

A Probabilistic Latent Factor Approach to Service Ranking

Gilbert Cassar, Payam Barnaghi, Klaus Moessner
Centre for Communication Systems Research
University of Surrey,
Guildford, UK, GU2 7XH
{g.cassar, p.barnaghi, k.moessner} @surrey.ac.uk

Abstract—In this paper we investigate the use of probabilistic machine-learning techniques to extract latent factors from semantically enriched service descriptions. These latent factors provide a model to represent service descriptions of any type in vector form. With this conversion, heterogeneous service descriptions can be represented on the same homogeneous plane thus achieving interoperability between different service description technologies. Automated service discovery and ranking is achieved by extracting latent factors from queries and representing the queries in vector form. Vector algebra can then be used to match services to the query. This approach is scalable to large service repositories and provides an efficient mechanism for publishing and new services after the system is deployed.

Keywords-service computing, ranking, machine-learning, LDA, semantics;

I. INTRODUCTION

Awareness of the ubiquitous presence and significant advantages of service-oriented computing is driving businesses to implement or transform their online business applications into Web Services (WS) and distributed service networks. Consequently, the Internet is shifting from a repository of data to a framework of data and service platforms. The vision of an Internet of Things (IoT) has also changed the concept of what a service is and drives the idea further away from enterprise WS. State-of-the-art research is witnessing an increase in small devices being embedded with processing capabilities [1]. In this scenario, a service can be either an enterprise software entity residing in the virtual layer or a so called real-world service which is provided by embedded systems that are linked directly to the physical world [2].

The common technologies used to describe, publish, and discover these services offer very limited expressive power to the service designers. The World Wide Web lacks a homogeneous structure (beyond that of the service interface) for describing functional and non-functional parameters in service descriptions. This results in different ways to name parameters, define parameters and describe internal processing which inhibits straightforward integration of WS [3]. The framework lacks the required foundations for implementing mechanisms such as service provisioning, service recommendation, and service composition. To achieve these high-level mechanisms, service discovery needs more machine-oriented expressive power, thus changing the process from

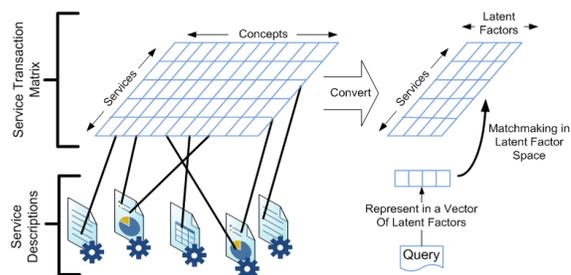


Figure 1. Abstraction of the Latent Factor Approach

syntactic content based retrieval to an automated reasoning of semantic annotated functionalities.

The aim of our research is to investigate the use of more expressive technologies to describe a service, and to use semantics, reasoning, and machine-learning techniques to process the service descriptions and generate new knowledge from what is supplied in the service descriptions; so this knowledge can be referred to by software agents or search clients to offer enhanced service search results and recommendations in different applications. Our work proposes a solution which transcends all the differences between WS technologies and creates a framework which supports the interoperability of different service description models and protocols.

This paper is a continuation of our work presented in [4] where we proposed using a probabilistic machine-learning method to extract latent factors $z_f \in Z = \{z_1, z_2, \dots, z_k\}$ from semantically enriched service descriptions for service clustering. In this paper we discuss how the latent factors can then be used to search through the services in a repository, providing an efficient approach to automated discovery and ranking of services. By describing the services in terms of latent factors, the dimensionality of the service descriptions is reduced considerably. The model is scalable to large service repositories and also provides an efficient mechanism for publishing new service descriptions to the repository. The proposed solution is interoperable and can work with different service description models. An abstract

representation of our model is shown in Figure 1.

The main contributions of this work include:

- Interoperability between different service description technologies.
- A probabilistic machine-learning approach which is completely unsupervised.
- An efficient mechanism for publication and discovery which reduces the dimensionality of all the computations involved.
- Automatic ranking of results in order of relevance.
- A process which automatically clusters all the services within a repository.
- A model which is scalable to large service repositories.

The rest of the paper is organised as follows. The next section gives a detailed discussion of the Latent-Variable approach. Section III describes the experiments used to evaluate our algorithm. Section IV provides concluding remarks on the implemented system and discusses the future work.

II. A LATENT-VARIABLE APPROACH

We propose a latent-variable approach to service ranking which relies on probabilistic machine-learning techniques to extract low dimensional latent factors from the higher dimensional concepts contained in service descriptions. The versatility of this approach resides in the inherent property of probabilistic machine-learning to assign a probability to each link between a service description and a latent factor. This property enables us to discover and rank services with a probabilistic approach.

In this section, we will first discuss the Service Description Model used in our experiments followed by an explanation of our Latent-Variable Model. We then explain how concepts are extracted from service descriptions and discuss the probabilistic machine-learning technique used to generate the latent factors for the model. Finally we explain how these latent-factors are used to provide an efficient discovery and ranking mechanism.

A. The Service Description Model

The *Web Service Description Language (WSDL)*¹ is a World Wide Web Consortium (W3C)² recommendation for describing the interface of web services [3]. The interface of a web service describes operations, inputs, and outputs of a web service but does not provide high-level descriptions about service function and profile descriptions. In the light of the new breed of services emerging in future generation networks, WSDL lacks the expressiveness needed to describe the wide spectrum of distributed services that we expect to find in such pervasive systems. Liv *et al.* [5] discuss how a service consists of different *Functional* and *Non-Functional*

parameters and how WSDL is only capable to describe a small set of Functional parameters.

The limitations of WSDL as a service description language have motivated various research contributions which focused on the creation of a richer and more expressive service description language. This gave rise to many heterogeneous service description models, all with their merits and limitations.

In our approach we assume all service descriptions are written in the Ontology Web Language for Services (OWL-S)³. OWL-S is a Service Description Model that describes the characteristics of a service by using three top-level concepts, namely *ServiceProfile*, *ServiceGrounding*, and *ServiceModel*. *ServiceProfile* provides the information needed to discover services. *ServiceGrounding* and *ServiceModel* provide information to deploy and use the service.

Various other approaches for creating service description models exist, including: SAWSDL⁴, USDL⁵, Web Service Modelling Language (WSML)⁶ and Web Service Modelling Ontology (WSMO)⁷, and SA-REST⁸. We do not aim to discuss how these models compare to each other; the aim of our work is to show how rich service descriptions can be processed in Probabilistic Latent-Variable Space and OWL-S is a Service Description Model that provides both rich expressive descriptions and well-defined semantics. However, the model can be also extended to other service description frameworks that describe similar attributes of the services.

B. A Probabilistic Latent-Variable Model

Latent-Variable Models provide a way to map high dimensional vectors into a lower dimensional representation in *Latent Semantic Space* [6]. Our approach is based on the Aspect Model (shown in Figure 2); a latent variable model which associates an unobserved class variable $z_f \in Z = \{z_1, z_2, \dots, z_k\}$ with each observation [7].

The Aspect Model is used together with probabilistic machine-learning techniques such as *Probabilistic Latent Semantic Analysis (PLSA)* [6] to discover the semantics behind the words in a text corpus, i.e. the topics which the words in the document belong to. Words are observable variables $w_j \in W = \{w_1, w_2, \dots, w_v\}$ which can be observed through the text corpus that consists of documents, topics on the other hand are latent variable which are not directly observable through the text.

Service descriptions are documents composed of a high number of concepts annotated with XML tags as shown in Figure 3. Thus a similar approach can be used to extract latent-factors from the observable concepts $c_j \in C =$

¹<http://www.w3.org/TR/wsdl20/>

²<http://www.w3.org/>

³<http://www.w3.org/Submission/OWL-S/>

⁴<http://www.w3.org/2002/ws/sawSDL/>

⁵<http://www.internet-of-services.com/index.php?id=24>

⁶<http://www.wsmo.org/wsml/>

⁷<http://www.wsmo.org/>

⁸<http://www.w3.org/Submission/SA-REST/>

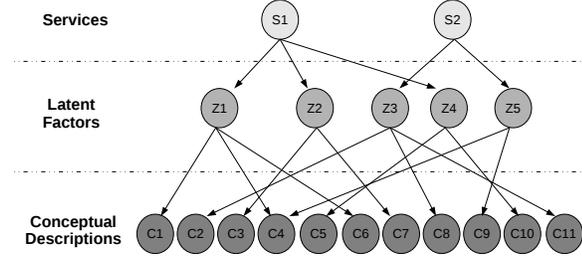


Figure 2. A graphical representation of the Aspect Model.

$\{c_1, c_2, \dots, c_v\}$ contained in a service description. Once the latent variables $z_f \in Z = \{z_1, z_2, \dots, z_k\}$ are identified, services can be described as a multinomial probability distribution $P(s_i|z_f)$ where s_i is the service description (or document) describing service i . The representation of a service with these latent variables reflects the likelihood that the service belongs to certain concept groups [8].

To construct our model, we first consider the joint probability of an observed pair. This approach is described by Hoffman in [6]. The joint probability of an observed pair $P(s_i, c_j)$ is given by:

$$P(s_i, c_j) = P(s_i) P(c_j|s_i) \quad (1)$$

where:

$$P(c_j|s_i) = \sum_{f=1}^k P(z_f|s_i) P(c_j|z_f) \quad (2)$$

assuming that a service description and a concept are conditionally independent given the latent factor.

By adopting this model we indirectly associates the concepts in a service description $c_j \in C = \{c_1, c_2, \dots, c_v\}$ to their corresponding service descriptions $s_i \in S = \{s_1, s_2, \dots, s_n\}$ by introducing an intermediate layer of latent factors $z_f \in Z = \{z_1, z_2, \dots, z_k\}$. Thus we are getting dimensionality reduction by mapping a high dimensional concept to service description matrix $P(s, c)$ into a lower k -dimension latent semantic space [8].

By substituting equation 2 in equation 1, we obtain:

$$P(s, c) = \sum_{f=1}^k P(z_f) P(s|z_f) P(c|z_f) \quad (3)$$

The parameters $P(z)$, $P(s|z)$, and $P(c|z)$ can be found using a model fitting technique such as the Expectation Maximization (EM) algorithm as described in [6]. Once the algorithm is trained and the parameters are found, we can

express service descriptions in terms of latent-factors using the learned probability distribution $P(s|z)$.

Ma *et al.* [8] present a service discovery mechanism called CPLSA which uses PLSA to extract latent factors from WSDL service descriptions after the search is narrowed down to a small cluster using a K-Means algorithm. However, this approach does not make use of the semantics in service descriptions in order to enhance the feature extraction process. In our previous work [4] we also showed how latent factors extracted from service descriptions can be used directly to cluster services in a repository; obtaining a more efficient clustering strategy than the one obtained by a K-Means algorithm.

C. Feature Extraction

Feature Extraction is used to convert data to a numerical form in order to facilitate computational analysis of the data. The *Text Frequency and Inverse Text Frequency (TF/IDF)* [9] are the common approaches in information retrieval to construct the *Document Matrix* (*i.e.* extracted features) in a text document corpus. Ma *et al.* use this technique in [8] to represent a dataset of WSDL service descriptions in the form of a *Service Transaction Matrix*, PLSA could then be used on the Service Transaction Matrix to extract latent factors from the data. TF/IDF gives a weight w_{ij} to every term j in a service description document i using the equation:

$$w_{ij} = tf_{ij} \cdot \log\left(\frac{n}{n_j}\right) \quad (4)$$

where tf_{ij} is the frequency of term j in service description document i , n is the total number of service descriptions documents in the dataset, and n_j is the number of services that contain term j .

We build a similar Service Transaction Matrix as the one described in [8]. Functional parameters and the profile data are extracted from the OWL-S descriptions using a reasoner. The functional parameters in OWL-S consist of the properties *hasInput*, *hasOutput*, *hasParameter*, *hasPrecondition*, and *hasResult* found in the service model.

In order to make the features fit in the vector space model, the property name and the type are appended together to produce a new term that is used as a concept in constructing the Service Transaction Matrix. For example, from the OWL-S sample shown in Figure 3 `<profile:hasInput rdf:resource="#_ORGANIZATION"/>` becomes *hasInput_ORGANIZATION*. This way, a service in which *ORGANIZATION* is defined as input will be distinguished from a service in which *ORGANIZATION* is defined as output. Text frequency is then used to create the Service Transaction Matrix from the constructed concepts.

D. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation is an unsupervised machine-learning technique which uses a generative probabilistic

```

<service:Service rdf:ID="ORGANIZATION_DIAGNOSTICPROCESS_SERVICE">
  <service:presents
    rdf:resource="#ORGANIZATION_DIAGNOSTICPROCESS_PROFILE"/>
  <service:describedBy
    rdf:resource="#ORGANIZATION_DIAGNOSTICPROCESS_PROCESS"/>
  <service:supports
    rdf:resource="#ORGANIZATION_DIAGNOSTICPROCESS_GROUNDING"/>
</service:Service>

<profile:Profile rdf:ID="ORGANIZATION_DIAGNOSTICPROCESS_PROFILE">
  <service:isPresentedBy
    rdf:resource="#ORGANIZATION_DIAGNOSTICPROCESS_SERVICE"/>
  <profile:serviceName xml:lang="en"> ODGService </profile:serviceName>
  <profile:textDescription xml:lang="en">
    ODG service informs for diagnosing system available in the given organization, such
    like hospital.
  </profile:textDescription>
  <profile:hasInput rdf:resource="#_ORGANIZATION"/>
  <profile:hasOutput rdf:resource="#_DIAGNOSTICPROCESS"/>
  <profile:has_process
    rdf:resource="#ORGANIZATION_DIAGNOSTICPROCESS_PROCESS"/>
</profile:Profile>

```

Figure 3. Sample from an OWL-S Service Description Document.

model for collections of discrete data [10]. LDA introduces a Dirichlet prior on the document-topic distribution in order to simplify the problem of statistical inference [11]. The principal of LDA is the same as that of PLSA: mapping high-dimensional count vectors (such as the ones yielded by TF/IDF in vector space model) to a lower dimensional representation in latent semantic space.

Using the same notation described in section II-B, with LDA the probability of the j th concept occurring in a given service description document is:

$$P(c_j) = \sum_{f=1}^k P(c_j|z_i = f) P(z_i = f) \quad (5)$$

where z_i is a latent factor from which the i th word was drawn, $P(z_i = f)$ is the probability of latent factor f being the latent factor from which c_j was drawn, and $P(c_j|z_i = f)$ is the probability of having concept c_j given the f th latent factor.

The generative model of LDA is obtained by letting:

$$\Phi^{(j)} = P(c|z = f) \quad (6)$$

$$\Theta^{(s)} = P(z) \quad (7)$$

Instead of estimating $P(s|z)$ and $P(c|z)$ as in PLSA, the LDA generative model estimates intermediary variables (*i.e.* Φ , Θ , and z). Different methods can be used to train the algorithm and estimate these parameters Blei *et al.* [10] use variational inference with *Expectation Maximisation (EM)* algorithm. Wang *et al.* [11] use a method based on Gibbs Sampling which was proposed in [12] and [13].

E. Automated Discovery and Ranking using Latent Factors

Automated Service Discovery aims to provide an approach to find and recommend services; however the way the results are presented to the user is also a very important

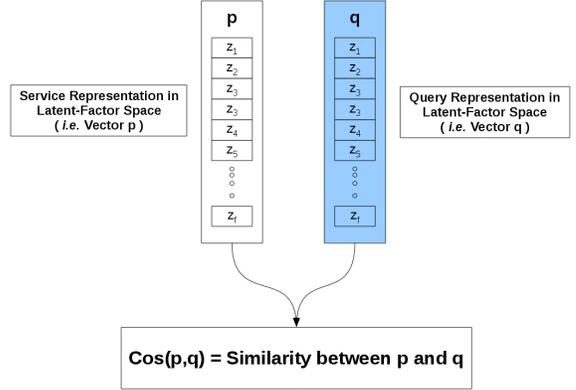


Figure 4. A graphical representation of query processing in Latent-Factor Space.

factor when it comes to the quality of user or service consumer experience. The amount of services available, even after narrowing the list down with an enhanced discovery algorithm, is very large and any potential service consumer would like to narrow the search down to the best five or ten matching results [14]. The key mechanism for narrowing down such results is Service Ranking. Service Ranking aims to give a value of relevance to each item returned by the automated discovery algorithm and proceeds to order the results in descending order starting from the most relevant ones. Probabilistic machine-learning techniques have the inherent property of assigning a probability to the likelihood of a service description belonging to each of the generated latent factors. We propose to use these generated probabilities as the base criteria for computing the similarity between a service description and a user query. The services can then be ranked based on the relevancy to the submitted query. We investigate two latent factor models for achieving automated discovery and ranking; one based on PLSA and another one based on LDA. The PLSA was implemented using the PennAspect⁸ model which uses maximum likelihood to compute the three parameters: $P(w|z)$, $P(d|z)$, and $P(z)$. Half of the the dataset is used to train the algorithm and the other half is used for validation in order to prevent overfitting [11]. The LDA model was implemented using LingPipe⁹ toolkit. This toolkit uses Gibbs sampling to train the algorithm and to obtain the parameter $P(s|z)$.

Before processing a query, the latent-factors must be extracted from the query so that the query processing can take place in latent factor space as shown in Figure 4. This process is called folding-in [11] and can be applied to queries and also new service descriptions which need to be published in the repository. For PLSA, once the model is trained, new service descriptions or queries can be folded-in

⁸http://www.cis.upenn.edu/ungar/Datamining/software_dist/PennAspect/index.html

⁹<http://alias-i.com/lingpipe/>

by using the following [8].

$$P(z_f|s_{new}) = \frac{P(s_{new}|z_f) \cdot P(z)}{\sum_{j=1}^k P(s_{new}|z_j)} \quad (8)$$

For LDA, after training the algorithm, new service descriptions or queries can be folded in by using Gibbs sampling with fixed service description to concept probabilities and sampling the assignments of concepts to latent-factors in the new service description [11]. This property of LDA and PLSA makes the models scalable to large service repositories (such as the ones we expect to deal with in our scenario) since the latent-variables need to be learned only once. Any new service description added to the models after the training phase can be folded into the model without the need to retrain the model all over again.

Once the distribution of latent factors over the query is obtained, we can compute the similarity between the query and a service by computing the *Multidimensional Angle* between a vector containing the query’s distribution of latent factors q and a vector containing the service’s distribution of latent factors p . This proximity measure applies cosine of the angle between two vectors and is used in various approaches dealing with vector-space analysis of data [15], [8].

The multidimensional angle between vectors p and q can be calculated using equation 9.

$$\cos(p, q) = \frac{p \cdot q}{\|p\| \cdot \|q\|} = \frac{\sum_{i=1}^f p_i q_i}{\sqrt{\sum_{i=1}^f p_i^2} \sqrt{\sum_{i=1}^f q_i^2}} \quad (9)$$

where f is the number of latent-factors.

For both models, service discovery in latent factor space consists of computing the multidimensional angle between the query and every service in the repository. The services are then ranked in order of their similarity score to the query thus automatically obtaining an efficient ranking of the services retrieved.

III. EVALUATION

The dataset of service descriptions used in this experiment was obtained from the OWL-S service retrieval test collection; called OWLS-TC v3¹⁰. This dataset consists of 1007 service descriptions defined in OWL-S form. The services are divided into seven categories and a total of 29 sample queries are provided together with a relevant answer set for each query. Table I shows the number of services and queries belonging to each of the seven categories.

We evaluated our approach by calculating the *Precision* and *Recall* for the results obtained by each of the sample queries. These are standard evaluation techniques used in *Information Retrieval* to measure the accuracy of a search

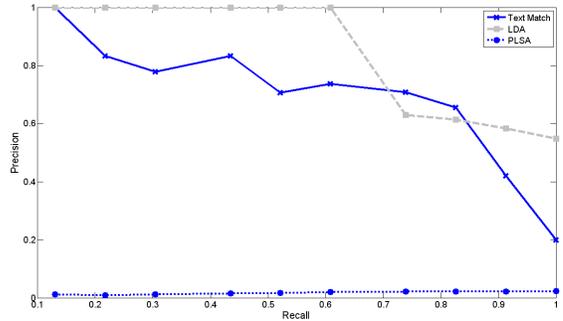


Figure 5. Comparison of Recall and Precision for the query: *title_comedyfilm_service.owl*

mechanism with respect to completeness of the results returned. Precision measures how many of the results returned are relevant to the query and recall measures how many relevant services have been retrieved from the set of relevant services present in the whole repository.

The sample queries used are all in the format of OWL-S templates and contain the semantic requirements together with a text description of the service being searched for. For PLSA and LDA, these query templates were converted to latent factor space using folding in and matched to the services in latent factor space. As a means to compare our two methods we use the full-featured text search engine: Apache Lucene¹¹. The text description of the query was submitted to the text search engine and the results returned were based on text matching.

Table I
NUMBER OF SERVICES AND QUERIES FOR EACH DOMAIN.

Domain	Services	Query
Education	284	6
Food	34	1
Medical	73	1
Travel	165	6
Communication	58	2
Economy	359	12
Weapon	40	1

A. Results

The comparison of Precision and Recall obtained for two of the sample queries are shown in Figures 5, 6. The average Precision and Recall obtained over all 29 queries for Text-Match, LDA, and PLSA are shown in Figure 7.

B. Discussion

The discussion of results here.

¹⁰<http://www.semwebcentral.org/projects/owl-s-tc/>

¹¹<http://lucene.apache.org/java/docs/index.html>

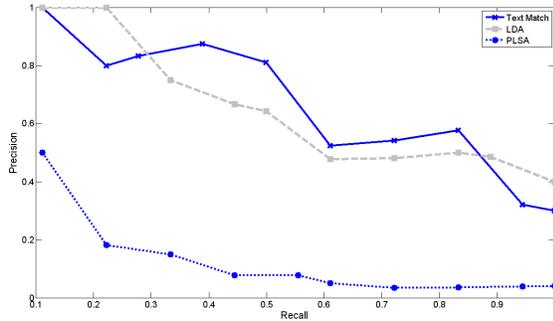


Figure 6. Comparison of Recall and Precision for the query: *title_comedyfilm_service.owl*

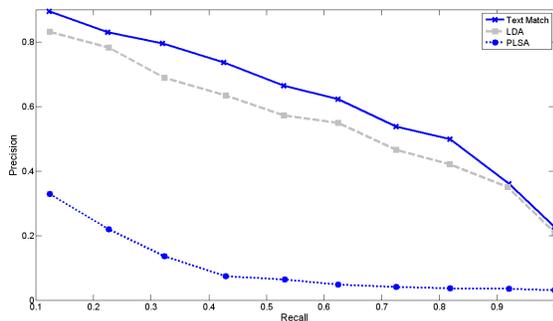


Figure 7. Average Recall and Precision over 29 queries

IV. CONCLUSION AND FUTURE WORK

In this paper we presented a latent-variable model which can map high-dimensionality concepts found in service descriptions to a lower-dimensionality space composed of latent-factors. This approach makes it possible to understand the hidden meaning behind a set of concepts and describe the services in terms of latent-factors rather than independent concepts. It is important to note that although this work is focused on OWL-S web service descriptions, our approach can be applied to other service description models such as WSMO, SA-REST, SA-WSDL or any other semantically annotated framework. By transforming service descriptions into the latent-variable space, this approach creates a homogeneous intermediary level between the heterogeneous service descriptions mentioned in Section II-A and the advanced automated discovery mechanisms discussed in Section I. Any kind of services which may be found in future pervasive systems can be mapped into the latent-variable model as long as they follow a semantically annotated service description model. Service Description Models which are more expressive and detailed will be mapped more accurately into the latent-variable space.

The probabilistic methods used in this paper (i.e. LDA and PLSA) can be trained using a small percentage of the whole

dataset, the rest of the service descriptions can be folded into the model as described in Section II-E. This makes these algorithms scalable to large service repositories. The inherent property of probabilistic machine-learning to assign a probability to each link between a service description and a latent factor enables us to discover and rank services with a probabilistic approach. The results obtained from comparing the two methods to a full-featured text search engine show that the PLSA model does not capture enough information in the latent factors to make it capable to search for information in Latent Factor space. On the other hand, the results show that LDA can capture Latent information much better than PLSA and searching for service in Latent Factor space with LDA yields results which are in an acceptable range compared to the full-featured text search engine.

Although LDA does not exceed text matching in performance, it still provides a scalable and interoperable solution for automated service discovery in large service repositories. The model also provides a versatile platform on top of which more machine-oriented mechanisms can be developed. Future work will focus on developing high-level mechanisms on top of the existing method; such as service provisioning, service recommendation, and service composition.

ACKNOWLEDGMENT

This paper describes work undertaken in the context of the IoT-A project, IoT-A: Internet of Things - Architecture (<http://www.ietf-a.eu/public>) contract number: 257521. The authors would like to thank Dr. Wei Wang from the University of Nottingham Malaysia Campus for his comments and suggestions.

REFERENCES

- [1] K. Van Laerhoven, "Iswc 2010: The latest in wearable computing research," *Pervasive Computing, IEEE*, vol. 10, no. 1, pp. 8–10, 2011.
- [2] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *Services Computing, IEEE Transactions on*, vol. 3, no. 3, pp. 223–235, 2010.
- [3] A. Segev and E. Toch, "Context-based matching and ranking of web services for composition," *IEEE Transactions on Services Computing*, vol. 99, no. PrePrints, pp. 210–222, 2009.
- [4] G. Cassar, P. Barganghi, and K. Moessner, "Probabilistic methods for service clustering," in *Proceedings of the 4th International Workshop on Semantic Web Service Matchmaking and Resource Retrieval, Organised in conjunction with the International Semantic Web Conference (ISWC) 2010*, November 2010.
- [5] C. Liu, Y. Peng, and J. Chen, "Web services description ontology-based service discovery model," in *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 633–636.

- [6] T. Hofmann, “Probabilistic latent semantic analysis,” in *Proc. of Uncertainty in Artificial Intelligence, UAI99*, 1999, pp. 289–296.
- [7] T. Hofmann, J. Puzicha, and M. I. Jordan, “Learning from dyadic data,” in *Proceedings of the 1998 conference on Advances in neural information processing systems II*. Cambridge, MA, USA: MIT Press, 1999, pp. 466–472.
- [8] J. Ma, Y. Zhang, and J. He, “Efficiently finding web services using a clustering semantic approach,” in *CSSSIA '08: Proceedings of the 2008 international workshop on Context enabled source and service selection, integration and adaptation*. New York, NY, USA: ACM, 2008, pp. 1–8.
- [9] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [10] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
- [11] W. Wei, P. Barnaghi, and A. Bargiela, “Probabilistic topic models for learning terminological ontologies,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1028–1040, 2010.
- [12] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” *Proceedings of the National Academy of Sciences*, vol. 101, no. Suppl. 1, pp. 5228–5235, April 2004.
- [13] M. Steyvers and T. Griffiths, *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum, 2007, ch. Probabilistic topic models.
- [14] U. Sehgal, K. Kaur, and P. Kumar, “The anatomy of a large-scale hyper textual web search engine,” in *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, vol. 2, 2009, pp. 491–495.
- [15] C. Platzer, F. Rosenberg, and S. Dustdar, “Web service clustering using multidimensional angles as proximity measures,” *ACM Trans. Internet Technol.*, vol. 9, no. 3, pp. 1–26, 2009.