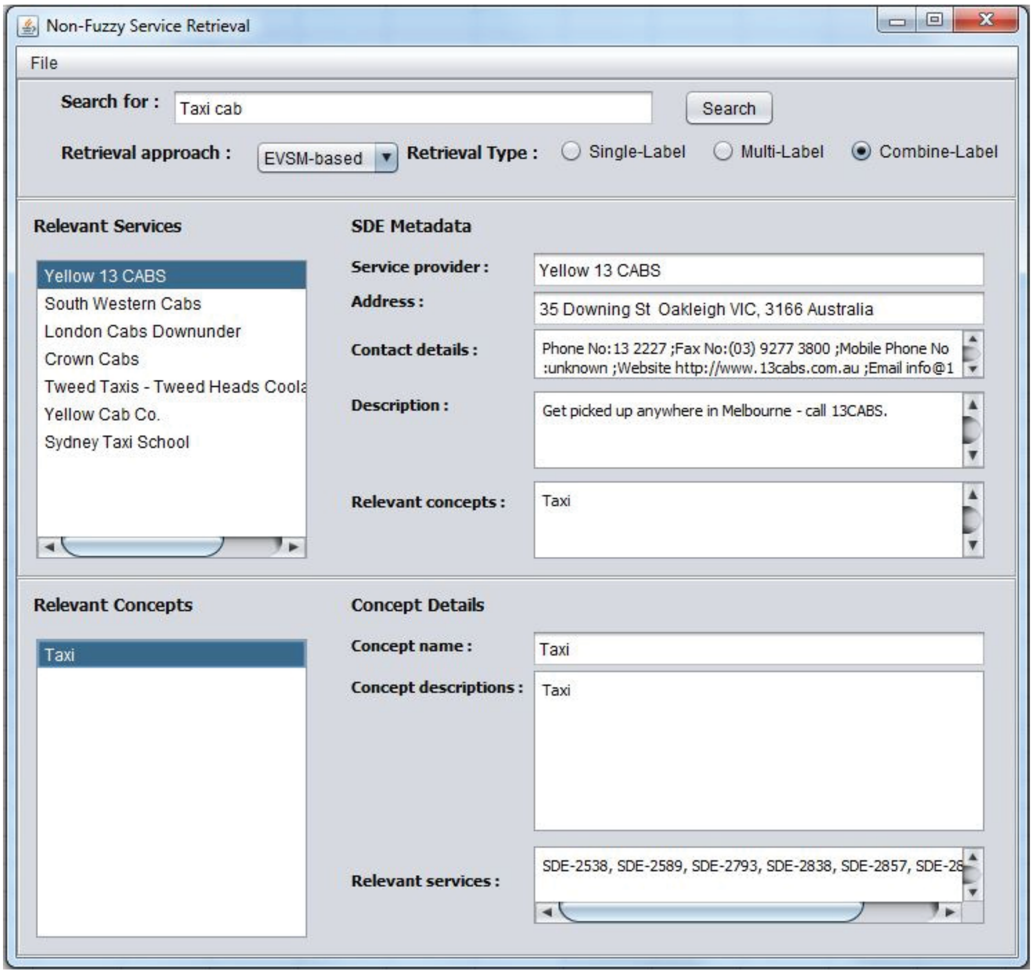


Figure 8.20 The output of the query "Taxi cab" by using non-fuzzy based approach

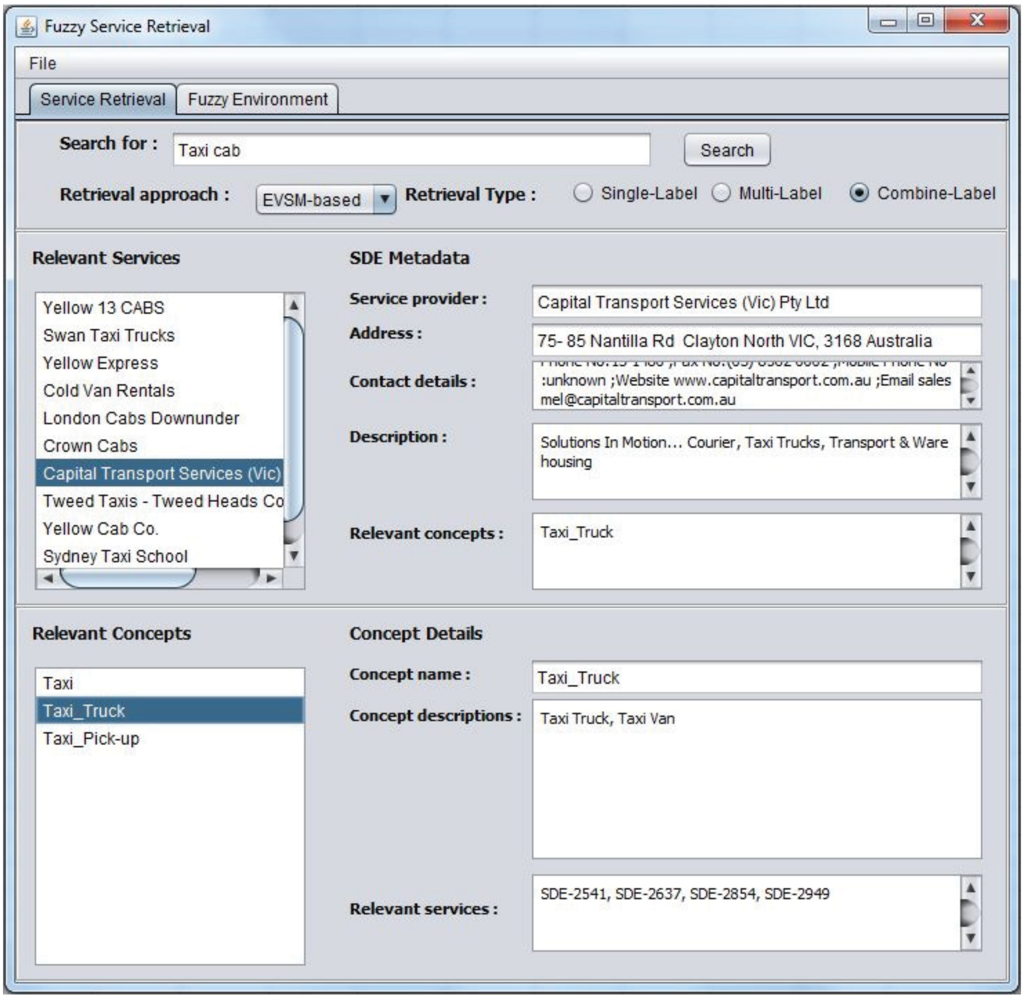


Figure 8.21 The output of the query "Taxi cab" by using fuzzy based approach

In addition, the fuzzy environment panel is also added into the service retrieval interface. As shown in Figure 8.22, fuzzy rules and membership functions of each fuzzy variable based on the loaded fuzzy file are presented.

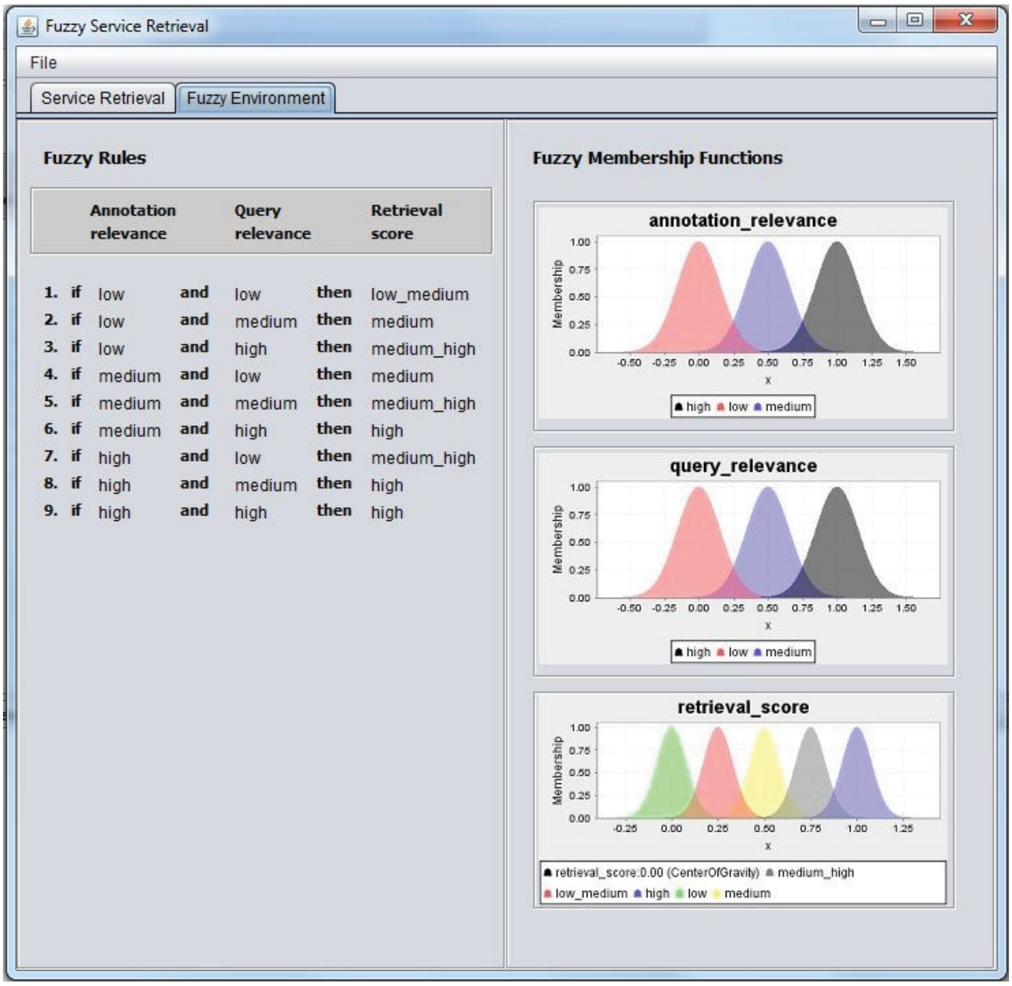


Figure 8.22 Fuzzy environment panel

8.4 Conclusion

This chapter presents the implementation of the proposed semantic service retrieval methodology. Programming tools and libraries, such as NetBeans IDE, Mathlab, Protégé OWL API, WordNet API, WS4J library and jFuzzyLogic library, were used for connecting the ontology, getting synonyms, semantically calculating word similarities and calculating fuzzy-based retrieval relevance score. The prototypes of the service annotation and the service retrieval system for the transport service domain were developed. The semantic service annotation prototype aims to annotate the given service information, whereas the semantic service retrieval system combines the service querying and service retrieval tasks together. That is, the prototype aims to get service concepts that are relevant to a given query and then retrieve relevant services based on those concepts.

Chapter 9 Conclusion and Future work

9.1 Introduction

We conclude the thesis in this chapter. The primary contribution of the thesis is to propose a semantic-based methodology for business service retrieval. Based on existing literature, the thesis identified the lack of an intelligent soft-computing based method for semantically retrieving services. To complete the service retrieval task, in this thesis, methodologies for semantic service annotation, semantic service querying, and semantic service retrieval were also developed and applied. Three method types, ECBR, Vector-based and Classification-based, and three service-concept connection types, single-label, multi-label and combine-label, are proposed for those methodologies. To validate the proposed methodologies, the prototypes of the proposed methodologies were developed and tested. Finally, the thesis tested the performances of all developed semantic-based approaches by using eight performance measures, namely, precision, recall, f-measure, fallout rate, annotation rate, querying rate, retrieval rate, and all-measures.

This remainder of this chapter is organized as follows: in Section 9.2 , the problems that were addressed in this thesis are briefly presented. Based on those research problems, the contributions of the thesis are subsequently summarized in Section 9.3 . The future work of this thesis is discussed in Section 9.4 .

9.2 Problems Addressed in this thesis

This thesis aims to develop the semantic-based methodology for service retrieval. To efficiently retrieve services from a user query, the semantic service annotation, semantic service querying, and semantic service retrieval methodologies are considered in the thesis. The semantic service annotation aims to understand the purposes or concepts of services, whereas the semantic service querying aims to get service concepts that are relevant to given queries. Given a query, the service querying methodology is used for getting relevant concepts and services that are annotated (by the service annotation methodology) to those concepts are then retrieved by the semantic service retrieval methodology.

Based on the literature review in Chapter 2, the research issues that were addressed in this thesis are summarized as follows:

1) Existing approaches for semantic service annotation and semantic service querying have focused on only Web services, not business service advertisements.

2) Existing work on web service annotation proposes semi-automated approaches which recommend annotations and still need service providers to select correct annotations. Consequently, the issues of time-consuming and up-to-date service information still remain.

3) The soft-computing techniques, which are applied in the complex real-world situation, have not been investigated for annotating the business service advertisements.

4) Existing work on semantic query expansion applies either a lexical database like WordNet or a domain-specific ontology to enlarge query terms. This is unsuitable for the real world tasks where the users possibly use terms based on both lexical synonyms and related concepts.

5) There is no existing work on a fuzzy semantic based approach for retrieving business service information.

6) Several semantic search methods just focus on the conceptual model. As a result, the validation process is not mentioned in many publications.

9.3 Contributions of this thesis to the existing literature

According to the identified research problems, the focus of the thesis is on developing an intelligent methodology for semantic annotation and semantic querying for online service advertising. The four principal contributions of this thesis are presented as follows:

9.3.1 Contribution 1: State of the art survey of present literature

This thesis reviews existing literature in the areas of semantic service annotation, semantic service crawling, semantic service querying, and semantic service retrieval for online services. To the best of our knowledge, the existing literature lacks such thorough review of the existing literature in these areas.

In terms of semantic service annotation, existing literature was categorized into semantic annotation for web services and semantic annotation for other services, such as library help desk and deep web resources. In addition, in this thesis, the annotation approaches proposed in the current work are also subdivided into non-ontology and ontology-based approaches. The non-ontology based approaches mainly use the idea of term-matching to find the relatedness between services and service concepts, whereas the ontology-based approaches mainly convert services into the ontology-based format and the relatedness between the services and the domain-specific ontology is calculated by using ontology alignment, machine learning, logical reasoning and heuristic-based techniques. The review showed that no research work focused on semantically annotating business service advertisement. Most of the current service annotation approaches are web service annotation and semi-automated.

Regarding the semantic querying in the service domain, the focus of the review in this thesis is comprised of two primary areas, namely semantic query expansion and semantic service retrieval. In terms of the semantic query expansion, the existing approaches expand a query by using synonyms of the query or using related terms from the domain-specific ontology. The review presented that there was no approach that applied both lexical synonyms and ontological terms to expand the query. In terms of the semantic service retrieval, most of the existing literature proposed semantic retrieval approaches for web services; meanwhile, few publications focused on other services, such as retrieving the traffic accident information and business service advertisements. Although there is research about semantic service advertisement retrieval, the proposed approach applied a term-based approach.

Moreover, soft-computing techniques have not been investigated for service retrieval.

9.3.2 Contribution 2: Semantic service annotation methodology

This thesis proposes the semantic-based methodology for annotating business services. This methodology aims to annotate services (SDEs) to relevant service concepts. The service annotation output is subsequently used for retrieving services.

The workflow of the proposed methodology is comprised of three main modules, namely term extraction, service-concept matching, and service-concept connection modules. Given a service and an ontological concept, a provider name and a description of the service are extracted into service terms by the term extraction module, whereas descriptions of the concept are extracted into concept terms. The service-concept matching module receives the service terms and the concept terms and then calculates the relevance score between the service and the concept. Then, the service-concept connection module links the service to the concept if the relevance score is greater than the defined relevance threshold.

In addition, this thesis proposes three types of semantic service annotation approaches; ECBR, Vector-based and Classification-based approaches. The main workflow of the proposed approaches is comprised of the same three main modules. The difference is in the way to calculate the relevance score. The ECBR approach applies the semantic term matching technique, whereas the Vector-based approach represents the service and the concept by using a vector and the relevance score is the cosine similarity between the service vector and the concept vector. In contrast, the Classification-based approach creates NN-based and ML-based service classifiers and uses them to classify the service to relevant service concepts.

To annotate the service, three annotation types, single-label, multi-label and combine-label, are proposed. The single-label annotation means the service is annotated to only the most relevant service concept; meanwhile, the multi-label annotation enables the services to be annotated to multiple service concepts. In the case that there is no relevant concept for the multi-label annotation, the combine-label annotation returns the output of the single-label annotation; otherwise, the output of the multi-label will be returned. That is, nine service annotation approaches are proposed in this thesis; 1) Single-label ECBR approach, 2) Multi-label ECBR approach, 3) Combine-label ECBR approach, 4) Single-label Vector-based approach, 5) Multi-label Vector-based approach, 6) Combine-label Vector-based approach, 7) Single-label Classification-based approach, 8) Multi-label Classification-based approach and 9) Combine-label Classification-based approach.

In addition, this is the first research that focuses on soft-computing based approaches for automated semantic annotation for online service information. Initial results and experimentation demonstrate the superiority of single-label and multi-label ECBR approaches over other annotation approaches. Unfortunately, the annotation rate of the ECBR approach is only 0.23% which is too low to apply this approach in the real world situation. This is because this thesis gives precision measure the most priority. The ECBR approach gives the best result when the annotation threshold is high; consequently, the number of annotated services is lower. Furthermore, the content of any SDE is short; as a result, it is difficult to indicate differences between services and classify them into correct service concepts.

9.3.3 Contribution 3: Semantic service querying methodology

This thesis proposes the semantic-based methodology for querying service concepts. Given a user query, this methodology aims to retrieve service concepts that are relevant to the query. The output of the service querying is subsequently used for retrieving services.

The workflow of the proposed methodology is comprised of three main modules; term extraction, query expansion, and querying modules. Given a query and an ontological concept, the query is firstly extracted into query terms by the term extraction module, whereas descriptions of the concept are extracted into concept terms. The query expansion module aims to enlarge the query terms in order to increase the correctness of retrieving relevant concepts. The output of this module is the expanded query terms. Next, the querying module receives the expanded query terms and the concept terms and then calculates the relevance score between the query and the service concept. Similar to the semantic service annotation methodology, the concept will be retrieved if the relevance score is greater than the querying relevance threshold.

In terms of the query expansion, this thesis proposes two primary semantic-based approaches for query expansion; WordNet-based and ontology-based approaches. The WordNet-based approach enlarges the query terms by using their synonyms which are obtained from WordNet; meanwhile, the ontology-based approach expands the query terms by using related terms which are obtained from the domain-specific ontology.

Regarding the querying module, the same as in the semantic service annotation, this thesis proposes three types of semantic service querying approaches; ECBR, Vector-based and Classification-based approaches. The working process of each service querying approach is the same as the process of the service annotation approach with the same type. Inputs of the service querying approaches are the expanded query terms and the concept terms and the approaches aim to calculate the query-concept relevance score, whereas the annotation approaches receive the service terms and the concept terms, and aim to calculate the service-concept relevance score. Furthermore, the thesis also proposes the hybrid-based service querying approach which combines the results of the querying approach with both WordNet-based and ontology-based query expansion methods.

Like the proposed service annotation approaches, the thesis proposes three querying types; single-label, multi-label and combine-label. That is, the most relevant concept is queried by the single-label based approach; meanwhile, all concepts whose relevance scores are more than the threshold are queried by the multi-label based approach. In contrast, the combine-based approach queries the most relevant concept if there is no concept whose relevance score is more than the threshold. In conclusion, the thesis proposes twenty-seven service querying approaches based on three query expansion methods (WordNet-based, ontology-based and Hybrid), three querying types (single-label, multi-label and combine-label) and three main querying approaches (ECBR, Vector, and Classification).

In addition, this is the first research that applies both WordNet-based and ontology-based methods to expand a given query. The experimental results demonstrate that the hybrid-based query expansion methods assist in the querying performance improvement over the WordNet-based or the ontology-based methods. Furthermore, the thesis is the first research that focuses on soft-computing based approaches for semantic querying for online service information. The experimental results show that the multi-label Classification-based approaches, especially the NN-based approaches, outperform other approaches. However, performances of the proposed service querying approaches may be improved if not only synonyms, but hypernyms (superordinate) and hyponyms (subordinate) are also used. Expanded queries by using superordinate and subordinate terms may reflect more precise relatedness between a query and a service concept. This is because the query may contain only general terms or specific terms of terms in the service concept. In this case, the relevance value from the proposed approaches in this thesis will be zero, although the query relates to the concept.

9.3.4 Contribution 4: Semantic service retrieval methodology

This thesis proposes the semantic-based methodology for service retrieval. Given a user query, this methodology aims to retrieve services that are relevant to the query. The workflow of the proposed methodology is comprised of three primary modules; service annotation, service querying, and service retrieval modules. Given a query, this methodology fetches the service annotations and relevant service concepts from the service annotation methodology and the service querying methodology respectively. Then, services that are annotated to the relevant concepts tend to be retrieved.

This thesis proposes two main types of service retrieval; non-fuzzy based and fuzzy based approaches. The non-fuzzy based approach retrieves services that are annotated to the relevant service concepts; meanwhile, the fuzzy based approach applies the fuzzy logic to calculate the retrieval score from the annotation relevance score and the querying relevance score. Services whose retrieval scores are greater than the defined retrieval threshold are then retrieved. Similar to the service annotation and the service querying, the service retrieval approaches are divided into single-label based, multi-label based and combine-label based approaches. The single-label service retrieval approach uses the service annotations and the relevant concepts from the single-label annotation and querying approaches respectively. This idea is also applied in the multi-label and combine-label retrieval approaches.

In conclusion, the thesis proposes fifty-four different service retrieval approaches based on two retrieval types (Non-fuzzy and fuzzy), three query expansion methods (WordNet-based, ontology-based and Hybrid), three annotation and querying types (single-label, multi-label and combine-label), and three main annotation and querying approaches (ECBR, Vector and Classification).

This is the first research that focuses on a fuzzy-based approach for semantic retrieval for online service information. The experimental results demonstrate that the fuzzy NN-based retrieval approach performs much better than the non-fuzzy NN-based approach, whereas overall results of the other fuzzy-based approaches are similar to the results of the non-fuzzy based approaches. Additionally, based on the proposed service annotation and the proposed service querying approaches, this thesis is the first research that compares the performance of various soft-computing approaches for semantic service retrieval.

Similar to the limitation of the service annotation, the retrieval rates of the proposed retrieval approaches are low, although average precision values are greater than 90%. This is because the proposed retrieval approaches retrieve services that are annotated to relevant service concepts. That is, the performances of service annotation and service querying approaches affect the service retrieval efficiency. Consequently, the retrieval rate will be low if the annotation rate is low.

Although this thesis does much research and proposes various semantic based service retrieval approaches, the additional research about service retrieval is still required. Some soft-computing methods; such as swarm intelligence and genetic algorithm, have not been applied. Furthermore, this thesis focuses on retrieving services in the transport service domain. To prove that the proposed approaches are suitable for any business services, service information from other service domains should be crawled and investigated; however, the thesis does not focus on semantically crawling online services.

9.4 Conclusion and Future work

To sum up, this thesis is the first research that proposes the soft-computing based methodology for service technologies, namely semantic service annotation, semantic querying, and semantic service querying. The experimental results demonstrate that the Classification-based approaches, especially NN-based approach outperform the others in all three methodologies. Additionally, expanding the query by using both synonyms from WordNet and related terms from the service ontology assists the querying and retrieval approaches to increase the performance. The fuzzy-based technique is able to improve the performance of the NN-based service retrieval approach.

Although much research about semantic service technologies has been investigated in this thesis, a lot of further work still needs to be explored. To continue working on this research topic, our future work is planned as follows:

1) The development of intelligent methods to increase the annotation rate and retrieval rate, is required. Although the proposed semantic-based approaches for service annotation and service retrieval performed well in precision, recall, f-measure and fallout rate, the percentages of annotated services and retrieved services were too low.

2) The development of the semantic service methodologies based on other soft-computing techniques is required. Although this thesis applies Neural Network based, Machine Learning based and Fuzzy based techniques for developing the various service methodologies, further techniques, use of other soft-computing methods such as swarm intelligence and genetic algorithm, needs to be investigated.

3) Using other semantic relations is required. This thesis focuses on the semantics from the synonyms in WordNet and the related terms in the service ontology. However, other semantic relations, such as hypernyms and hyponyms, needs to be considered.

4) The validation in other service domains is required. This thesis tests the performance of the prototype in transport service ontology. However, other service

domains, such as health services and mining services, should be also used for testing whether the proposed methodologies are suitable for other real-world situation.

5) The development of the semantic-based methodology for service crawling is required. This thesis focuses on only service annotation, service querying, and service retrieval. However, the service crawling, which gathers service information from the Web and collects it in the service knowledge base, should be researched to broaden the semantic service retrieval system.

6) Updating and gaining more service information is required. This thesis uses the dataset of service information provided by (Dong, Hussain & Chang 2011). However, the dataset needs to be updated because it was provided five years ago and it contains a lot of out-of-date service information. Consequently, much business service information in various business service directories, such as Yellow Pages and Nationwide Business Directory of Australia (Nationwide), needs to be crawled.

Appendix A Additional Experimental Results of the Semantic Service Querying

A.1 ECBR Querying Approach

A.1.1 Single-Label Service Querying

All experimental results for single-label ECBR-based querying approach are presented in Section 6.7.1 .

A.1.2 Multi-Label Service Querying

1) No Query Expansion (QE-0)

Table A.1 The experiments of ECBR approach without the query expansion for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	13.33%	98.64%	18.60%	-	99.16%	-
0.1	14.65%	95.94%	20.48%	-	97.32%	-
0.2	18.86%	90.44%	25.78%	-	94.37%	-
0.3	26.06%	83.30%	32.44%	-	84.94%	-
0.4	27.49%	76.24%	33.88%	-	78.00%	-
0.5	56.70%	81.32%	60.91%	0.53%	45.49%	65.76%
0.6	52.61%	75.60%	56.50%	0.55%	38.54%	61.56%
0.7	42.71%	50.13%	42.51%	0.29%	27.32%	50.03%
0.8	32.78%	35.86%	32.00%	0.31%	18.77%	41.06%
0.9	20.11%	22.03%	18.72%	0.35%	16.05%	31.51%

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.2 The experiments of ECBR approach with QE-1 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	10.62%	98.73%	15.24%	-	99.96%	-
0.1	11.76%	96.91%	16.88%	-	99.52%	-
0.2	16.98%	92.27%	23.48%	-	97.92%	-
0.3	25.42%	88.12%	32.06%	-	89.94%	-
0.4	26.91%	81.71%	33.41%	-	84.27%	-
0.5	58.75%	87.07%	63.66%	0.84%	56.27%	69.42%

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.6	55.96%	82.70%	60.63%	0.88%	49.52%	66.18%
0.7	48.17%	63.66%	50.27%	0.37%	35.74%	56.66%
0.8	38.62%	51.37%	40.61%	0.41%	24.20%	47.81%
0.9	23.90%	33.70%	24.80%	0.46%	20.05%	36.27%

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.3 The experiments of ECBR approach with QE-2 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	11.99%	98.65%	16.78%	-	99.80%	-
0.1	13.06%	96.46%	18.32%	-	99.12%	-
0.2	17.95%	91.28%	24.39%	-	96.81%	-
0.3	25.95%	86.64%	32.32%	-	88.06%	-
0.4	26.81%	79.73%	33.30%	-	81.99%	-
0.5	58.48%	84.70%	62.84%	0.80%	52.76%	68.32%
0.6	55.10%	79.70%	59.27%	0.85%	46.13%	64.68%
0.7	46.74%	59.22%	47.97%	0.35%	32.75%	54.63%
0.8	37.08%	46.50%	38.10%	0.39%	22.44%	45.83%
0.9	23.04%	29.89%	23.05%	0.44%	18.73%	34.90%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.4 The experiments of ECBR approach with QE-3 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	9.64%	99.14%	13.28%	-	99.16%	-
0.1	9.83%	98.72%	13.57%	-	98.92%	-
0.2	10.39%	97.62%	14.45%	-	98.48%	-
0.3	11.08%	96.29%	15.56%	-	96.85%	-
0.4	13.74%	93.49%	19.66%	-	95.49%	-
0.5	39.75%	89.26%	47.58%	1.31%	84.27%	63.87%
0.6	39.43%	85.50%	47.00%	1.18%	81.63%	62.71%
0.7	44.11%	77.29%	50.68%	0.61%	75.16%	63.02%
0.8	42.19%	70.72%	48.41%	0.50%	70.61%	60.26%
0.9	34.47%	53.53%	38.35%	0.47%	68.09%	52.71%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.5 The experiments of ECBR approach with QE-4 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	10.90%	98.87%	15.31%	-	99.16%	-
0.1	11.31%	97.67%	15.88%	-	98.92%	-
0.2	13.89%	95.76%	19.41%	-	96.81%	-
0.3	15.38%	93.19%	21.27%	-	94.21%	-
0.4	18.98%	90.63%	26.29%	-	90.62%	-
0.5	46.75%	84.66%	53.45%	0.82%	76.24%	65.73%
0.6	45.85%	81.27%	52.21%	0.77%	72.40%	64.11%
0.7	48.28%	69.78%	52.99%	0.37%	64.02%	62.28%
0.8	41.58%	57.50%	44.90%	0.34%	57.47%	55.56%
0.9	24.73%	33.02%	25.43%	0.38%	48.68%	40.90%

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.6 The experiments of ECBR approach with QE-5 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	7.86%	99.18%	11.11%	-	99.96%	-
0.1	9.04%	98.41%	12.73%	-	99.84%	-
0.2	11.61%	97.15%	16.34%	-	99.00%	-
0.3	18.52%	95.73%	25.54%	-	96.01%	-
0.4	30.86%	93.64%	39.10%	-	89.62%	-
0.5	52.41%	90.30%	59.39%	0.75%	80.27%	70.35%
0.6	54.83%	85.42%	61.10%	0.56%	73.60%	69.87%
0.7	52.63%	72.19%	56.58%	0.35%	61.78%	64.58%
0.8	56.50%	67.83%	57.93%	0.26%	35.46%	61.74%
0.9	31.96%	34.56%	30.80%	0.31%	19.17%	40.42%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.7 The experiments of ECBR approach with QE-6 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	10.01%	98.88%	14.06%	-	99.80%	-
0.1	11.41%	97.45%	16.00%	-	99.52%	-
0.2	15.48%	95.03%	21.32%	-	97.52%	-
0.3	20.67%	92.47%	27.67%	-	92.77%	-

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.4	32.16%	89.51%	39.93%	-	85.90%	-
0.5	51.89%	84.89%	57.91%	0.67%	72.80%	68.00%
0.6	53.60%	78.93%	58.51%	0.52%	64.22%	66.61%
0.7	45.97%	61.10%	48.51%	0.35%	50.48%	57.35%
0.8	43.62%	52.95%	43.94%	0.31%	27.52%	51.06%
0.9	31.23%	33.02%	29.76%	0.30%	18.85%	39.69%

A.1.3 Combine-Label Service Querying

Regarding combine-label ECBR-based approach, this thesis focuses on combining the best results of the single-label querying and the multi-label querying. All experimental results for combine-label ECBR-based querying approach are presented in Section 6.7.1 .

A.2 VSM-based Querying Approach (Vector-based Approach)

A.2.1 Single-Label Service Querying

All experimental results for single-label VSM-based querying approach are presented in Section 6.7.2 .

A.2.2 Multi-Label Service Querying

1) No Query Expansion (QE-0)

Table A.8 The experiments of VSM-based approach without query expansion for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	13.39%	97.43%	18.79%	-	97.56%	-
0.1	13.52%	97.05%	19.02%	-	97.48%	-
0.2	16.04%	94.72%	22.23%	-	96.69%	-
0.3	19.80%	89.13%	26.80%	-	95.13%	-
0.4	23.92%	84.93%	31.36%	-	93.21%	-
0.5	39.51%	80.76%	46.02%	-	84.86%	-
0.6	51.60%	78.39%	55.88%	-	76.40%	-
0.7	56.63%	72.85%	58.78%	-	59.74%	-
0.8	72.37%	80.04%	72.18%	-	31.71%	-
0.9	66.61%	66.48%	64.92%	0.20%	11.38%	63.03%

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.9 The experiments of VSM-based approach with QE-1 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	10.80%	97.68%	15.44%	-	99.52%	-
0.1	11.28%	96.99%	16.15%	-	99.36%	-
0.2	14.17%	94.03%	20.02%	-	98.88%	-
0.3	21.43%	89.90%	28.70%	-	97.84%	-
0.4	33.40%	87.26%	41.00%	-	93.65%	-
0.5	56.02%	87.43%	61.48%	0.73%	79.19%	71.51%
0.6	69.47%	87.52%	73.00%	0.34%	57.59%	75.45%
0.7	71.28%	81.90%	72.80%	0.25%	33.67%	71.73%
0.8	81.34%	83.27%	80.45%	0.16%	13.34%	74.07%
0.9	77.10%	75.24%	74.06%	0.13%	4.27%	68.86%

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.10 The experiments of VSM-based approach with QE-2 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	11.84%	97.47%	16.60%	-	98.60%	-
0.1	12.05%	97.10%	17.02%	-	98.44%	-
0.2	14.61%	94.54%	20.46%	-	97.88%	-
0.3	20.80%	89.47%	27.71%	-	96.96%	-
0.4	28.57%	85.61%	36.19%	-	93.93%	-
0.5	50.54%	83.97%	55.95%	0.86%	82.79%	68.49%
0.6	63.72%	83.34%	67.31%	0.42%	65.93%	72.91%
0.7	67.53%	77.45%	68.44%	0.28%	43.73%	70.41%
0.8	78.95%	81.31%	77.89%	0.18%	18.21%	73.16%
0.9	72.82%	70.23%	69.91%	0.15%	5.95%	66.02%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.11 The experiments of VSM-based approach with QE-3 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	9.27%	99.07%	12.93%	-	97.92%	-
0.1	9.49%	98.73%	13.31%	-	97.88%	-
0.2	10.80%	97.98%	15.44%	-	97.80%	-
0.3	15.92%	95.12%	23.41%	-	97.52%	-
0.4	29.02%	87.15%	38.02%	-	96.77%	-

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.5	50.15%	85.03%	56.26%	-	82.35%	-
0.6	61.22%	82.22%	65.10%	-	59.11%	-
0.7	66.55%	83.89%	69.82%	0.35%	35.06%	69.88%
0.8	67.38%	75.54%	67.90%	0.29%	20.65%	66.52%
0.9	69.46%	67.38%	65.79%	0.17%	10.94%	64.38%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.12 The experiments of VSM-based approach with QE-4 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	10.62%	98.59%	15.11%	-	97.92%	-
0.1	10.80%	98.25%	15.41%	-	97.88%	-
0.2	12.05%	97.13%	17.27%	-	97.80%	-
0.3	14.86%	93.95%	21.14%	-	97.52%	-
0.4	20.94%	90.80%	28.54%	-	96.65%	-
0.5	35.22%	87.51%	43.86%	-	92.37%	-
0.6	45.89%	83.29%	53.81%	0.68%	84.35%	66.47%
0.7	53.11%	78.98%	59.13%	0.43%	73.64%	67.94%
0.8	73.19%	79.98%	73.41%	0.20%	47.16%	74.33%
0.9	73.06%	71.04%	70.09%	0.13%	21.96%	68.67%

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.13 The experiments of VSM-based approach with QE-5 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	7.63%	99.11%	10.92%	-	98.24%	-
0.1	8.87%	98.54%	12.81%	-	98.20%	-
0.2	13.08%	97.25%	18.92%	-	97.84%	-
0.3	22.39%	94.33%	30.51%	-	97.28%	-
0.4	39.53%	91.46%	48.16%	-	94.73%	-
0.5	62.46%	91.09%	68.10%	-	81.99%	-
0.6	77.42%	92.38%	80.58%	0.25%	53.83%	79.95%
0.7	83.17%	87.52%	82.62%	0.12%	25.60%	77.61%
0.8	84.32%	81.03%	79.93%	0.08%	8.11%	74.08%
0.9	95.00%	92.57%	91.66%	0.03%	2.00%	80.93%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.14 The experiments of VSM-based approach with QE-6 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	9.69%	98.56%	13.81%	-	98.16%	-
0.1	10.46%	98.12%	14.98%	-	98.12%	-
0.2	13.43%	96.24%	19.16%	-	97.80%	-
0.3	19.13%	93.74%	26.12%	-	96.96%	-
0.4	27.51%	90.28%	35.91%	-	94.61%	-
0.5	41.47%	86.70%	50.00%	-	89.66%	-
0.6	57.67%	84.55%	63.16%	0.47%	78.19%	71.88%
0.7	79.57%	86.00%	79.98%	0.16%	54.87%	79.93%
0.8	78.66%	76.77%	76.00%	0.10%	22.88%	72.80%
0.9	91.33%	87.43%	87.27%	0.05%	3.91%	78.32%

A.2.3 Combine-Label Service Querying

Regarding combine-label VSM-based approach, this thesis focuses on combining the best results of the single-label querying and the multi-label querying. All experimental results for combine-label VSM-based querying approach are presented in Section 6.7.2 .

A.3 EVSM-based Querying Approach (Vector-based Approach)

A.3.1 Single-Label Service Querying

All experimental results for single-label VSM-based querying approach are presented in Section 6.7.2 .

A.3.2 Multi-Label Service Querying

1) No Query Expansion (QE-0)

Table A.15 The experiments of EVSM-based approach without query expansion for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	12.12%	97.53%	17.16%	-	97.96%	-
0.1	12.61%	96.92%	17.84%	-	97.84%	-
0.2	15.11%	94.41%	21.31%	-	97.12%	-
0.3	22.80%	89.57%	30.31%	-	95.65%	-
0.4	31.03%	84.89%	38.20%	-	91.13%	-
0.5	46.61%	82.52%	52.04%	1.16%	78.00%	65.35%
0.6	54.10%	77.83%	57.84%	0.69%	61.54%	66.12%
0.7	53.21%	66.86%	55.30%	0.43%	40.26%	60.58%
0.8	71.81%	77.29%	71.70%	0.25%	15.54%	68.37%
0.9	58.55%	55.38%	55.26%	0.18%	4.67%	55.69%

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.16 The experiments of EVSM-based approach with QE-1 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	10.04%	97.69%	14.58%	-	99.52%	-
0.1	10.56%	97.31%	15.29%	-	99.48%	-
0.2	12.75%	95.05%	18.49%	-	99.16%	-
0.3	17.63%	91.60%	24.47%	-	98.48%	-
0.4	23.74%	88.32%	31.26%	-	96.81%	-
0.5	39.01%	86.19%	46.01%	-	91.61%	-
0.6	52.33%	84.08%	57.72%	-	82.15%	-
0.7	61.88%	81.13%	64.52%	-	64.46%	-
0.8	74.22%	85.94%	74.64%	-	38.54%	-
0.9	79.82%	85.57%	79.95%	0.14%	16.37%	74.19%

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.17 The experiments of EVSM-based approach with QE-2 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	10.88%	97.59%	15.52%	-	98.68%	-
0.1	11.45%	97.03%	16.22%	-	98.64%	-

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.2	13.50%	95.02%	19.28%	-	98.08%	-
0.3	19.52%	90.94%	26.35%	-	97.24%	-
0.4	26.36%	86.85%	33.35%	-	94.49%	-
0.5	42.86%	86.08%	49.32%	-	85.18%	-
0.6	56.09%	84.22%	60.29%	-	73.00%	-
0.7	61.22%	80.14%	63.56%	-	53.19%	-
0.8	69.73%	79.77%	69.41%	-	28.47%	-
0.9	67.85%	73.34%	67.24%	0.22%	10.10%	64.71%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.18 The experiments of EVSM-based approach with QE-3 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	8.26%	99.04%	11.67%	-	98.24%	-
0.1	8.85%	98.74%	12.54%	-	98.24%	-
0.2	11.22%	97.73%	16.28%	-	98.12%	-
0.3	19.17%	93.36%	27.09%	-	97.80%	-
0.4	37.19%	86.82%	45.07%	-	93.01%	-
0.5	57.09%	85.53%	61.99%	-	68.29%	-
0.6	65.45%	88.01%	69.89%	0.46%	42.69%	71.20%
0.7	67.08%	81.18%	69.68%	0.30%	24.60%	68.11%
0.8	67.33%	70.67%	66.19%	0.20%	13.10%	64.40%
0.9	67.62%	60.62%	61.12%	0.14%	4.87%	61.02%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.19 The experiments of EVSM-based approach with QE-4 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	9.55%	98.60%	13.74%	-	98.24%	-
0.1	10.05%	98.28%	14.42%	-	98.24%	-
0.2	12.08%	96.75%	17.42%	-	98.12%	-
0.3	17.62%	93.03%	24.31%	-	97.80%	-
0.4	25.95%	90.13%	33.48%	-	95.65%	-
0.5	40.37%	88.51%	48.98%	1.10%	86.58%	64.59%
0.6	50.08%	82.78%	57.47%	0.58%	72.96%	66.93%
0.7	64.57%	79.59%	67.67%	0.28%	52.08%	70.69%
0.8	74.76%	77.04%	73.89%	0.14%	27.56%	71.66%
0.9	71.68%	66.57%	67.00%	0.12%	7.83%	64.86%

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.20 The experiments of EVSM-based approach with QE-5 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	6.97%	99.12%	10.09%	-	98.24%	-
0.1	8.69%	98.55%	12.79%	-	98.24%	-
0.2	12.39%	97.23%	18.02%	-	98.00%	-
0.3	19.72%	94.29%	27.36%	-	97.68%	-
0.4	34.46%	90.79%	43.31%	-	96.09%	-
0.5	56.69%	89.29%	62.58%	-	87.42%	-
0.6	76.58%	92.88%	79.31%	0.37%	64.86%	81.13%
0.7	83.48%	91.33%	84.06%	0.18%	32.27%	79.51%
0.8	82.55%	83.75%	79.66%	0.14%	9.90%	74.00%
0.9	95.28%	91.24%	90.50%	0.03%	2.12%	80.69%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.21 The experiments of EVSM-based approach with QE-6 for multi-label service querying

QT	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0	8.81%	98.63%	12.77%	-	98.24%	-
0.1	10.12%	98.14%	14.64%	-	98.24%	-
0.2	12.97%	96.22%	18.63%	-	97.92%	-
0.3	19.00%	93.63%	25.83%	-	97.16%	-
0.4	29.13%	90.65%	37.32%	-	94.97%	-
0.5	43.15%	87.75%	51.32%	-	87.46%	-
0.6	60.15%	85.67%	65.09%	0.58%	74.84%	72.81%
0.7	79.69%	86.70%	79.97%	0.20%	49.72%	79.30%
0.8	79.70%	80.49%	78.02%	0.13%	16.97%	73.18%
0.9	87.74%	83.69%	82.95%	0.08%	2.12%	75.40%

A.3.3 Combine-Label Service Querying

Regarding combine-label VSM-based approach, this thesis focuses on combining the best results of the single-label querying and the multi-label querying. All experimental results for combine-label VSM-based querying approach are presented in Section 6.7.2 .

A.4 FF-based Querying Approach (Classification-based Approach)

A.4.1 Single-Label Service Querying

1) No Query Expansion (QE-0)

Table A.22 The experiments of FF-based approach without query expansion for single-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	54.60%	54.60%	54.60%	0.17%	100.00%	68.19%
20	57.10%	57.10%	57.10%	0.16%	100.00%	69.95%
30	59.33%	59.33%	59.33%	0.16%	100.00%	71.51%
40	63.51%	63.51%	63.51%	0.14%	100.00%	74.44%
50	63.51%	63.51%	63.51%	0.14%	100.00%	74.44%
60	60.72%	60.72%	60.72%	0.15%	100.00%	72.48%
70	61.00%	61.00%	61.00%	0.15%	100.00%	72.68%
80	65.46%	65.46%	65.46%	0.13%	100.00%	75.80%
90	61.56%	61.56%	61.56%	0.15%	100.00%	73.07%

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.23 The experiments of FF-based approach with QE-1 for single-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	79.67%	79.67%	79.67%	0.08%	100.00%	85.76%
20	78.55%	78.55%	78.55%	0.08%	100.00%	84.97%
30	77.99%	77.99%	77.99%	0.08%	100.00%	84.58%
40	77.99%	77.99%	77.99%	0.08%	100.00%	84.58%
50	79.94%	79.94%	79.94%	0.08%	100.00%	85.95%
60	78.55%	78.55%	78.55%	0.08%	100.00%	84.97%
70	81.34%	81.34%	81.34%	0.07%	100.00%	86.93%
80	79.67%	79.67%	79.67%	0.08%	100.00%	85.76%
90	77.99%	77.99%	77.99%	0.08%	100.00%	84.58%

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.24 The experiments of FF-based approach with QE-2 for single-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	72.14%	72.14%	72.14%	0.11%	100.00%	80.48%
20	72.98%	72.98%	72.98%	0.10%	100.00%	81.07%
30	77.72%	77.72%	77.72%	0.09%	100.00%	84.39%
40	76.60%	76.60%	76.60%	0.09%	100.00%	83.61%
50	76.88%	76.88%	76.88%	0.09%	100.00%	83.80%
60	77.44%	77.44%	77.44%	0.09%	100.00%	84.19%
70	75.49%	75.49%	75.49%	0.09%	100.00%	82.83%
80	76.04%	76.04%	76.04%	0.09%	100.00%	83.21%
90	74.09%	74.09%	74.09%	0.10%	100.00%	81.85%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.25 The experiments of FF-based approach with QE-3 for single-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	61.84%	61.84%	61.84%	0.15%	100.00%	73.27%
20	68.25%	68.25%	68.25%	0.12%	100.00%	77.76%
30	69.64%	69.64%	69.64%	0.12%	100.00%	78.73%
40	68.52%	68.52%	68.52%	0.12%	100.00%	77.95%
50	70.47%	70.47%	70.47%	0.11%	100.00%	79.31%
60	68.52%	68.52%	68.52%	0.12%	100.00%	77.95%
70	69.64%	69.64%	69.64%	0.12%	100.00%	78.73%
80	69.36%	69.36%	69.36%	0.12%	100.00%	78.53%
90	70.19%	70.19%	70.19%	0.11%	100.00%	79.12%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.26 The experiments of FF-based approach with QE-4 for single-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	64.62%	64.62%	64.62%	0.14%	100.00%	75.21%
20	67.97%	67.97%	67.97%	0.12%	100.00%	77.56%
30	66.57%	66.57%	66.57%	0.13%	100.00%	76.58%
40	68.52%	68.52%	68.52%	0.12%	100.00%	77.95%
50	67.13%	67.13%	67.13%	0.13%	100.00%	76.97%
60	67.41%	67.41%	67.41%	0.13%	100.00%	77.17%

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
70	67.13%	67.13%	67.13%	0.13%	100.00%	76.97%
80	69.64%	69.64%	69.64%	0.12%	100.00%	78.73%
90	69.92%	69.92%	69.92%	0.12%	100.00%	78.93%

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.27 The experiments of FF-based approach with QE-5 for single-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	75.49%	75.49%	75.49%	0.09%	100.00%	82.83%
20	80.78%	80.78%	80.78%	0.07%	100.00%	86.54%
30	80.50%	80.50%	80.50%	0.07%	100.00%	86.34%
40	80.78%	80.78%	80.78%	0.07%	100.00%	86.54%
50	80.50%	80.50%	80.50%	0.07%	100.00%	86.34%
60	81.62%	81.62%	81.62%	0.07%	100.00%	87.12%
70	80.78%	80.78%	80.78%	0.07%	100.00%	86.54%
80	79.11%	79.11%	79.11%	0.08%	100.00%	85.37%
90	79.39%	79.39%	79.39%	0.08%	100.00%	85.56%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.28 The experiments of FF-based approach with QE-6 for single-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	72.42%	72.42%	72.42%	0.11%	100.00%	72.42%
20	77.99%	77.99%	77.99%	0.08%	100.00%	77.99%
30	80.78%	80.78%	80.78%	0.07%	100.00%	80.78%
40	80.22%	80.22%	80.22%	0.08%	100.00%	80.22%
50	77.72%	77.72%	77.72%	0.09%	100.00%	77.72%
60	77.99%	77.99%	77.99%	0.08%	100.00%	77.99%
70	78.55%	78.55%	78.55%	0.08%	100.00%	78.55%
80	79.67%	79.67%	79.67%	0.08%	100.00%	79.67%
90	77.99%	77.99%	77.99%	0.08%	100.00%	77.99%

A.4.2 Multi-Label Service Querying

1) No Query Expansion (QE-0)

Table A.29 The experiments of FF-based approach without query expansion for multi-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	84.93%	83.43%	82.61%	0.08%	37.50%	79.49%
20	87.28%	85.26%	85.13%	0.06%	42.55%	81.84%
30	86.46%	85.54%	84.59%	0.06%	38.30%	80.84%
40	87.07%	87.05%	85.71%	0.08%	39.36%	81.63%
50	86.61%	84.84%	84.39%	0.06%	42.55%	81.40%
60	86.96%	86.18%	85.61%	0.06%	42.82%	81.97%
70	88.25%	86.34%	86.44%	0.06%	40.16%	82.23%
80	86.52%	86.55%	85.16%	0.07%	47.61%	82.50%
90	-	-	-	-	0.00%	-

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.30 The experiments of FF-based approach with QE-1 for multi-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	94.74%	92.73%	92.37%	0.04%	35.37%	85.96%
20	92.30%	92.00%	91.30%	0.04%	45.48%	86.23%
30	95.22%	94.12%	93.43%	0.03%	45.21%	88.00%
40	91.96%	92.87%	91.59%	0.05%	47.61%	86.59%
50	93.73%	92.52%	92.21%	0.05%	51.33%	87.89%
60	95.05%	94.03%	93.14%	0.03%	46.54%	88.07%
70	92.73%	91.73%	90.94%	0.05%	44.15%	86.11%
80	93.72%	92.09%	91.87%	0.03%	52.93%	88.02%
90	-	-	-	-	0.00%	-

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.31 The experiments of FF-based approach with QE-2 for multi-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	92.83%	91.69%	90.94%	0.05%	33.24%	84.51%
20	92.86%	92.37%	91.44%	0.05%	42.82%	86.13%
30	93.80%	93.91%	92.41%	0.06%	45.74%	87.32%
40	93.84%	94.88%	93.07%	0.06%	49.20%	88.10%

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
50	90.60%	91.82%	89.60%	0.07%	46.81%	85.46%
60	92.08%	91.18%	90.45%	0.05%	43.09%	85.53%
70	90.34%	89.70%	88.47%	0.06%	36.17%	83.28%
80	89.94%	92.34%	89.39%	0.07%	47.07%	85.29%
90	78.57%	72.32%	73.02%	0.08%	3.72%	68.78%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.32 The experiments of FF-based approach with QE-3 for multi-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	90.92%	89.22%	88.42%	0.05%	49.20%	85.39%
20	90.73%	88.01%	88.22%	0.05%	49.73%	85.18%
30	94.02%	91.96%	91.63%	0.04%	52.66%	88.04%
40	90.84%	89.75%	89.07%	0.05%	60.64%	87.25%
50	91.97%	91.82%	90.49%	0.05%	54.52%	87.31%
60	92.29%	90.03%	89.85%	0.04%	54.52%	87.07%
70	91.42%	90.43%	89.42%	0.05%	59.04%	87.39%
80	92.38%	91.27%	90.53%	0.04%	55.59%	87.55%
90	92.26%	90.06%	90.21%	0.04%	57.71%	87.60%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.33 The experiments of FF-based approach with QE-4 for multi-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	90.06%	90.29%	88.85%	0.06%	52.66%	85.79%
20	90.68%	88.35%	88.22%	0.05%	54.79%	85.97%
30	91.12%	89.47%	88.98%	0.05%	54.79%	86.43%
40	92.93%	92.39%	91.81%	0.05%	55.85%	88.17%
50	90.87%	90.98%	89.76%	0.06%	61.17%	87.63%
60	91.58%	91.20%	90.17%	0.05%	53.46%	86.85%
70	91.71%	90.16%	89.37%	0.06%	54.52%	86.78%
80	92.31%	91.30%	90.68%	0.04%	56.38%	87.67%
90	-	-	-	-	0.00%	-

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.34 The experiments of FF-based approach with QE-5 for multi-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	92.26%	91.42%	90.14%	0.05%	56.12%	87.55%
20	91.19%	89.78%	88.97%	0.05%	53.99%	86.38%
30	93.44%	92.52%	91.86%	0.04%	58.78%	88.84%
40	93.48%	95.64%	93.45%	0.04%	61.17%	89.93%
50	94.71%	93.73%	93.27%	0.03%	61.17%	90.11%
60	94.47%	93.48%	92.90%	0.03%	67.29%	90.83%
70	92.07%	93.75%	91.57%	0.05%	65.43%	89.43%
80	91.13%	90.98%	90.23%	0.04%	66.49%	88.60%
90	93.97%	93.89%	93.08%	0.03%	67.55%	90.76%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.35 The experiments of FF-based approach with QE-6 for multi-label service querying

#Neural	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
10	93.05%	93.81%	91.86%	0.09%	50.53%	87.64%
20	90.55%	90.43%	88.40%	0.09%	58.78%	86.85%
30	93.89%	93.27%	91.99%	0.06%	61.17%	89.51%
40	90.60%	90.04%	88.70%	0.08%	59.84%	87.02%
50	89.03%	89.50%	87.48%	0.08%	64.36%	86.80%
60	90.04%	91.73%	89.56%	0.07%	62.23%	87.53%
70	93.88%	94.19%	92.38%	0.06%	54.52%	88.71%
80	92.30%	91.91%	90.98%	0.06%	59.31%	88.24%
90	-	-	-	-	0.00%	-

A.4.3 Combine-Label Service Querying

Regarding combine-label FF-based approach, this thesis focuses on combining the best results of the single-label querying and the multi-label querying. All experimental results for combine-label FF-based querying approach are presented in Section 6.7.3 .

A.5 RBF-based Querying Approach (Classification-based Approach)

A.5.1 Single-Label Service Querying

1) No Query Expansion (QE-0)

Table A.36 The experiments of RBF-based approach without query expansion for single-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	45.96%	45.96%	45.96%	0.21%	100.00%	62.14%
0.2	46.52%	46.52%	46.52%	0.21%	100.00%	62.53%
0.3	45.96%	45.96%	45.96%	0.21%	100.00%	62.14%
0.4	45.96%	45.96%	45.96%	0.21%	100.00%	62.14%
0.5	45.96%	45.96%	45.96%	0.21%	100.00%	62.14%
0.6	45.96%	45.96%	45.96%	0.21%	100.00%	62.14%
0.7	43.18%	43.18%	43.18%	0.22%	100.00%	60.19%
0.8	39.28%	39.28%	39.28%	0.23%	100.00%	57.46%
0.9	30.36%	30.36%	30.36%	0.27%	100.00%	51.21%

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.37 The experiments of RBF-based approach with QE-1 for single-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	65.46%	65.46%	65.46%	0.13%	100.00%	75.80%
0.2	65.18%	65.18%	65.18%	0.13%	100.00%	75.61%
0.3	64.90%	64.90%	64.90%	0.13%	100.00%	75.41%
0.4	64.90%	64.90%	64.90%	0.13%	100.00%	75.41%
0.5	64.90%	64.90%	64.90%	0.13%	100.00%	75.41%
0.6	64.90%	64.90%	64.90%	0.13%	100.00%	75.41%
0.7	63.23%	63.23%	63.23%	0.14%	100.00%	74.24%
0.8	61.84%	61.84%	61.84%	0.15%	100.00%	73.27%
0.9	54.32%	54.32%	54.32%	0.18%	100.00%	68.00%

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.38 The experiments of RBF-based approach with QE-2 for single-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	57.94%	57.94%	57.94%	0.16%	100.00%	70.53%
0.2	57.94%	57.94%	57.94%	0.16%	100.00%	70.53%
0.3	57.94%	57.94%	57.94%	0.16%	100.00%	70.53%
0.4	57.94%	57.94%	57.94%	0.16%	100.00%	70.53%
0.5	57.94%	57.94%	57.94%	0.16%	100.00%	70.53%
0.6	58.77%	58.77%	58.77%	0.16%	100.00%	71.12%
0.7	55.43%	55.43%	55.43%	0.17%	100.00%	68.78%
0.8	52.37%	52.37%	52.37%	0.18%	100.00%	66.63%
0.9	43.45%	43.45%	43.45%	0.22%	100.00%	60.38%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.39 The experiments of RBF-based approach with QE-3 for single-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	64.35%	64.35%	64.35%	0.14%	100.00%	75.02%
0.2	65.46%	65.46%	65.46%	0.13%	100.00%	75.80%
0.3	66.02%	66.02%	66.02%	0.13%	100.00%	76.19%
0.4	66.02%	66.02%	66.02%	0.13%	100.00%	76.19%
0.5	66.02%	66.02%	66.02%	0.13%	100.00%	76.19%
0.6	66.02%	66.02%	66.02%	0.13%	100.00%	76.19%
0.7	66.02%	66.02%	66.02%	0.13%	100.00%	76.19%
0.8	64.90%	64.90%	64.90%	0.13%	100.00%	75.41%
0.9	63.51%	63.51%	63.51%	0.14%	100.00%	74.44%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.40 The experiments of RBF-based approach with QE-4 for single-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	66.02%	66.02%	66.02%	0.13%	100.00%	76.19%
0.2	66.30%	66.30%	66.30%	0.13%	100.00%	76.39%
0.3	66.30%	66.30%	66.30%	0.13%	100.00%	76.39%
0.4	66.30%	66.30%	66.30%	0.13%	100.00%	76.39%
0.5	66.30%	66.30%	66.30%	0.13%	100.00%	76.39%
0.6	66.30%	66.30%	66.30%	0.13%	100.00%	76.39%

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.7	66.02%	66.02%	66.02%	0.13%	100.00%	76.19%
0.8	64.35%	64.35%	64.35%	0.14%	100.00%	75.02%
0.9	62.67%	62.67%	62.67%	0.14%	100.00%	73.85%

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.41 The experiments of RBF-based approach with QE-5 for single-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	73.82%	73.82%	73.82%	0.10%	100.00%	81.66%
0.2	75.21%	75.21%	75.21%	0.10%	100.00%	82.63%
0.3	75.77%	75.77%	75.77%	0.09%	100.00%	83.03%
0.4	75.77%	75.77%	75.77%	0.09%	100.00%	83.03%
0.5	75.77%	75.77%	75.77%	0.09%	100.00%	83.03%
0.6	75.77%	75.77%	75.77%	0.09%	100.00%	83.03%
0.7	75.49%	75.49%	75.49%	0.09%	100.00%	82.83%
0.8	74.93%	74.93%	74.93%	0.10%	100.00%	82.44%
0.9	73.54%	73.54%	73.54%	0.10%	100.00%	81.46%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.42 The experiments of RBF-based approach with QE-6 for single-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	77.16%	77.16%	77.16%	0.09%	100.00%	84.00%
0.2	76.88%	76.88%	76.88%	0.09%	100.00%	83.80%
0.3	76.88%	76.88%	76.88%	0.09%	100.00%	83.80%
0.4	76.88%	76.88%	76.88%	0.09%	100.00%	83.80%
0.5	76.88%	76.88%	76.88%	0.09%	100.00%	83.80%
0.6	76.88%	76.88%	76.88%	0.09%	100.00%	83.80%
0.7	76.32%	76.32%	76.32%	0.09%	100.00%	83.41%
0.8	73.54%	73.54%	73.54%	0.10%	100.00%	81.46%
0.9	69.92%	69.92%	69.92%	0.12%	100.00%	78.93%

A.5.2 Multi-Label Service Querying

1) No Query Expansion (QE-0)

Table A.43 The experiments of RBF-based approach without query expansion for multi-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	60.02%	64.03%	59.31%	0.47%	71.01%	68.09%
0.2	68.85%	69.31%	67.12%	0.19%	59.57%	71.91%
0.3	68.85%	69.31%	67.12%	0.19%	59.57%	71.91%
0.4	68.85%	69.31%	67.12%	0.19%	59.57%	71.91%
0.5	80.89%	80.48%	78.20%	0.13%	39.63%	77.08%
0.6	90.82%	89.80%	89.80%	0.04%	13.03%	80.22%
0.7	100.00%	100.00%	100.00%	0.00%	2.66%	85.40%
0.8	-	-	-	-	0.00%	-
0.9	-	-	-	-	0.00%	-

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.44 The experiments of RBF-based approach with QE-1 for multi-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	74.80%	80.53%	74.94%	0.60%	88.83%	81.48%
0.2	78.86%	81.61%	78.36%	0.15%	82.45%	82.88%
0.3	78.86%	81.61%	78.36%	0.15%	82.45%	82.88%
0.4	78.86%	81.61%	78.36%	0.15%	82.45%	82.88%
0.5	81.57%	82.53%	80.74%	0.10%	75.00%	83.35%
0.6	92.56%	93.87%	92.70%	0.04%	47.07%	87.06%
0.7	94.44%	94.58%	93.77%	0.03%	19.15%	83.90%
0.8	100.00%	100.00%	100.00%	0.00%	2.66%	85.40%
0.9	100.00%	100.00%	100.00%	0.00%	0.27%	85.04%

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.45 The experiments of RBF-based approach with QE-2 for multi-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	70.06%	75.84%	69.62%	0.89%	84.04%	77.32%
0.2	73.68%	76.60%	72.69%	0.39%	77.13%	78.38%
0.3	73.68%	76.60%	72.69%	0.39%	77.13%	78.38%
0.4	74.20%	77.13%	73.19%	0.39%	76.60%	78.66%

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.5	86.67%	87.78%	85.01%	0.31%	60.64%	84.64%
0.6	93.56%	93.93%	93.11%	0.03%	31.65%	85.22%
0.7	97.06%	95.17%	95.67%	0.01%	9.04%	83.80%
0.8	100.00%	89.35%	89.78%	0.00%	2.39%	82.23%
0.9	100.00%	100.00%	100.00%	0.00%	0.27%	85.04%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.46 The experiments of RBF-based approach with QE-3 for multi-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	74.54%	75.36%	73.30%	0.16%	86.17%	80.02%
0.2	78.92%	77.11%	76.86%	0.11%	80.32%	81.70%
0.3	78.92%	77.11%	76.86%	0.11%	80.32%	81.70%
0.4	78.92%	77.11%	76.86%	0.11%	80.32%	81.70%
0.5	79.53%	77.80%	77.61%	0.10%	78.19%	81.84%
0.6	86.29%	84.57%	84.43%	0.07%	69.68%	85.31%
0.7	90.84%	89.07%	89.05%	0.06%	59.31%	86.94%
0.8	90.73%	89.02%	88.92%	0.06%	50.00%	85.47%
0.9	93.06%	88.83%	89.13%	0.06%	38.03%	84.61%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.47 The experiments of RBF-based approach with QE-4 for multi-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	76.31%	77.75%	75.36%	0.47%	84.04%	81.03%
0.2	79.31%	78.22%	77.53%	0.45%	79.26%	81.91%
0.3	79.31%	78.22%	77.53%	0.45%	79.26%	81.91%
0.4	79.31%	78.22%	77.53%	0.45%	79.26%	81.91%
0.5	80.93%	79.84%	79.19%	0.45%	75.53%	82.49%
0.6	88.21%	86.20%	86.09%	0.06%	64.10%	85.73%
0.7	95.93%	94.26%	94.29%	0.04%	45.21%	88.43%
0.8	100.00%	96.74%	97.55%	0.00%	29.26%	88.53%
0.9	100.00%	100.00%	100.00%	0.00%	11.70%	86.76%

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.48 The experiments of RBF-based approach with QE-5 for multi-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	78.95%	82.63%	78.20%	0.99%	96.01%	84.96%
0.2	82.25%	82.54%	80.84%	0.41%	90.96%	85.99%
0.3	82.35%	82.49%	80.89%	0.41%	90.69%	85.99%
0.4	82.35%	82.49%	80.89%	0.41%	90.69%	85.99%
0.5	82.59%	82.73%	81.12%	0.41%	90.43%	86.12%
0.6	84.71%	83.68%	83.02%	0.07%	85.90%	86.76%
0.7	87.81%	86.23%	85.93%	0.06%	75.27%	87.23%
0.8	91.56%	89.99%	90.17%	0.04%	64.63%	88.34%
0.9	96.30%	92.22%	92.99%	0.01%	50.27%	88.84%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.49 The experiments of RBF-based approach with QE-6 for multi-label service querying

Spread	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
0.1	82.01%	84.81%	81.30%	0.72%	93.62%	86.66%
0.2	83.36%	84.62%	82.25%	0.58%	90.69%	86.89%
0.3	83.36%	84.62%	82.25%	0.58%	90.69%	86.89%
0.4	83.36%	84.62%	82.25%	0.58%	90.69%	86.89%
0.5	84.73%	85.11%	83.45%	0.54%	86.97%	87.14%
0.6	89.29%	88.48%	87.51%	0.08%	78.72%	88.91%
0.7	96.43%	94.68%	94.62%	0.04%	59.04%	90.82%
0.8	98.66%	97.26%	97.34%	0.03%	37.77%	89.32%
0.9	98.46%	97.69%	97.95%	0.01%	17.29%	86.32%

A.5.3 Combine-Label Service Querying

Regarding combine-label RBF-based approach, this thesis focuses on combining the best results of the single-label querying and the multi-label querying. All experimental results for combine-label RBF-based querying approach are presented in Section 6.7.3 .

A.6 KNN-based Querying Approach (Classification-based Approach)

A.6.1 Single-Label based Service Querying

1) No Query Expansion (QE-0)

Table A.50 The experiments of KNN-based approach without query expansion for single-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	42.06%	42.06%	42.06%	0.22%	100.00%	59.41%
2	31.20%	31.20%	31.20%	0.26%	100.00%	51.80%
3	28.69%	28.69%	28.69%	0.27%	100.00%	50.04%
4	31.48%	31.48%	31.48%	0.26%	100.00%	52.00%
5	32.03%	32.03%	32.03%	0.26%	100.00%	52.38%
6	30.92%	30.92%	30.92%	0.27%	100.00%	51.60%
7	32.87%	32.87%	32.87%	0.26%	100.00%	52.97%
8	32.87%	32.87%	32.87%	0.26%	100.00%	52.97%
9	33.98%	33.98%	33.98%	0.25%	100.00%	53.75%
10	33.70%	33.70%	33.70%	0.25%	100.00%	53.55%

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.51 The experiments of KNN-based approach with QE-1 for single-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	64.07%	64.07%	64.07%	0.14%	100.00%	74.83%
2	57.66%	57.66%	57.66%	0.16%	100.00%	70.34%
3	50.97%	50.97%	50.97%	0.19%	100.00%	65.65%
4	46.52%	46.52%	46.52%	0.21%	100.00%	62.53%
5	45.13%	45.13%	45.13%	0.21%	100.00%	61.56%
6	40.39%	40.39%	40.39%	0.23%	100.00%	58.24%
7	42.34%	42.34%	42.34%	0.22%	100.00%	59.61%
8	40.11%	40.11%	40.11%	0.23%	100.00%	58.04%
9	41.50%	41.50%	41.50%	0.22%	100.00%	59.02%
10	40.95%	40.95%	40.95%	0.23%	100.00%	58.63%

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.52 The experiments of KNN-based approach with QE-2 for single-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	56.55%	56.55%	56.55%	0.17%	100.00%	69.56%
2	47.63%	47.63%	47.63%	0.20%	100.00%	63.31%
3	39.83%	39.83%	39.83%	0.23%	100.00%	57.85%
4	38.72%	38.72%	38.72%	0.24%	100.00%	57.07%
5	35.93%	35.93%	35.93%	0.25%	100.00%	55.11%
6	33.43%	33.43%	33.43%	0.26%	100.00%	53.36%
7	33.70%	33.70%	33.70%	0.25%	100.00%	53.55%
8	32.87%	32.87%	32.87%	0.26%	100.00%	52.97%
9	34.54%	34.54%	34.54%	0.25%	100.00%	54.14%
10	34.82%	34.82%	34.82%	0.25%	100.00%	54.34%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.53 The experiments of KNN-based approach with QE-3 for single-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	65.18%	65.18%	65.18%	0.13%	100.00%	75.61%
2	58.50%	58.50%	58.50%	0.16%	100.00%	70.93%
3	55.71%	55.71%	55.71%	0.17%	100.00%	68.97%
4	54.32%	54.32%	54.32%	0.18%	100.00%	68.00%
5	52.37%	52.37%	52.37%	0.18%	100.00%	66.63%
6	49.86%	49.86%	49.86%	0.19%	100.00%	64.87%
7	49.03%	49.03%	49.03%	0.20%	100.00%	64.29%
8	48.75%	48.75%	48.75%	0.20%	100.00%	64.10%
9	49.03%	49.03%	49.03%	0.20%	100.00%	64.29%
10	49.30%	49.30%	49.30%	0.19%	100.00%	64.48%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.54 The experiments of KNN-based approach with QE-4 for single-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	62.95%	62.95%	62.95%	0.14%	100.00%	74.04%
2	59.33%	59.33%	59.33%	0.16%	100.00%	71.51%
3	59.05%	59.05%	59.05%	0.16%	100.00%	71.31%
4	59.05%	59.05%	59.05%	0.16%	100.00%	71.31%

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
5	58.77%	58.77%	58.77%	0.16%	100.00%	71.12%
6	57.38%	57.38%	57.38%	0.16%	100.00%	70.14%
7	56.55%	56.55%	56.55%	0.17%	100.00%	69.56%
8	55.71%	55.71%	55.71%	0.17%	100.00%	68.97%
9	54.32%	54.32%	54.32%	0.18%	100.00%	68.00%
10	53.20%	53.20%	53.20%	0.18%	100.00%	67.21%

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.55 The experiments of KNN-based approach with QE-5 for single-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	72.98%	72.98%	72.98%	0.10%	100.00%	81.07%
2	66.85%	66.85%	66.85%	0.13%	100.00%	76.78%
3	64.90%	64.90%	64.90%	0.13%	100.00%	75.41%
4	62.40%	62.40%	62.40%	0.14%	100.00%	73.66%
5	61.28%	61.28%	61.28%	0.15%	100.00%	72.87%
6	59.61%	59.61%	59.61%	0.16%	100.00%	71.70%
7	57.10%	57.10%	57.10%	0.16%	100.00%	69.95%
8	57.10%	57.10%	57.10%	0.16%	100.00%	69.95%
9	57.38%	57.38%	57.38%	0.16%	100.00%	70.14%
10	57.94%	57.94%	57.94%	0.16%	100.00%	70.53%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.56 The experiments of KNN-based approach with QE-6 for single-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	73.82%	73.82%	73.82%	0.10%	100.00%	81.66%
2	68.52%	68.52%	68.52%	0.12%	100.00%	77.95%
3	67.69%	67.69%	67.69%	0.12%	100.00%	77.37%
4	66.85%	66.85%	66.85%	0.13%	100.00%	76.78%
5	66.57%	66.57%	66.57%	0.13%	100.00%	76.58%
6	65.18%	65.18%	65.18%	0.13%	100.00%	75.61%
7	64.62%	64.62%	64.62%	0.14%	100.00%	75.21%
8	64.62%	64.62%	64.62%	0.14%	100.00%	75.21%
9	61.56%	61.56%	61.56%	0.15%	100.00%	73.07%
10	62.95%	62.95%	62.95%	0.14%	100.00%	74.04%

A.6.2 Multi-Label Service Querying

1) No Query Expansion (QE-0)

Table A.57 The experiments of KNN-based approach without query expansion for multi-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	45.45%	47.27%	43.68%	0.78%	100.00%	61.71%
2	32.46%	49.56%	34.62%	2.44%	99.73%	55.20%
3	48.12%	51.65%	48.06%	0.29%	60.37%	58.22%
4	46.65%	54.43%	47.41%	0.52%	65.16%	58.63%
5	66.29%	67.87%	64.29%	0.20%	29.52%	65.74%
6	62.32%	63.59%	60.66%	0.19%	39.10%	64.40%
7	74.42%	72.80%	72.47%	0.11%	22.87%	69.97%
8	61.71%	60.08%	60.15%	0.16%	29.52%	62.12%
9	71.43%	67.78%	68.30%	0.12%	16.76%	66.48%
10	70.67%	67.21%	68.00%	0.13%	19.95%	66.52%

2) WordNet-based Query Expansion with All-Senses Expansion Technique (QE-1)

Table A.58 The experiments of KNN-based approach with QE-1 for multi-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	68.05%	70.74%	67.43%	0.96%	100.00%	77.80%
2	55.30%	72.04%	58.06%	2.89%	99.73%	71.16%
3	67.12%	73.79%	68.21%	0.42%	73.67%	74.14%
4	55.51%	70.43%	58.38%	1.03%	79.26%	68.26%
5	62.95%	66.59%	62.70%	0.18%	56.12%	67.96%
6	53.15%	61.41%	54.11%	0.61%	61.70%	62.75%
7	68.46%	71.95%	68.31%	0.15%	42.02%	69.70%
8	65.58%	69.11%	65.42%	0.19%	44.95%	68.13%
9	74.03%	76.16%	73.78%	0.12%	31.91%	71.87%
10	72.35%	74.73%	72.25%	0.13%	34.31%	71.11%

3) WordNet-based Query Expansion with Proper-Sense Expansion Technique (QE-2)

Table A.59 The experiments of KNN-based approach with QE-2 for multi-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	58.56%	61.45%	57.79%	0.46%	100.00%	71.24%
2	46.44%	62.04%	49.15%	1.83%	98.94%	64.82%

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
3	62.44%	64.82%	61.46%	0.40%	67.82%	69.03%
4	50.04%	61.82%	51.99%	0.79%	74.73%	63.18%
5	58.34%	59.95%	55.88%	0.97%	49.73%	63.02%
6	55.10%	59.73%	53.94%	0.84%	53.72%	62.02%
7	66.36%	68.94%	65.35%	0.20%	36.97%	67.20%
8	60.06%	63.66%	59.02%	0.26%	39.63%	63.33%
9	74.53%	77.48%	73.34%	0.22%	24.73%	71.11%
10	74.40%	78.27%	73.85%	0.20%	27.93%	71.74%

4) Ontology-based Query Expansion with All-Related Terms Expansion Technique (QE-3)

Table A.60 The experiments of KNN-based approach with QE-3 for multi-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	65.38%	63.73%	62.89%	0.24%	100.00%	75.11%
2	59.11%	67.24%	59.81%	0.80%	99.20%	72.46%
3	72.83%	72.12%	70.54%	0.16%	74.47%	76.68%
4	64.71%	71.05%	65.38%	0.23%	78.19%	73.04%
5	74.02%	72.43%	71.42%	0.14%	63.56%	75.70%
6	70.33%	72.24%	68.68%	0.18%	66.22%	74.18%
7	75.08%	76.04%	73.79%	0.14%	55.85%	75.86%
8	70.62%	73.65%	70.04%	0.18%	57.71%	73.43%
9	80.54%	81.54%	79.26%	0.12%	45.21%	78.10%
10	74.69%	75.61%	73.51%	0.14%	48.40%	74.48%

5) Ontology-based Query Expansion with Most-Related Term Extraction Technique (QE-4)

Table A.61 The experiments of KNN-based approach with QE-4 for multi-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	65.64%	64.35%	63.44%	0.43%	100.00%	75.36%
2	60.24%	69.76%	61.28%	1.22%	99.47%	73.49%
3	77.11%	76.67%	75.25%	0.14%	74.47%	79.78%
4	70.14%	74.75%	69.90%	0.73%	78.72%	76.45%
5	78.43%	77.24%	76.26%	0.14%	66.49%	79.35%
6	73.39%	74.28%	71.71%	0.18%	70.74%	76.84%
7	77.33%	76.73%	75.85%	0.12%	59.84%	77.78%
8	76.56%	76.49%	75.07%	0.14%	61.17%	77.51%
9	85.31%	84.26%	83.38%	0.09%	48.14%	81.48%
10	83.04%	82.41%	81.16%	0.10%	50.53%	80.32%

6) Hybrid-based Approach with All-Senses and All-Related Terms Expansion Technique (QE-5)

Table A.62 The experiments of KNN-based approach with QE-5 for multi-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	75.42%	75.61%	73.79%	1.14%	100.00%	82.41%
2	64.05%	78.19%	65.76%	2.62%	99.47%	76.74%
3	77.85%	80.50%	77.14%	0.24%	80.32%	81.80%
4	70.34%	77.75%	70.47%	0.71%	82.98%	77.71%
5	83.58%	83.09%	81.27%	0.12%	67.29%	83.16%
6	74.55%	78.26%	73.71%	0.40%	72.87%	78.49%
7	80.30%	81.10%	78.85%	0.13%	59.31%	79.99%
8	78.72%	81.91%	78.15%	0.14%	60.37%	79.53%
9	86.55%	88.31%	85.55%	0.09%	50.27%	83.23%
10	83.72%	85.84%	82.82%	0.11%	50.53%	81.35%

7) Hybrid-based Approach with Proper-Sense and Most-Related Term Expansion Technique (QE-6)

Table A.63 The experiments of KNN-based approach with QE-6 for multi-label service querying

#K	Avg Precision	Avg Recall	Avg F-Measure	Avg Fallout	Querying Rate	All-measures
1	74.69%	75.52%	73.68%	0.55%	100.00%	82.17%
2	70.04%	78.96%	71.14%	1.66%	99.20%	80.16%
3	82.05%	82.33%	80.35%	0.27%	81.91%	84.47%
4	77.84%	81.40%	77.25%	0.43%	86.17%	82.79%
5	85.39%	84.24%	82.80%	0.27%	74.47%	85.34%
6	79.97%	81.09%	78.39%	0.41%	77.66%	82.50%
7	85.08%	83.70%	82.94%	0.26%	69.95%	84.48%
8	83.18%	83.66%	81.74%	0.40%	69.15%	83.39%
9	88.45%	87.45%	86.28%	0.08%	61.17%	85.60%
10	86.77%	86.16%	84.55%	0.09%	60.64%	84.40%

A.6.3 Combine-Label Service Querying

Regarding combine-label KNN-based approach, this thesis focuses on combining the best results of the single-label querying and the multi-label querying. All experimental results for combine-label KNN-based querying approach are presented in Section 6.7.3 .

A.7 CT-based Querying Approach (Classification-based Approach)

All experimental results of CT-based approach for single-label, multi-label, and combine-label querying are presented in Section 6.7.3 .

A.8 SVM-based Querying Approach

All experimental results of SVM-based approach for multi-label querying are presented in Section 6.7.3 .

Appendix B Additional Experimental Results of the Fuzzy-based Service Retrieval

B.1 Fuzzy Variables (Trapezoidal-shaped membership function)

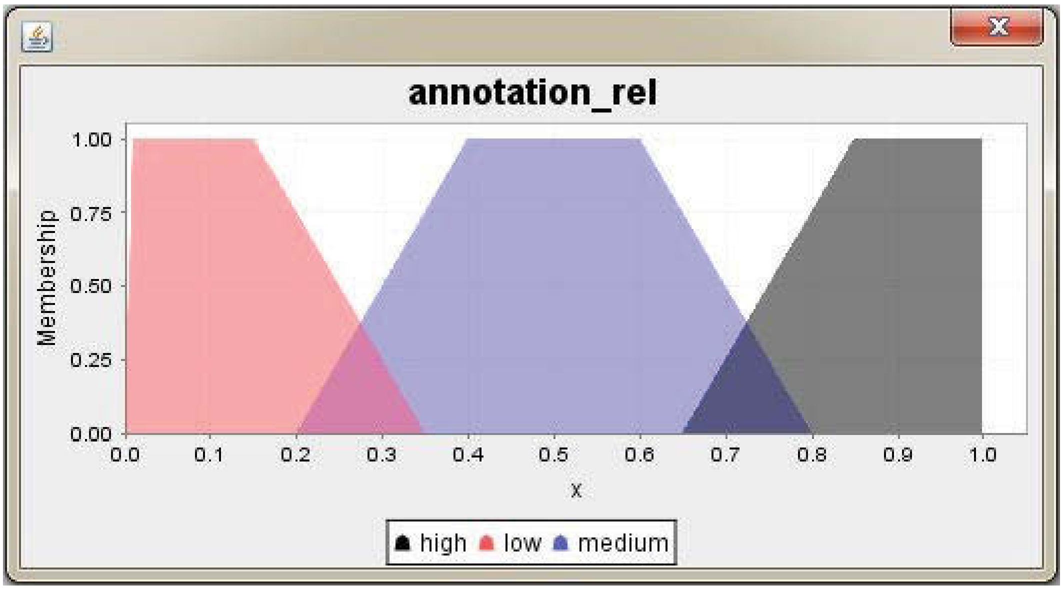


Figure B.1 The membership functions of the *annotation_rel* variable

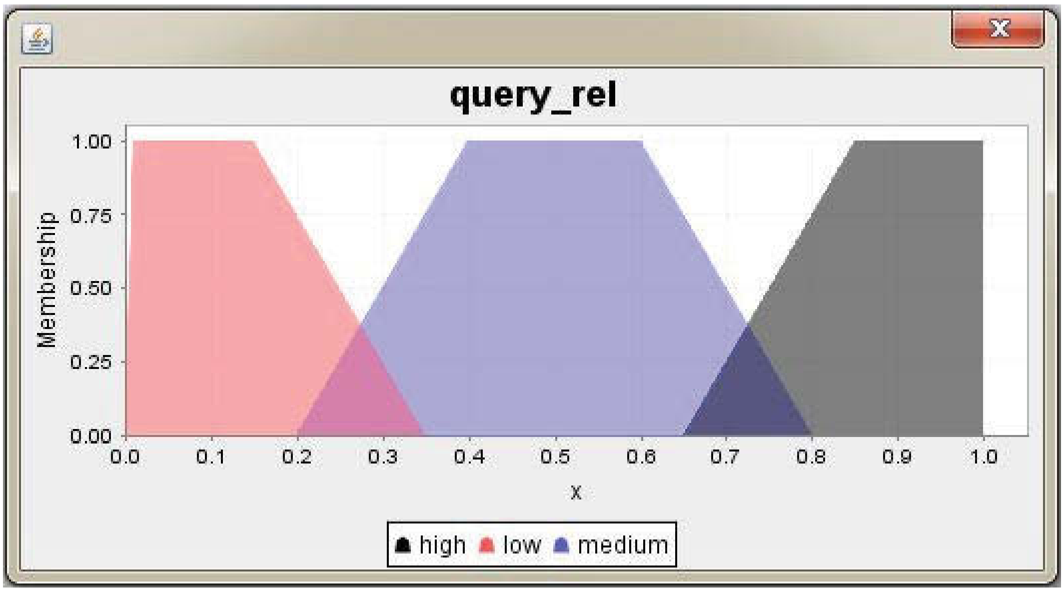


Figure B.2 The membership functions of the *query_rel* variable

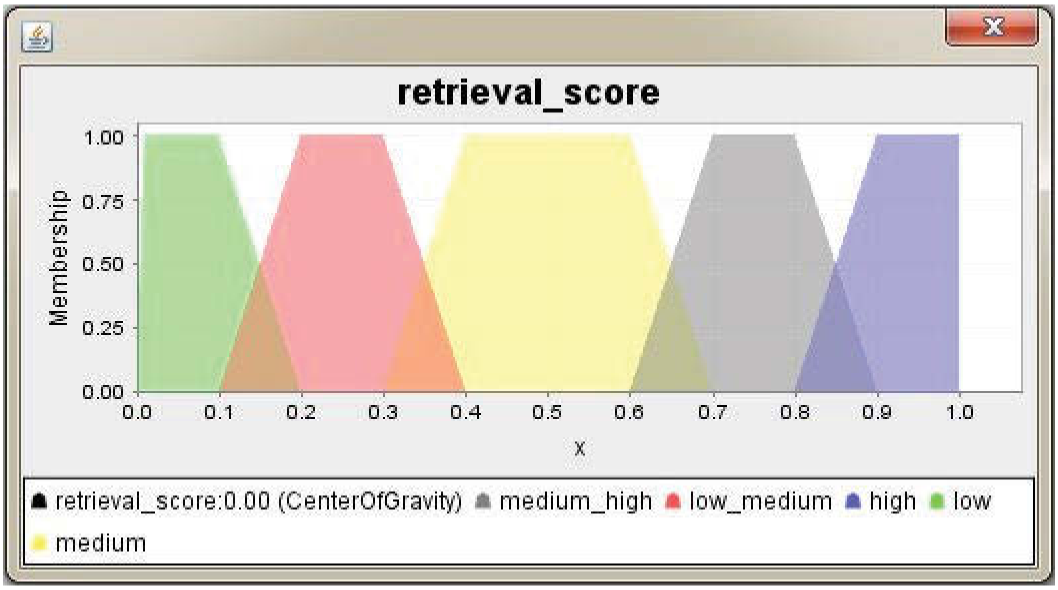


Figure 9.3 The membership functions of the *retrieval_score* variable

B.2 Fuzzy Rules

RULE 1 : IF *annotation_rel* IS low AND *query_rel* IS low
THEN *retrieval_score* IS low_medium

RULE 2 : IF *annotation_rel* IS low AND *query_rel* IS medium
THEN *retrieval_score* IS medium

RULE 3 : IF *annotation_rel* IS low AND *query_rel* IS high
THEN *retrieval_score* is medium_high

RULE 4 : IF *annotation_rel* IS medium AND *query_rel* IS low
THEN *retrieval_score* IS medium

RULE 5 : IF *annotation_rel* IS medium AND *query_rel* IS medium
THEN *retrieval_score* IS medium_high

RULE 6 : IF *annotation_rel* IS medium AND *query_rel* IS high
THEN *retrieval_score* is high

RULE 7 : IF *annotation_rel* IS high AND *query_rel* IS low
THEN *retrieval_score* IS medium_high

RULE 8 : IF *annotation_rel* IS high AND *query_rel* IS medium
THEN *retrieval_score* IS high

RULE 9 : IF *annotation_rel* IS high AND *query_rel* IS high
THEN *retrieval_score* is high

B.3 Experimental Results

B.3.1 ECBR Retrieval Approach

1) Single-Label Service Retrieval (*RT*=0.6)

Table B.1 The experiments of ECBR approach for a single-label fuzzy service retrieval (*RT*=0.6)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	11.26%	15.79%	6.35%	0.35%	11.43%	24.49%
QE-1	10.36%	25.32%	6.75%	0.45%	13.33%	25.89%
QE-2	10.97%	19.58%	6.89%	0.38%	12.38%	25.16%
QE-3	20.78%	33.50%	21.38%	0.19%	51.43%	39.23%
QE-4	20.88%	36.70%	20.73%	0.19%	39.05%	37.80%
QE-5	26.37%	43.77%	26.58%	0.22%	42.86%	42.50%
QE-6	22.11%	39.24%	22.28%	0.20%	38.10%	38.76%

2) Multi-Label Service Retrieval (*RT*=0.8)

Table B.2 The experiments of ECBR approach for a multi-label fuzzy service retrieval (*RT*=0.8)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	50.00%	12.63%	16.23%	0.03%	2.50%	39.70%
QE-1	33.33%	22.22%	26.67%	0.05%	2.50%	36.03%
QE-2	25.00%	16.67%	20.00%	0.05%	1.67%	30.74%
QE-3	50.00%	12.63%	16.23%	0.03%	2.50%	39.70%
QE-4	50.00%	12.63%	16.23%	0.03%	2.50%	39.70%
QE-5	25.00%	16.67%	20.00%	0.05%	1.67%	30.74%
QE-6	25.00%	16.67%	20.00%	0.05%	1.67%	30.74%

3) Combine-Label Service Retrieval (*RT*=0.8)

Table B.3 The experiments of ECBR approach for a combine-label fuzzy service retrieval (*RT*=0.8)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	50.00%	12.63%	16.23%	0.03%	2.50%	39.70%
QE-1	33.33%	22.22%	26.67%	0.05%	2.50%	36.03%
QE-2	25.00%	16.67%	20.00%	0.05%	1.67%	30.74%
QE-3	50.00%	12.63%	16.23%	0.03%	2.50%	39.70%
QE-4	50.00%	12.63%	16.23%	0.03%	2.50%	39.70%
QE-5	25.00%	16.67%	20.00%	0.05%	1.67%	30.74%
QE-6	25.00%	16.67%	20.00%	0.05%	1.67%	30.74%

B.3.2 VSM-based Retrieval Approach (Vector-based Approach)

1) Single-Label Service Retrieval (*RT*=0.9)

Table B.4 The experiments of VSM-based approach for a single-label fuzzy service retrieval (*RT*=0.9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	16.41%	15.81%	9.68%	0.20%	11.43%	27.07%
QE-1	0.00%	0.00%	0.00%	0.07%	1.90%	15.27%
QE-2	20.00%	2.11%	3.81%	0.22%	4.76%	24.57%
QE-3	13.36%	17.34%	9.12%	0.24%	10.48%	25.85%
QE-4	24.43%	21.51%	16.50%	0.15%	19.05%	33.31%
QE-5	0.00%	0.00%	0.00%	0.07%	0.95%	15.13%
QE-6	23.33%	14.50%	14.71%	0.20%	8.57%	29.97%

2) Multi-Label Service Retrieval (*RT*=0.9)

Table B.5 The experiments of VSM-based approach for a multi-label fuzzy service retrieval (*RT*=0.9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	17.60%	22.25%	12.45%	0.27%	27.50%	31.33%
QE-1	11.67%	26.67%	15.71%	0.09%	4.17%	26.64%
QE-2	18.18%	25.44%	14.45%	0.27%	9.17%	29.59%
QE-3	21.37%	33.36%	16.66%	0.30%	20.00%	34.01%
QE-4	19.64%	29.47%	19.64%	0.24%	33.33%	35.19%
QE-5	41.67%	33.89%	19.47%	0.06%	5.83%	40.54%
QE-6	32.17%	37.53%	22.68%	0.25%	16.67%	39.36%

3) Combine-Label Service Retrieval (*RT*=0.9)

Table B.6 The experiments of VSM-based approach for a combine-label fuzzy service retrieval (*RT*=0.9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	17.60%	22.25%	12.45%	0.27%	27.50%	31.33%
QE-1	11.67%	26.67%	15.71%	0.09%	4.17%	26.64%
QE-2	18.18%	25.44%	14.45%	0.27%	9.17%	29.59%
QE-3	21.37%	33.36%	16.66%	0.30%	20.00%	34.01%
QE-4	19.64%	29.47%	19.64%	0.24%	33.33%	35.19%
QE-5	41.67%	33.89%	19.47%	0.06%	5.83%	40.54%
QE-6	32.17%	37.53%	22.68%	0.25%	16.67%	39.36%

B.3.3 EVSM-based Retrieval Approach (Vector-based Approach)

1) Single-Label Service Retrieval (*RT*=0.9)

Table B.7 The experiments of EVSM-based approach for a single-label fuzzy service retrieval (*RT*=0.9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	0.00%	0.00%	0.00%	0.07%	1.90%	15.27%
QE-1	31.31%	34.18%	29.22%	0.19%	20.95%	40.15%
QE-2	28.51%	42.98%	30.33%	0.22%	13.33%	39.37%
QE-3	25.00%	3.95%	6.82%	0.10%	3.81%	27.17%
QE-4	31.25%	14.80%	16.53%	0.09%	7.62%	33.33%
QE-5	2.50%	5.00%	3.33%	0.37%	3.81%	17.77%
QE-6	30.00%	20.53%	21.00%	0.14%	4.76%	33.92%

2) Multi-Label Service Retrieval (*RT*=0.9)

Table B.8 The experiments of EVSM-based approach for a multi-label fuzzy service retrieval (*RT*=0.9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	24.48%	30.37%	22.44%	0.22%	7.50%	33.81%
QE-1	25.89%	35.90%	24.66%	0.30%	34.17%	39.52%
QE-2	30.79%	47.99%	30.28%	0.35%	20.83%	42.13%
QE-3	44.89%	50.69%	32.24%	0.13%	10.00%	46.88%
QE-4	39.24%	49.02%	37.56%	0.17%	14.17%	45.78%
QE-5	36.43%	26.46%	16.13%	0.31%	7.50%	37.04%
QE-6	50.87%	35.96%	30.79%	0.20%	9.17%	46.71%

3) Combine-Label Service Retrieval (*RT*=0.9)

Table B.9 The experiments of EVSM-based approach for a combine-label fuzzy service retrieval (*RT*=0.9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	24.48%	30.37%	22.44%	0.22%	7.50%	33.81%
QE-1	25.89%	35.90%	24.66%	0.30%	34.17%	39.52%
QE-2	30.79%	47.99%	30.28%	0.35%	20.83%	42.13%
QE-3	44.89%	50.69%	32.24%	0.13%	10.00%	46.88%
QE-4	39.24%	49.02%	37.56%	0.17%	14.17%	45.78%
QE-5	36.43%	26.46%	16.13%	0.31%	7.50%	37.04%
QE-6	50.87%	35.96%	30.79%	0.20%	9.17%	46.71%

B.3.4 FF-based Retrieval Approach (Classification-based Approach)

1) Single-Label Service Retrieval

Table B.10 The experiments of FF-based approach for a single-label fuzzy service retrieval (*AN*=90, *QN*=30)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	0.00%	0.00%	0.00%	13.33%	65.71%	22.86%
QE-1	1.78%	100.00%	3.38%	14.76%	74.29%	40.15%
QE-2	1.75%	100.00%	3.31%	14.77%	64.76%	38.70%
QE-3	1.63%	100.00%	3.07%	14.78%	49.52%	36.32%
QE-4	1.71%	100.00%	3.21%	14.51%	47.62%	36.13%
QE-5	1.75%	100.00%	3.31%	14.77%	72.38%	39.84%
QE-6	1.74%	100.00%	3.30%	14.77%	67.62%	39.12%

2) Multi-Label Service Retrieval

Table B.11 The experiments of FF-based approach for a multi-label fuzzy service retrieval (*AN*=10, *QN*=30)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	87.81%	70.25%	77.71%	0.33%	3.33%	72.77%
QE-1	94.49%	57.95%	69.29%	0.07%	3.33%	72.37%
QE-2	89.16%	79.51%	83.98%	0.33%	3.33%	75.64%
QE-3	94.49%	57.95%	69.29%	0.07%	3.33%	72.37%
QE-4	90.53%	76.24%	82.43%	0.27%	4.17%	75.60%
QE-5	90.53%	76.24%	82.43%	0.27%	4.17%	75.60%
QE-6	73.78%	64.84%	68.66%	0.44%	5.00%	65.22%

Table B.12 The experiments of FF-based approach for a multi-label fuzzy service retrieval (*AN*=60, *QN*=30)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	78.35%	46.54%	54.77%	0.14%	7.50%	62.64%
QE-1	82.81%	52.56%	61.31%	0.10%	7.50%	66.31%
QE-2	78.42%	66.18%	70.01%	0.23%	9.17%	68.14%
QE-3	72.86%	60.26%	64.32%	0.14%	10.00%	64.31%
QE-4	72.62%	53.91%	58.25%	0.15%	10.83%	62.47%
QE-5	85.93%	78.23%	80.80%	0.24%	10.83%	74.82%
QE-6	70.56%	63.08%	64.76%	0.44%	10.00%	63.83%

3) Combine-Label Service Retrieval

Table B.13 The experiments of FF-based approach for a combine-label fuzzy service retrieval (*AN*=90, *QN*=30)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	0.00%	0.00%	0.00%	15.04%	100.00%	27.74%
QE-1	0.00%	0.00%	0.00%	5.29%	100.00%	29.21%
QE-2	0.00%	0.00%	0.00%	4.83%	100.00%	29.28%
QE-3	0.00%	0.00%	0.00%	4.71%	100.00%	29.29%
QE-4	72.80%	41.44%	47.61%	0.11%	13.33%	59.46%
QE-5	68.90%	39.88%	48.18%	0.16%	15.00%	58.00%
QE-6	64.37%	41.01%	45.93%	0.26%	16.67%	56.25%

B.3.5 RBF-based Retrieval Approach (Classification-based Approach)

1) Single-Label Service Retrieval

Table B.14 The experiments of RBF-based approach for a single-label fuzzy service retrieval (*AS*=0.6, *QS*=0.4)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	0.00%	0.00%	0.00%	0.16%	8.57%	16.26%
QE-1	55.56%	17.22%	21.71%	0.06%	8.57%	44.34%
QE-2	42.86%	18.41%	21.43%	0.07%	6.67%	39.11%
QE-3	38.89%	10.21%	13.35%	0.14%	8.57%	35.35%
QE-4	50.00%	25.92%	28.75%	0.14%	6.67%	44.18%
QE-5	50.00%	21.91%	25.21%	0.17%	11.43%	43.76%
QE-6	50.00%	24.45%	27.14%	0.19%	9.52%	44.14%

2) Multi-Label Service Retrieval

Table B.15 The experiments of RBF-based approach for a multi-label fuzzy service retrieval (*AS*=0.8, *QS*=0.9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	-	-	-	-	0.00%	-
QE-1	-	-	-	-	0.00%	-
QE-2	-	-	-	-	0.00%	-
QE-3	96.11%	51.88%	66.30%	0.06%	4.17%	71.79%

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-4	-	-	-	-	0.00%	-
QE-5	97.84%	36.83%	49.95%	0.03%	7.50%	68.27%
QE-6	100.00%	34.62%	51.43%	0.00%	0.83%	68.03%

3) Combine-Label Service Retrieval

Table B.16 The experiments of RBF-based approach for a combine-label fuzzy service retrieval (*AS*=0.6, *QS*=0.9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	82.45%	71.75%	74.28%	0.23%	3.33%	70.35%
QE-1	62.51%	41.20%	47.07%	0.20%	4.17%	53.84%
QE-2	78.62%	60.40%	66.39%	0.13%	6.67%	66.45%
QE-3	65.57%	45.24%	50.15%	0.22%	12.50%	57.38%
QE-4	49.92%	40.10%	42.45%	0.23%	11.67%	49.07%
QE-5	64.20%	41.47%	46.48%	0.18%	18.33%	56.60%
QE-6	61.29%	42.32%	47.36%	0.18%	10.83%	54.57%

B.3.6 KNN-based Retrieval Approach (Classification-based Approach)

1) Single-Label Service Retrieval

Table B.17 The experiments of KNN-based approach for a single-label fuzzy service retrieval (*AK*=3, *QK*=1)

2) Multi-Label Service Retrieval

Table B.18 The experiments of KNN-based approach for a multi-label fuzzy service retrieval (*AK*=1, *QK*=9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	36.60%	28.09%	27.36%	0.28%	5.83%	38.79%
QE-1	38.47%	30.97%	30.19%	0.23%	15.00%	41.78%
QE-2	47.23%	38.20%	38.74%	0.24%	7.50%	46.52%
QE-3	33.23%	33.44%	29.37%	0.16%	28.33%	41.94%
QE-4	30.50%	37.77%	31.17%	0.19%	21.67%	40.76%
QE-5	32.04%	40.33%	31.99%	0.27%	29.17%	43.00%
QE-6	26.00%	34.45%	27.24%	0.22%	25.83%	38.50%

3) Combine-Label Service Retrieval

Table B.19 The experiments of KNN-based approach for a combine-label fuzzy service retrieval (AK=1, QK=9)

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	29.65%	27.49%	22.19%	0.27%	33.33%	39.27%
QE-1	37.64%	28.10%	27.79%	0.23%	25.83%	42.28%
QE-2	37.60%	37.81%	29.81%	0.36%	25.83%	44.00%
QE-3	29.68%	25.90%	23.50%	0.18%	40.83%	40.38%
QE-4	25.31%	27.34%	23.74%	0.26%	37.50%	38.37%
QE-5	27.49%	38.94%	27.33%	0.45%	43.33%	42.37%
QE-6	25.35%	30.79%	23.44%	0.29%	41.67%	39.48%

B.3.7 CT-based Retrieval Approach (Classification-based Approach)

1) Single-Label Service Retrieval

Table B.20 The experiments of CT-based approach for a single-label fuzzy service retrieval

2) Multi-Label Service Retrieval

Table B.21 The experiments of CT-based approach for a multi-label fuzzy service retrieval

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	61.01%	68.87%	64.32%	0.42%	10.83%	60.94%
QE-1	62.38%	66.90%	62.80%	0.38%	16.67%	61.85%
QE-2	60.14%	64.77%	60.42%	0.41%	15.00%	60.02%
QE-3	62.38%	71.80%	66.22%	0.39%	14.17%	62.72%
QE-4	62.99%	72.52%	66.91%	0.39%	15.00%	63.30%
QE-5	55.09%	62.96%	58.37%	0.37%	20.00%	58.18%
QE-6	59.04%	66.94%	62.30%	0.39%	16.67%	60.44%

3) Combine-Label Service Retrieval

Table B.22 The experiments of CT-based approach for a combine-label fuzzy service retrieval

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	50.54%	57.52%	53.32%	0.39%	15.00%	54.03%
QE-1	59.32%	64.09%	60.10%	0.45%	20.00%	60.29%
QE-2	54.73%	59.38%	55.39%	0.46%	19.17%	56.91%

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-3	63.09%	72.50%	66.99%	0.38%	15.83%	63.48%
QE-4	57.52%	65.41%	60.46%	0.37%	16.67%	59.33%
QE-5	50.93%	61.97%	54.04%	0.42%	21.67%	55.96%
QE-6	55.09%	62.96%	58.37%	0.39%	20.00%	58.18%

B.3.8 SVM-based Retrieval Approach (Classification-based Approach)

1) Multi-Label Service Retrieval

Table B.23 The experiments of SVM-based approach for a multi-label fuzzy service retrieval

QE-Method	Avg Precision	Avg Recall	F-measure	Avg Fallout	Retrieval Rate	All-measures
QE-0	2.16%	90.86%	3.90%	12.87%	99.17%	43.02%
QE-1	13.23%	45.12%	15.47%	3.09%	42.50%	35.29%
QE-2	12.39%	48.06%	14.57%	3.56%	40.00%	34.82%
QE-3	8.57%	62.68%	11.03%	7.21%	45.83%	35.28%
QE-4	10.16%	60.67%	12.78%	6.30%	42.50%	35.51%
QE-5	13.43%	47.94%	15.58%	3.70%	38.33%	35.09%
QE-6	14.08%	55.22%	16.73%	4.32%	36.67%	36.28%

Bibliography

[1] Akkiraju, R., Ivan, A., Goodwin, R., Srivastava, B. & Syeda-Mahmood, T. 2006, 'Semantic Matching to Achieve Web Service Discovery and Composition', *The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06)*, pp. 70-.

[2] Almpanidis, G. & Kotropoulos, C. 2005, 'Combining Text and Link Analysis for Focused Crawling', in S. Singh, M. Singh, C. Apte & P. Perner (eds), *Pattern Recognition and Data Mining*, vol. 3686, Springer Berlin Heidelberg, pp. 278-87.

[3] Almpanidis, G., Kotropoulos, C. & Pitas, I. 2005, 'Focused Crawling Using Latent Semantic Indexing – An Application for Vertical Search Engines', in A. Rauber, S. Christodoulakis & A. Tjoa (eds), *Research and Advanced Technology for Digital Libraries*, vol. 3652, Springer Berlin Heidelberg, pp. 402-13.

[4] Awad, A., Polyvyanyy, A. & Weske, M. 2008, 'Semantic Querying of Business Process Models', *Enterprise Distributed Object Computing Conference, 2008. EDOC '08. 12th International IEEE*, pp. 85-94.

[5] Baeza-Yates, R. & Ribeiro-Neto, B. 1999, *Modern information retrieval*, vol. 463, ACM press New York.

[6] Bernard, K., Cassidy, A., Clark, M., Liu, K., Lobaton, K., McNeill, D. & Brown, D. 2011, 'Identifying and tracking online financial services through web mining and latent semantic indexing', *Systems and Information Engineering Design Symposium (SIEDS), 2011 IEEE*, pp. 158-63.

[7] Bhagwat, D. & Polyzotis, N. 2005, 'Searching a file system using inferred semantic links', paper presented to the *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, Salzburg, Austria.

[8] Bhogal, J., Macfarlane, A. & Smith, P. 2007, 'A review of ontology based query expansion', *Information Processing & Management*, vol. 43, no. 4, pp. 866-86.

[9] Bo, J. & Zhiyuan, L. 2013, 'A New Algorithm for Semantic Web Service Matching', *Journal of Software (1796217X)*, vol. 8, no. 2, pp. 351-6.

[10] Boley, H. & Chang, E. 2007, 'Digital Ecosystems: Principles and Semantics', *2007 Inaugural IEEE-IES Digital EcoSystems and Technologies Conference*, pp. 398-403.

[11] Bruno, M., Canfora, G., Penta, M.D. & Scognamiglio, R. 2005, 'An approach to support Web service classification and annotation', *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, pp. 138-43.

[12] Bryman, A. 2015, *Social research methods*, Oxford university press.

[13] Canturk, D. & Senkul, P. 2011, 'Semantic Annotation of Web Services with Lexicon-Based Alignment', *Services (SERVICES), 2011 IEEE World Congress on*, pp. 355-62.

[14] Chang, E. & West, M. 2006, 'Digital Ecosystems A Next Generation of the Collaborative Environment', *iiWAS*, pp. 3-24.

[15] Cingolani, P. 2016, *jFuzzyLogic*, 'The most complete fuzzy logic library in Java [...]', <http://jfuzzylogic.sourceforge.net/html/index.html>.

[16] Cingolani, P. & Alcala-Fdez, J. 2012, 'jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation', *FUZZ-IEEE*, Citeseer, pp. 1-8.

[17] Cingolani, P. & Alcalá-Fdez, J. 2013, 'jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming', *International Journal of Computational Intelligence Systems*, vol. 6, no. sup1, pp. 61-75.

[18] Crasso, M., Zunino, A. & Campo, M. 2011, 'Combining query-by-example and query expansion for simplifying web service discovery', *Information Systems Frontiers*, vol. 13, no. 3, pp. 407-28.

[19] de Castilho, R.E. & Gurevych, I. 2011, 'Semantic Service Retrieval Based on Natural Language Querying and Semantic Similarity', *ICSC*, pp. 173-6.

[20] Dichev, C. & Dicheva, D. 2006, 'View-Based Semantic Search and Browsing', *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pp. 919-25.

[21] Dong, H. & Hussain, F.K. 2011, 'Focused crawling for automatic service discovery, annotation, and classification in industrial digital ecosystems', *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 6, pp. 2106-16.

[22] Dong, H. & Hussain, F.K. 2013, 'SOF: a semi-supervised ontology-learning-based focused crawler', *Concurrency and Computation: Practice and Experience*, vol. 25, no. 12, pp. 1755-70.

[23] Dong, H. & Hussain, F.K. 2014, 'Self-Adaptive Semantic Focused Crawler for Mining Services Information Discovery', *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 2, pp. 1616-26.

[24] Dong, H., Hussain, F.K. & Chang, E. 2008a, 'A survey in semantic search technologies', *Digital Ecosystems and Technologies, 2008. DEST 2008. 2nd IEEE International Conference on*, pp. 403-8.

[25] Dong, H., Hussain, F.K. & Chang, E. 2008b, 'A Transport Service Ontology-based Focused Crawler', *Semantics, Knowledge and Grid, 2008. SKG '08. Fourth International Conference on*, pp. 49-56.

[26] Dong, H., Hussain, F.K. & Chang, E. 2009, 'A QoS-based service retrieval methodology for digital ecosystems', *International Journal of Web and Grid Services*, vol. 5, no. 3, pp. 261-83.

[27] Dong, H., Hussain, F.K. & Chang, E. 2011, 'A service search engine for the industrial digital ecosystems', *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 6, pp. 2183-96.

[28] Dong, H., Hussain, F.K. & Chang, E. 2012, 'Ontology-learning-based focused crawling for online service advertising information discovery and classification', *Service-Oriented Computing*, Springer, pp. 591-8.

[29] Du, Y., Pen, Q. & Gao, Z. 2013, 'A topic-specific crawling strategy based on semantics similarity', *Data & Knowledge Engineering*, vol. 88, no. 0, pp. 75-93.

[30] Duo, Z., Juan-Zi, L. & Bin, X. 2005, 'Web service annotation using ontology mapping', *Service-Oriented System Engineering, 2005. SOSE 2005. IEEE International Workshop*, pp. 235-42.

[31] Eftychiou, A., Vrusias, B. & Antonopoulos, N. 2012, 'A dynamically semantic platform for efficient information retrieval in P2P networks', *International Journal of Grid and Utility Computing*, vol. 3, no. 4, pp. 271-83.

[32] Fang, W., Cui, Z. & Zhao, P. 2007, 'Ontology-Based Focused Crawling of Deep Web Sources', in Z. Zhang & J. Siekmann (eds), *Knowledge Science, Engineering and Management*, vol. 4798, Springer Berlin Heidelberg, pp. 514-9.

[33] Fellbaum, C. 1998, *WordNet*, Wiley Online Library.

[34] Galliers, R. 1992, *Information systems research: Issues, methods and practical guidelines*, Blackwell Scientific.

[35] Garcés, P.J., Olivas, J.A. & Romero, F.P. 2003, 'Conceptual matching in web search using FIS-CRM for representing documents', *EUSFLAT Conf.*, Citeseer, pp. 63-6.

[36] Hai, D., Hussain, F.K. & Chang, E. 2008, 'A Transport Service Ontology-based Focused Crawler', *Semantics, Knowledge and Grid, 2008. SKG '08. Fourth International Conference on*, pp. 49-56.

[37] Heß, A., Johnston, E. & Kushmerick, N. 2004, 'ASSAM: A Tool for Semi-automatically Annotating Semantic Web Services', in S.A. McIlraith, D. Plexousakis & F. van Harmelen (eds), *The Semantic Web – ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 320-34.

[38] Huang, W., Zhang, L., Zhang, J. & Zhu, M. 2009, 'Semantic Focused Crawling for Retrieving E-Commerce Information', *Journal of Software*, vol. 4, no. 5, pp. 436-43.

[39] Hui, W., Zhiyong, F., Shizhan, C., Jinna, X. & Yang, S. 2010, 'Constructing Service Network via Classification and Annotation', *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*, pp. 69-73.

[40] Iqbal, K., Sbodio, M.L., Peristeras, V. & Giuliani, G. 2008, 'Semantic Service Discovery using SAWSDL and SPARQL', *2008 Fourth International Conference on Semantics, Knowledge and Grid*, pp. 205-12.

[41] Jalilian, O. & Khotanlou, H. 2011, 'A new fuzzy-based method to weigh the related concepts in semantic focused web crawlers', *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, vol. 3, pp. 23-7.

[42] Ji, L., Yan, J., Liu, N., Zhang, W., Fan, W. & Chen, Z. 2009, 'ExSearch: a novel vertical search engine for online barter business', paper presented to the *Proceedings of the 18th ACM conference on Information and knowledge management*, Hong Kong, China.

[43] Jiang, B. & Luo, Z. 2013, 'A new algorithm for semantic web service matching', *Journal of Software*, vol. 8, no. 2, pp. 351-6.

[44] Jin, H., Ning, X. & Chen, H. 2006, 'Efficient search for peer-to-peer information retrieval using semantic small world', paper presented to the *Proceedings of the 15th international conference on World Wide Web*, Edinburgh, Scotland.

[45] Jin, H., Sun, A., Zheng, R., He, R. & Zhang, Q. 2008, 'Ontology-based semantic integration scheme for medical image grid', *International Journal of Grid and Utility Computing*, vol. 1, no. 2, pp. 86-97.

[46] Jin, Y. 2010, *A Definition of Soft Computing*, 'Soft computing differs from conventional (hard) computing [...]', viewed 30 August 2016, <http://www.soft-computing.de/def.html>.

[47] Kearns, K. 2000, 'Semantics. Modern Linguistics Series', *England: Macmillan*.

[48] Klir, G. & Yuan, B. 1995, *Fuzzy sets and fuzzy logic*, vol. 4, Prentice hall New Jersey.

[49] Lai, C. & Moulin, C. 2013, 'Semantic indexing modelling of resources within a distributed system', *International Journal of Grid and Utility Computing*, vol. 4, no. 1, pp. 21-39.

[50] Leacock, C. & Chodorow, M. 1998, 'Combining local context and WordNet similarity for word sense identification', *WordNet: An electronic lexical database*, vol. 49, no. 2, pp. 265-83.

[51] Lerman, K., Plangprasopchok, A. & Knoblock, C.A. 2006, 'Automatically labeling the inputs and outputs of web services', *AAAI*, vol. 6, pp. 1363-8.

[52] Liu, Y. & Shao, Z. 2010, 'A framework for semantic Web Services annotation and discovery based on ontology', *Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on*, vol. 2, IEEE, pp. 1034-9.

[53] Liu, Y., Zhang, D., Lu, G. & Ma, W.-Y. 2007, 'A survey of content-based image retrieval with high-level semantics', *Pattern Recognition*, vol. 40, no. 1, pp. 262-82.

[54] Ma, S.P., Li, C.H., Tsai, Y.Y. & Lan, C.W. 2013, 'Web Service Discovery Using Lexical and Semantic Query Expansion', *e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on*, pp. 423-8.

[55] Madkour, M., Maach, A., Driss, E. & Hasbi, A. 2012, 'Fuzzy-based approach for context-aware service retrieval', *Innovative Computing Technology (INTECH), 2012 Second International Conference on*, pp. 396-401.

[56] Mangold, C. 2007, 'A survey and classification of semantic search approaches', *International Journal of Metadata, Semantics and Ontologies*, vol. 2, no. 1, pp. 23-34.

[57] Mekhzoumi, D., Aliane, H. & Nouali, O. 2011, 'Queries reformulation for the discovery of semantic web services', *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on*, pp. 256-61.

[58] Meyer, H. & Weske, M. 2006, 'Light-Weight Semantic Service Annotations Through Tagging', in A. Dan & W. Lamersdorf (eds), *Service-Oriented Computing – ICSOC 2006*, vol. 4294, Springer Berlin Heidelberg, pp. 465-70.

[59] Miller, G.A. 1995, 'WordNet: a lexical database for English', *Commun. ACM*, vol. 38, no. 11, pp. 39-41.

[60] Mokarizadeh, S., Kungas, P. & Matskin, M. 2011, 'Evaluation of a Semi-automated Semantic Annotation Approach for Bootstrapping the Analysis of Large-Scale Web Service Networks', *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, vol. 1, pp. 388-95.

[61] Nachira, F., Nicolai, A., Dini, P., Le Louarn, M. & Leon, L.R. 2007, 'Digital business ecosystems'.

[62] Negnevitsky, M. 2005, *Artificial intelligence: a guide to intelligent systems*, Pearson Education.

[63] Ohura, Y., Takahashi, K., Pramudiono, I. & Kitsuregawa, M. 2002, 'Experiments on query expansion for internet yellow page services using web log mining', paper presented to the *Proceedings of the 28th international conference on Very Large Data Bases*, Hong Kong, China.

[64] Oldham, N., Thomas, C., Sheth, A. & Verma, K. 2005, 'METEOR-S Web Service Annotation Framework with Machine Learning Classification', in J. Cardoso & A. Sheth (eds), *Semantic Web Services and Web Process Composition: First International Workshop, SWSWPC 2004, San Diego, CA, USA, July 6, 2004, Revised Selected Papers*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 137-46.

[65] Olivas, J.A., Garcés, P.J. & Romero, F.P. 2003, 'An application of the FIS-CRM model to the FISS metasearcher: Using fuzzy synonymy and fuzzy generality for representing concepts in documents', *International Journal of Approximate Reasoning*, vol. 34, no. 2–3, pp. 201-19.

[66] Osuna-Ontiveros, D., Lopez-Arevalo, I. & Sosa-Sosa, V. 2011, 'A semantic information retrieval model for focused crawling', *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on*, pp. 285-9.

[67] Paliwal, A.V., Adam, N.R. & Bornhovd, C. 2007, 'Web Service Discovery: Adding Semantics through Service Request Expansion and Latent Semantic Indexing', *IEEE International Conference on Services Computing (SCC 2007)*, pp. 106-13.

[68] Patel, C., Supekar, K., Lee, Y. & Park, E.K. 2003, 'OntoKhoj: a semantic web portal for ontology searching, ranking and classification', paper presented to the *Proceedings of the 5th ACM international workshop on Web information and data management*, New Orleans, Louisiana, USA.

[69] Patil, A.A., Oundhakar, S.A., Sheth, A.P. & Verma, K. 2004, 'Meteor-s web service annotation framework', paper presented to the *Proceedings of the 13th international conference on World Wide Web*, New York, NY, USA.

[70] Peffers, K., Tuunanen, T., Rothenberger, M.A. & Chatterjee, S. 2007, 'A Design Science Research Methodology for Information Systems Research', *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77.

[71] Protege 2016, *Protege-OWL API Programmer's Guide*, 'The Protege-OWL API is an open-source Java library for the Web Ontology Language [...]', <http://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide>.

[72] Ross, T.J. 2009, *Fuzzy logic with engineering applications*, John Wiley & Sons.

[73] Shchekotykhin, K., Jannach, D. & Friedrich, G. 2010, 'xCrawl: a high-recall crawling method for Web mining', *Knowledge and Information Systems*, vol. 25, no. 2, pp. 303-26.

[74] Shima, H. 2016, *ws4j*, 'In summer 2015, the developer (hideki.shima AT gmail.com) [...]', <https://code.google.com/archive/p/ws4j/>.

[75] Soon Ae, C. & Warner, J. 2008, 'Semantic Annotation and Search for Deep Web Services', *E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on*, pp. 389-95.

[76] Stavropoulos, T.G., Vrakas, D. & Vlahavas, I. 2013, 'Iridescent: a tool for rapid semantic annotation of web service descriptions', paper presented to the *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, Madrid, Spain.

[77] Su, C., Gao, Y., Yang, J. & Luo, B. 2005, 'An efficient adaptive focused crawler based on ontology learning', *Hybrid Intelligent Systems, 2005. HIS'05. Fifth International Conference on*, IEEE, p. 6 pp.

[78] Toch, E., Gal, A., Reinhartz-Berger, I. & Dori, D. 2007, 'A semantic approach to approximate service retrieval', *ACM Transactions on Internet Technology (TOIT)*, vol. 8, no. 1, p. 2.

[79] TrueLocal 2016, viewed 31 August 2016, <http://www.truelocal.com.au>.

[80] University, P. 2016, *WordNet*, 'What is WordNet? Any opinions, findings, and conclusions [...]', <https://wordnet.princeton.edu/>.

[81] Vehviläinen, A., Hyvönen, E. & Alm, O. 2006, 'A semi-automatic semantic annotation and authoring tool for a library help desk service', *Proceedings of the first Semantic Authoring and Annotation Workshop, ISWC-2006, Athens, GA, USA*.

[82] Walenz, B., Barton, G. & Didion, J. 2016, *JWNL (Java WordNet Library)*, 'JWNL is a Java API for accessing the WordNet relational [...]', <https://sourceforge.net/projects/jwordnet/>.

[83] Wang, H., Feng, Z., Chen, S., Xu, J. & Sui, Y. 2010, 'Constructing Service Network via Classification and Annotation', *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*, pp. 69-73.

[84] Wang, H., Liang, Y., Fu, L., Xue, G.-R. & Yu, Y. 2009, 'Efficient query expansion for advertisement search', paper presented to the *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, Boston, MA, USA.

[85] Wang, X., Huang, D., Akturk, I., Balman, M., Allen, G. & Kosar, T. 2009, 'Semantic enabled metadata management in PetaShare', *International Journal of Grid and Utility Computing*, vol. 1, no. 4, pp. 275-86.

[86] Wikipedia 2016, *Semantics*, 'Semantics (from Ancient Greek: σημαντικός sēmantikos, "significant") [...]', viewed 30 August 2016, <https://en.wikipedia.org/wiki/Semantics>.

[87] Wlodarczyk, T.W., Yi, H., Yu, X. & Rong, C. 2012, 'Evaluation of some optimisation techniques for semantic query answering on shared-nothing architecture', *International Journal of Space-Based and Situated Computing*, vol. 2, no. 1, pp. 12-22.

[88] Xia, Z., Guo, L., Liang, C., Li, X. & Yang, Z. 2007, 'Focused Crawling for Retrieving Chemical Information', in E. Corchado, J. Corchado & A. Abraham (eds),

Innovations in Hybrid Intelligent Systems, vol. 44, Springer Berlin Heidelberg, pp. 433-8.

[89] YellowPages 2016, viewed 31 August 2016, <http://www.yellowpages.com.au/>.

[90] Yelp 2016, viewed 31 August 2016, <https://www.yelp.com.au>.

[91] Zadeh, L.A. 1965, 'Fuzzy sets', *Information and Control*, vol. 8, no. 3, pp. 338-53.

[92] Zadeh, L.A. 1994, 'Fuzzy logic, neural networks, and soft computing', *Commun. ACM*, vol. 37, no. 3, pp. 77-84.

[93] Zhai, J., Cao, Y. & Chen, Y. 2008, 'Semantic information retrieval based on fuzzy ontology for intelligent transportation systems', *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, IEEE, pp. 2321-6.

[94] Zhang, R., Zettsu, K., Kidawara, Y. & Kiyoki, Y. 2011, 'Context-Sensitive Query Expansion over the Bipartite Graph Model for Web Service Search', in J.X. Yu, M.H. Kim & R. Unland (eds), *Database Systems for Advanced Applications: 16th International Conference, DASFAA 2011, Hong Kong, China, April 22-25, 2011, Proceedings, Part I*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 418-33.

[95] Zhao, W., Peng, L., Zhang, J. & Tian, C. 2012, 'A fuzzy service matching algorithm based on Bloom filter', *Communication Technology (ICCT), 2012 IEEE 14th International Conference on*, IEEE, pp. 945-50.

[96] Zheng, H.-T., Kang, B.-Y. & Kim, H.-G. 2008, 'An ontology-based approach to learnable focused crawling', *Information Sciences*, vol. 178, no. 23, pp. 4512-22.