Correlated Topic Model for Web Services Ranking

Mustapha AZNAG*, Mohamed QUAFAFOU* and Zahi JARIR**
* Aix-Marseille University, LSIS UMR 7296, France.
{mustapha.aznag,mohamed.quafafou}@univ-amu.fr
** University of Cadi Ayyad, LISI Laboratory, FSSM, Morocco.
jarir@uca.ma

***Abstract*—With the increasing number of published Web services providing similar functionalities, it's very tedious for a service consumer to make decision to select the appropriate one according to her/his needs. In this paper, we explore several probabilistic topic models: Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA) and Correlated Topic Model (CTM) to extract latent factors from web service descriptions. In our approach, topic models are used as efficient dimension reduction techniques, which are able to capture semantic relationships between word-topic and topic-service interpreted in terms of probability distributions. To address the limitation of keywords-based queries, we represent web service description as a vector space and we introduce a new approach for discovering and ranking web services using latent factors. In our experiment, we evaluated our Service Discovery and Ranking approach by calculating the precision (P@n) and normalized discounted cumulative gain (NDCGn).**

***Keywords*—Web service, Data Representation, Discovery, Ranking, Machine Learning, Topic Models**

I. Introduction

Web services[1] [25] are defined as a software systems designed to support interoperable machine-to-machine interaction over a network. They are loosely coupled reusable software components that encapsulate discrete functionality and are distributed and programmatically accessible over the Internet. They are self contain, modular business applications that have open, internet-oriented, standards based interfaces [2]. The Service Oriented Architecture (SOA) is a model currently used to provide services on the internet. The SOA follows the find-bind-execute paradigm in which service providers register their services in public or private registries, which clients use to locate web services. SOA services have self-describing interfaces in platform-independent XML documents. Web Services Description Language (WSDL) is the standard language used to describe services. Web services communicate with messages formally defined via XML Schema. Different tasks like matching, ranking, discovery and composition have been intensively studied to improve the general web services management process. Thus, the web services community has proposed different approaches and methods to deal with these tasks. Empirical evaluations are generally proposed considering different simulation scenarios. Nowadays, we are moving from web of data to web of services as the number of UDDI Business Registries (URBs) is increasing. Moreover, the number of hosts that offer available web services is also increasing significantly. Consequently, discovering services which can match with the user query is becoming a challenging and an important task. The keyword-based discovery mechanism supported by the most existing services search engines suffers from some key problems:

- User finds difficulties to select a desired service which satisfies his requirements as the number of retrieved services is huge.
- Keywords are insufficient in expressing semantic concepts. This is due to the fact that the functional requirements (keywords) are often described by natural language.

To enrich web service description, several Semantic Web methods and tools are developed, for instance, the authors of [10], [23], [1] use ontology to annotate the elements in web services. Nevertheless, the creation and maintenance of ontologies may be difficult and involve a huge amount of human effort [3], [14].

With the increasing number of published Web services providing similar functionalities, it's very tedious for a service consumer to make decision to select the appropriate one according to her/his needs. Therefore mechanisms and techniques are required to help consumers to discover which one is better. In this case one of the major filters adopted to evaluate these services is using Quality of Service (QoS) as a criterion. Generally QoS can be defined as an aggregation of non-functional attribute that may influence the quality of the provided Web service [26], [21], [17]. Although, in various approaches [26], [17] the authors propose to calculate an averall score that combines the quality of service (availability, response time, ...) and use it to classify the web services.

To address the limitation of keywords-based queries, we represent web service description as a vector and introduce a new approach for discovering and ranking web services based on probabilistic topic models. The probabilistic topic models are a way to deal with large volumes of data by discovering their hidden thematic structure. Their added value is that they can treat the textual data that have not been manually categorized by humans. The probabilistic topic models use their hidden variables to discover the latent semantic structure in large textual data.

In this paper we investigate using probabilistic machine-learning methods to extract latent factors $z_f \in Z = \{z_1, z_2, ..., z_k\}$ from service descriptions. We will explore several probabilistic topic models : PLSA (Probabilistic latent

[1] http://www.w3.org/standards/webofservices

semantic analysis), LDA (Latent Dirichlet Allocation) and CTM (Correlated Topic Model) and use them to analyze search in repository of web services and define which achieves the best results. By describing the services in terms of latent factors, the dimensionality of the system is reduced considerably. The latent factors can then also be used to provide an efficient discovery and ranking system. In our experiments, we consider that web services are mixtures of hidden topics, where a topic defines a probability distribution over words.

The rest of this paper is organized as follows. In Section II we describe in detail our Service Discovery and Ranking approach. Section III describes the experimental evaluation. Section IV provides an overview of related work. Finally, the conclusion and future work can be found in Section V.

II. Web Service Discovery and Ranking Approach

In this section, we will first describe the necessary preprocessing of WSDL document to construct a web service representation. We then discuss the probabilistic machine-learning techniques used to generate the latent factors. Finally, we explain how these latent factors are used to provide an efficient discovery and ranking mechanism.

A. Web Service Representation

Generally, every web service has a WSDL (Web Service Description Language) document that contains the description of the service. The WSDL document is an XML-based language, designed according to standards specified by the W3C, that provides a model for describing web services. It describes one or more services as collections of network endpoints, or ports. It provides the specifications necessary to use the web service by describing the communication protocol, the message format required to communicate with the service, the operations that the client can invoke and the service location. Two versions of WSDL recommendation exist: the 1.1[2] version, which is used in almost all existing systems, and the 2.0[3] version which is intended to replace 1.1. These two versions are functionally quite similar but have substantial differences in XML structure.

To manage efficiently web service descriptions, we extract all features that describe a web service from the WSDL document. We recognize both WSDL versions (1.1 and 2.0). During this process, we proceed in two steps. The first step consists of checking availability of web service and validating the content of WSDL document. The second step is to get the WSDL document and read it directly from the WSDL URI to extract all information of the document.

Before representing web services as TF-IDF (Text Frequency and Inverse Frequency) [22] vectors, we need some preprocessing. There are commonly several steps:

- *Features extraction* extracts all features that describe a web service from the WSDL document, such as service name and documentation, messages, types and operations.
- *Tokenization:* Some terms are composed by several words, which is a combination of simple terms (*e.g., get_ComedyFilm_MaxPrice_Quality*). We use therefore regular expression to extract these simple terms (*e.g., get, Comedy, Film, Max, Price, Quality*).
- *Tag and stop words removal*: This step removes all HTML tags, CSS components, symbols (punctuation, etc.) and stop words, such as 'a', 'what', etc. The Standford POS Tagger[4] is then used to eliminate all the tags and stop words and only words tagged as nouns, verbs and adjectives are retained. We also remove the WSDL specific stopwords, such as *host, url, http, ftp, soap, type, binding, endpoint, get, set, request, response*, etc.
- *Word stemming*: We need to stem the words to their origins, which means that we only consider the root form of words. In this step we use the Porter Stemmer [19] to remove words which have the same stem. Words with the same stem will usually have the same meaning. For example, 'computer', 'computing' and 'compute' have the stem 'comput'. The Stemming process is more effective to identify the correlation between web services by representing them using these common stems (root forms).
- *Service Matrix construction*: After identifying all the functional terms, we calculate the frequency of these terms for all web services. We use the Vector Space Model (VSM) technique to represent each web service as a vector of these terms. In fact, it converts service description to vector form in order to facilitate the computational analysis of data. In information retrieval, VSM is identified as the most widely used representation for documents and is a very useful method for analyzing service descriptions. The TF-IDF algorithm [22] is used to represent a dataset of WSDL documents and convert it to VSM form. We use this technique, to represent a services descriptions in the form of *Service Matrix*. In the service matrix, each row represents a WSDL service description, each column represents a word from the whole text corpus (vocabulary) and each entry represents the TF-IDF weight of a word appearing in a WSDL document. TF-IDF gives a weight w_{ij} to every term j in a service description i using the equation: $w_{ij} = tf_{ij}.\log(\frac{n}{n_j})$. Where tf_{ij} is the frequency of term j in WSDL document i, n is the total number of WSDL documents in the dataset, and n_j is the number of services that contain term j.

B. A Probabilistic Topic Model Approach

Service Discovery and Selection aim to find web services with user required functionalities. While Service Discovery process assumes that services with similar functionalities should be discovered, Service Selection and Ranking aim to find a proper services with the best user desired quality of services. Thus, Service Ranking aims to give a value of

[2]http://www.w3.org/TR/wsdl

[3]http://www.w3.org/TR/wsdl20/

[4]http://nlp.stanford.edu/software/tagger.shtml

relevance to each service returned by the discovery process and proceeds to order the results in descending order starting from the most relevant ones. In our approach, we apply probabilistic machine-learning techniques; Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA) and Correlated Topic Model (CTM); to extract latent factors (or topics) $z_f \in Z = \{z_1, z_2, ..., z_k\}$ from web service descriptions (i.e., *Service Matrix*). In our work, topic models are used as efficient dimension reduction techniques, which are able to capture semantic relationships between *word-topic* and *topic-service* interpreted in terms of probability distributions. In our context, an observed event corresponds to occurrence of a word w in a service description s. We propose to use the learned latent factors 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.

The Probabilistic Latent Semantic Analysis (PLSA) is a generative statistical model for analyzing co-occurrence of data. PLSA is based on the aspect model [11]. Considering observations in the form of co-occurrences (s_i, w_j) of words and services, PLSA models the joint probability of an observed pair $P(s_i, w_j)$ obtained from the probabilistic model is shown as follows [11]:

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

We assume that service descriptions and words are conditionally independent given the latent factor. We have implemented the PLSA model using the PennAspect[5] model which uses maximum likelihood to compute the parameters. The dataset was divided into two equal segments which are then transformed into the specific format required by the PennAspect. We use words extracted from service descriptions and create a PLSA model. Once the latent variables $z_f \in Z = \{z_1, z_2, ..., z_k\}$ are identified, services can be described as a multinomial probability distribution $P(z_f|s_i)$ where s_i is the description of the service i.The representation of a service with these latent variables reflects the likelihood that the service belongs to certain concept groups [16]. To construct a PLSA model, we first consider the joint probability of an observed pair $P(s_i, w_j)$ (Equation 1). The parameters $P(z)$, $P(s|z)$ and $P(w|z)$ can be found using a model fitting technique such as the Expectation Maximization (EM) algorithm [11].

The Latent Dirichlet Allocation (LDA) is a probabilistic topic model, which uses a generative probabilistic model for collections of discrete data [4]. LDA is an attempt to improve the PLSA by introducing a Dirichlet prior on service-topic distribution. As a conjugate prior for multinomial distributions, Dirichlet prior simplifies the problem of statistical inference. The principle of LDA is the same as that of PLSA: mapping high-dimensional count vectors to a lower dimensional representation in latent semantic space. Each word w in a service description s is generated by sampling a topic z from topic distribution, and then sampling a word from topic-word distribution. The probability of the ith word occurring in a given service is given by Equation 2:

$$P(w_i) = \sum_{f=1}^{k} P(w_i|z_i = f)P(z_i = f) \quad (2)$$

Where z_i is a latent factor (or topic) from which the *i*th word was drawn, $P(z_i = f)$ is the probability of topic f being the topic from which w_i was drawn, and $P(w_i|z_i = f)$ is the probability of having word w_i given the *f*th topic.

Let $\theta^{(s)} = P(z)$ refer to the multinomial distribution over topics in the service description s and $\phi^{(j)} = P(w|z = j)$ refer to the multinomial distribution over words for the topic j. There are various algorithms available for estimating parameters in the LDA: Variational EM [4] and Gibbs sampling [24]. In this paper, we adopt an approach using Variational EM. See [4] for further details on the calculations.

For the LDA training, we used Blei's implementation[6], which is a C implementation of LDA using Variational EM for Parameter Estimation and Inference. The key objective is to find the best set of latent variables that can explain the observed data. This can be made by estimating $\phi^{(j)}$ which provides information about the important words in topics and $\theta^{(s)}$ which provides the weights of those topics in each web service.

The Correlated Topic Model (CTM) is another probabilistic topic model that enhances the basic LDA [4], by modeling of correlations between topics. One key difference between LDA and CTM is that in LDA, there is an independence assumption between topics due to the Dirichlet prior on the distribution of topics. In fact, under a Dirichlet prior, the components of the distribution are independent whereas the logistic normal used in CTM, models correlation between the components through the covariance matrix of the normal distribution. However, in CTM, a topic may be consistent with the presence of other topics. Assume we have *S* web services as a text collection, each web service *s* contains N_s word tokens, *T* topics and a vocabulary of size *W*. The Logistic normal is obtained by :

- For each service, draw a K-dimensional vector η_s from a multivariate Gaussian distribution with mean μ and covariance matrix Σ : $\eta_s \sim \mathcal{N}(\mu, \Sigma)$
- We consider the mapping between the mean parameterization and the natural parameterization: $\theta = f(\eta_i) = \frac{\exp \eta}{\sum_i \exp \eta_i}$
- Map η into a simplex so that it sums to 1.

The main problem is to compute the posterior distribution of the latent variables given a web service : $P(\eta, z_{1:N}, w_{1:N})$. Since this quantity is intractable, we use approximate techniques. In this case, we choose variational methods rather than gibbs sampling because of the non-conjugacy between logistic normal and multinomial. The problem is then to bound the log probability of a web service :

[5]http://cis.upenn.edu/ ungar/Datamining/software_dist/PennAspect/

[6]http://www.cs.princeton.edu/ blei/lda-c/

$$\begin{aligned}\log P(w_{1:N}|\mu,\Sigma,\beta) &\geq E_q[\log P(\eta|\mu,\Sigma)] \\ &+ \sum_{n=1}^{N} E_q[\log P(z_n|\eta)] \\ &+ \sum_{n=1}^{N} E_q[\log P(w_n|z_n,\beta)] \\ &+ H(q)\end{aligned} \tag{3}$$

The expectation is taken with respect to a variational distribution of the latent variables :

$$q(\eta, z|\lambda, \nu^2, \phi) = \prod_{i=1}^{K} q(\eta_i|\lambda_i, \nu_i^2) \prod_{n=1}^{N} q(z_n|\phi_n) \tag{4}$$

and $H(q)$ denotes the entropy of that distribution (See [5] for more details).

Given a model parameters $\{\beta_{1:K}, \mu, \Sigma\}$ and a web service $w_{1:N}$, the variational inference algorithm optimizes the lower bound (Equation 3)) with respect to the variational parameters using the variational EM algorithm. In the E-step, we maximize the bound with respect to the variational parameters by performing variational inference for each web service. In the M-step, we maximize the bound with respect to the model parameters. The E-step and M-step are repeated until convergence.

For the CTM training, we used the Blei's implementation[7], which is a C implementation of Correlated Topic Model using Variational EM for Parameter Estimation and Inference. We estimate the *topic-service* distribution by computing: $\theta = \frac{\exp(\eta)}{\sum_i \exp(\eta_i)}$. Where $\exp(\eta_i) = \exp(\lambda_i + \frac{\nu_i^2}{2})$ and the variational parameters $\{\lambda_i, \nu_i^2\}$ are respectively the mean and the variance of the normal distribution. Then, we estimate the *topic-word* distribution ϕ by calculating the exponential of the log probabilities of words for each topic.

After training the three probabilistic topic model, a set of matched services can be returned by comparing the similarity between the query and services in the dataset. We propose to use the probabilistic topic model to discover and rank the web services that match with the user query. Let $Q = \{w_1, w_2, \ldots, w_n\}$ be a user query that contains a set of words w_i produced by a user. In our approach, we use the generated probabilities θ and ϕ as the base criteria for computing the similarity between a service description and a user query. For this, we model information retrieval as a probabilistic query to the topic model. We note this as $P(Q|s_i)$ where Q is the set of words contained in the query. Thus, using the assumptions of the topic model, $P(Q|s_i)$ can be calculated by equation 5.

$$P(Q|s_i) = \prod_{w_k \in Q} P(w_k|s_i) = \prod_{w_k \in Q} \sum_{z=1}^{T} P(w_k|z_f) P(z_f|s_i) \tag{5}$$

The most relevant services are the ones that maximize the conditional probability of the query $P(Q|s_i)$. Consequently, relevant services are ranked in order of their similarity score to the query. Thus, we obtain automatically an efficient ranking of the services retrieved.

[7]http://www.cs.princeton.edu/ blei/ctm-c/index.html

We propose also to use an other approach based on the proximity measure called *Multidimentional Angle* (also known as *Cosine Similarity*); a measure which uses the cosine of the angle between two vectors [20], [7]. In the first time, we represent the user's query as a distribution over topics. Thus, for each topic z_f we calculate the relatedness between query Q and z_f based on $topic-word$ distribution ϕ using Equation 6.

$$P(Q|z_f) = \prod_{w_i \in Q} P(w_i|z_f) \tag{6}$$

Then, we calculate the similarity between the user's query and a web service by computing the Cosine Similarity between a vector containing the query's distribution over topics q and a vector containing the service's distribution of topics p. The multidimensional angle between a vector p and a vector q can be calculated using Equation 7:

$$Cos(p,q) = \frac{p.q}{\| p \| . \| q \|} = \frac{\sum_{i=1}^{t} p_i q_i}{\sqrt{\sum_{i=1}^{t} p_i^2 \sum_{i=1}^{t} q_i^2}} \tag{7}$$

where t is the number of topics.

In our experiments, we will compare the results obtained for the two methods (i.e. Conditional Probability, Multidimentional Angle) for the three probabilistic topic models.

III. Evaluation

A. Web Services Corpus

Our experiments are performed out based on real-world web services obtained from [27]. The WSDL corpus consists of over 1051 web services from 8 different application domains. Each web service belongs to one out of eight service domains named as: Communication, Education, Economy, Food, Travel, Medical and Military. Table I lists the number of services from each domain.

Before applying the proposed Web Service Discovery and Ranking, we deal the WSDL corpus. The objective of this preprocessing is to identify the functional terms of services, which describe the semantics of their functionalities. WSDL corpus processing consists of several steps: *Features extraction*, *Tokenization:*, *Tag and stop words removal*, *Word stemming* and *Service Matrix construction* (See Section II-A).

#	Domains	Number of services
1	Communication	59
2	Economy	354
3	Education	264
4	Food	41
5	Geography	60
6	Medical	72
7	Travel	161
8	Military	40
	Total	**1051**

TABLE I: Domains of Web services

We evaluated the effectiveness of our Web Service Discovery and Ranking for the three probabilistic topic models (labeled *PLSA*, *LDA* and *CTM*) using both methods Conditional

Probability (labeled *CP*) and Multidimentional Angle (labeled *MA*). The probabilistic methods are compared with a text-matching approach (labeled *Text-Search*). For this experiment, we use the services description collected from the WSDL corpus. As described previously, the services are divided into eight domains and some queries templates are provided together with a relevant response set for each query. The relevance sets for each query consists of a set of relevant service and each service *s* has a graded relevance value $relevance(s) \in \{1, 2, 3\}$ where 3 denotes *high relevance* to the query and 1 denotes a *low relevance*.

B. Evaluation Metrics

In order to evaluate the accuracy of our approach, we compute two standard measures used in *Information Retrieval*: *Precision at n* (*Precision@n*) and *Normalised Discounted Cumulative Gain* ($NDCG_n$). These evaluation techniques are used to measure the accuracy of a search and matchmaking mechanism.

1) Precision@n: In our context, $Precision@n$ is a measure of the precision of the service discovery system taking into account the first n retrieved services. Therefore, $Precision@n$ reflects the number of services which are relevant to the user query. The precision@n for a list of retrieved services is given by Equation 8:

$$Precision@n = \frac{|RelevantServices \cap RetrievedServices|}{|RetrievedServices|} \quad (8)$$

Where the list of relevant services to a given query is defined in the test collection. For this evaluation, we have considered only the services with a graded relevance value of 3 and 2.

2) Normalised Discounted Cumulative Gain: $NDCG_n$ uses a graded relevance scale of each retrieved service from the result set to evaluate the gain, or usefulness, of a service based on its position in the result list. This measure is particularly useful in Information Retrieval for evaluating ranking results. The $NDCG_n$ for n retrieved services is given by Equation 9.

$$NDCG_n = \frac{DCG_n}{IDCG_n} \quad (9)$$

Where DCG_n is the Discounted Cumulative Gain and $IDCG_n$ is the Ideal Discounted Cumulative Gain. The $IDCG_n$ is found by calculating the DCG_n of the first n returned services. The DCG_n is given by Equation 10

$$DCG_n = \sum_{i=1}^{n} \frac{2^{relevance(i)} - 1}{log_2(1+i)} \quad (10)$$

Where n is the number of services retrieved and $relevance(s)$ is the graded relevance of the service in the *i*th position in the ranked list. The $NDCG_n$ values for all queries can be averaged to obtain a measure of the average performance of a ranking algorithm. In our experiments, we consider only services with graded relevance values from 3 to 2 for this evaluation. $NDCG_n$ values vary from 0 to 1.

C. Results and Discussion

We evaluated our Service Discovery and Ranking approach by calculating the $Precision@n$ and $NDCG_n$. In this experiment, we have selected 8 queries - One query for each domain - (See Table II) from the test collection.

#	Domains	Query Name
1	Communication	Title Video Media
2	Economy	Shopping Mall Camera Price
3	Education	Researcher In Academia Address
4	Food	Grocery Store Food
5	Geography	Get Location Of City State
6	Medical	Hospital Investigating
7	Travel	City Country Hotel
8	Military	Government Missile Funding

TABLE II: Overview of the Queries used in our evaluation

The text description is retrieved from the query templates and used as the query string. We consider that the size of the services to be returned was set to 30.

Generally, the top most relevant services retrieved (i.e. the first 5 or 10) by a search engine are the main results that will be selected and used by the user. The $Precision@n$ values and $NDCG_n$ scores are obtained over all eight queries for the two probabilistic methods (i.e. CP: Conditional Probability, MA: Multidimensional Angle) based on the three probabilistic topic models (i.e. CTM, LDA, PLSA) and Text-Search.

The $Precision@5$ and $Precision@10$ values over eight queries are shown respectively in Table III and IV. The results show that the probabilistic method CP performs better than the MA for all the three probabilistic topic models. We remark that the CP based on CTM performs significantly than others methods. In fact, it gives a higher precision values (i.e. Average P@5 = 73% and Average P@10 = 68%) for all domains except Geography. We note also that the CP based on LDA performs better than MA based on LDA, CP/MA based PLSA and Text-Search. The methods based on PLSA and Text-Search were unable to find some of the relevant services that were not directly related to the queries. They give the lowest precision values.

The comparison of average $Precision@n$ (See Figure 1) shows that the probabilistic method CP performs better than the MA for all the probabilistic topic models. The results show that the CTM and LDA perform better than Text-Search and PLSA. The probabilistic methods based on CTM and LDA used the information captured in the latent factors to match web services based on the conditional probability of the user query. Text-Search and PLSA were unable to find some of the relevant web services that were not directly related to the user's queries through CTM and LDA. The low precision results obtained by probabilistic method based on PLSA are due to limited number of concepts used for training the model. In this context, web service descriptions are similar to short documents. Therefore, the method based on PLSA model is not able to converge to a high precision using these limited concepts.

In Information retrieval, $NDCG_N$ gives higher scores to systems which rank a search result list with higher relevance

Domain	Text-Search	PLSA		LDA		CTM	
		CP	MA	CP	MA	CP	MA
Communication	0.0	0.6	0.0	0.6	0.2	0.8	0.2
Economy	0.0	0.8	0.0	0.8	0.4	1.0	0.8
Education	0.0	0.8	0.4	0.6	0.2	0.4	0.0
Food	0.0	0.0	0.0	1.0	1.0	0.8	0.6
Geography	0.2	0.0	0.0	0.0	0.4	0.2	0.4
Medical	0.6	0.0	0.0	0.0	0.0	0.8	0.0
Travel	0.0	0.8	0.0	1.0	0.0	1.0	0.0
Military	0.0	0.0	0.0	0.6	0.6	0.8	0.6
Average	**0.1**	**0.38**	**0.05**	**0.57**	**0.35**	**0.73**	**0.33**

TABLE III: $Precision@5$ values for the eight queries.
(CP: Conditional Probability, MA: Multidimensional Angle)

Domain	Text-Search	PLSA		LDA		CTM	
		CP	MA	CP	MA	CP	MA
Communication	0.4	0.6	0.0	0.5	0.5	0.6	0.4
Economy	0.0	0.8	0.0	0.7	0.7	0.7	0.9
Education	0.0	0.6	0.6	0.8	0.3	0.5	0.0
Food	0.1	0.0	0.0	0.9	0.9	0.9	0.8
Geography	0.1	0.0	0.0	0.0	0.2	0.2	0.2
Medical	0.5	0.0	0.0	0.2	0.2	0.8	0.1
Travel	0.0	0.8	0.0	0.6	0.0	0.9	0.0
Military	0.0	0.1	0.0	0.5	0.5	0.8	0.6
Average	**0.14**	**0.36**	**0.08**	**0.53**	**0.41**	**0.68**	**0.38**

TABLE IV: $Precision@10$ values for the eight queries.
(CP: Conditional Probability, MA: Multidimensional Angle)

Domain	Text-Search	PLSA		LDA		CTM	
		CP	MA	CP	MA	CP	MA
Communication	0.0	0.83	0.0	0.29	0.39	0.36	0.39
Economy	0.0	0.75	0.0	0.74	0.55	0.74	0.72
Education	0.0	0.57	0.27	0.44	0.17	0.64	0.0
Food	0.0	0.0	0.0	0.54	0.65	0.43	0.52
Geography	0.52	0.0	0.0	0.0	0.41	0.52	0.41
Medical	0.5	0.0	0.0	0.0	0.0	0.74	0.0
Travel	0.0	0.53	0.0	0.45	0.0	0.45	0.0
Military	0.0	0.0	0.0	0.83	0.69	0.52	0.5
Average	**0.13**	**0.33**	**0.03**	**0.41**	**0.36**	**0.55**	**0.32**

TABLE V: $NDCG_5$ values for the eight queries.
(CP: Conditional Probability, MA: Multidimensional Angle)

Domain	Text-Search	PLSA		LDA		CTM	
		CP	MA	CP	MA	CP	MA
Communication	0.29	0.73	0.0	0.29	0.46	0.31	0.54
Economy	0.0	0.84	0.0	0.78	0.76	0.78	0.9
Education	0.0	0.49	0.4	0.68	0.33	0.58	0.0
Food	0.04	0.0	0.0	0.7	0.79	0.6	0.61
Geography	0.47	0.0	0.0	0.0	0.37	0.52	0.37
Medical	0.57	0.0	0.0	0.32	0.32	0.8	0.04
Travel	0.0	0.55	0.0	0.47	0.0	0.54	0.0
Military	0.0	0.1	0.0	0.64	0.61	0.48	0.52
Average	**0.17**	**0.34**	**0.05**	**0.48**	**0.45**	**0.58**	**0.37**

TABLE VI: $NDCG_{10}$ values for the eight queries.
(CP: Conditional Probability, MA: Multidimensional Angle)

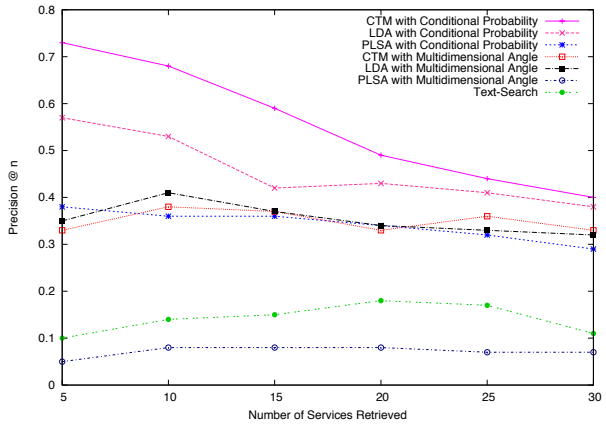

Fig. 1: Comparison of average Precision@n values over 8 queries.

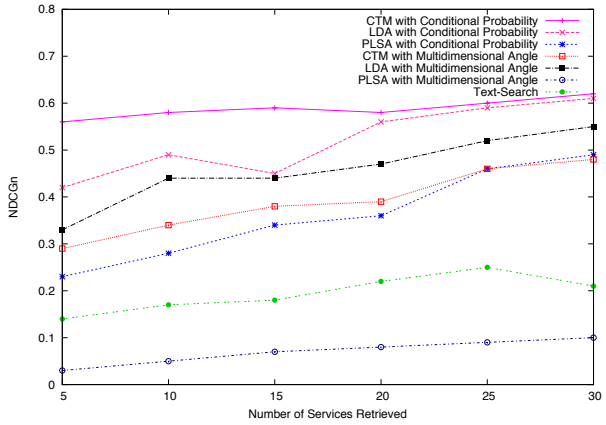

Fig. 2: Comparison of average $NDCG_n$ values over 8 queries.

first and penalizes systems which return services with low relevance. The $NDCG_5$ and $NDCG_{10}$ values over eight queries are shown respectively in Table III and IV. The $NDCG_n$ values for all queries can be averaged to obtain a measure of the average performance of a ranking algorithm. In our experiments, we consider services with graded relevance values from 3 to 2 for this evaluation. $NDCG_n$ values vary from 0 to 1. The results obtained for $NDCG_n$ show that the both CTM and LDA perform better than the other search methods. Thus, the probabilistic methods based on both CTM and LDA give a higher $NDCG_n$ than all other methods for any number of web services retrieved (See Figure 2). This reflects the accuracy of the ranking mechanism used by our method. Text-Search and PLSA methods have a low $NDCG_n$ because, as shown in the $Precision@n$ results, both methods are unable to find some of the highly relevant services.

As can be seen from Figure 1 and 2, CTM based on the Condional Probability performs significantly than others methods.

Finally, we evaluate the ranked lists obtained for both

ranking methods using the **Canberra distance**. In fact, the Canberra distance is used to measure the disarray for ranking lists, where rank differences in the top of the lists should be penalized more than those at the end of the lists [13]. Given two real-valued vectors $l, m \in \mathbb{R}^n$, their Canberra distance is defined as follows:

$$\text{Ca}(l, m) = \sum_{i=1}^{N} \frac{|l_i - m_i|}{|l_i| + |m_i|} \tag{11}$$

We consider only services with graded relevance values from 3 to 2 for this evaluation.

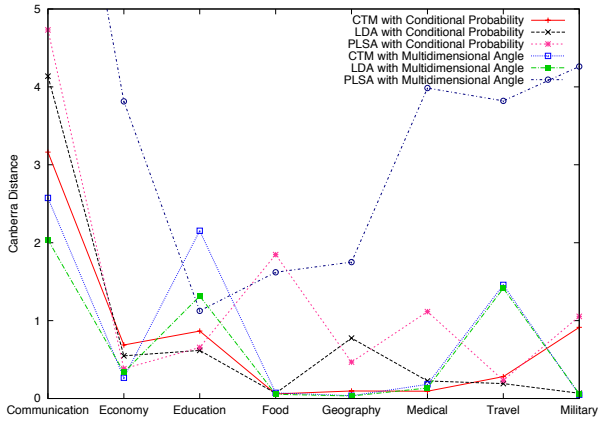

Fig. 3: Comparaison of $CanberraDistance$ values over 8 queries for the Ranking Methods.

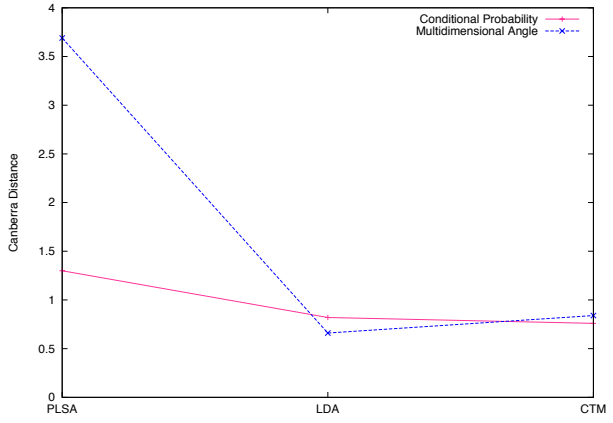

Fig. 4: Comparaison of average $CanberraDistance$ values for the Ranking Methods.

Figure 3 shows the Canberra Distance between the results abtained by both methods (CP and MA) based on the three probabilistic models and the relevant services for all eight queries. The comparaison of average CanberraDistance values for the Ranking Methods is shown in Figure 4.

The results show that the *CTM with Conditional Probability* method based on the Corelated Topic Model gives the lowest CanberraDistance values. This reflects the accuracy of the ranking mechanism used by our method.

IV. Related Work

In this section, we briefly discuss some of research works related to discovering Web services. In [1], the authors proposed an architecture for Web services filtering and clustering. The service filtering mechanism is based on user and application profiles that are described using OWL-S (Web Ontology Language for Services). The objectives of this matchmaking process are to save execution time and to improve the refinement of the stored data. Another similar approach [18] concentrates on Web service discovery with OWL-S and clustering technology. Nevertheless, the creation and maintenance of ontologies may be difficult and involve a huge amount of human effort [3], [14].

Generally, every web service associates with a WSDL document that contains the description of the service. A lot of research efforts have been devoted in utilizing WSDL documents [9], [3], [14], [15], [8], [16], [20]. Dong et al. [9] proposed the Web services search engine Woogle that is capable of providing Web services similarity search. However, their engine does not adequately consider data types, which usually reveal important information about the functionalities of Web services [12]. Liu and Wong [15] apply text mining techniques to extract features such as service content, context, host name, and service name, from Web service description files in order to cluster Web services. Elgazzar et al. [8] proposed a similar approach which clusters WSDL documents to improve the non-semantic web service discovery. They take the elements in WSDL documents as their feature, and cluster web services into functionality based clusters. The clustering results can be used to improve the quality of web service search results.

Some researchers use the proximity measures to calculate the similarity between services [18], [20]. Nayak et al. [18] proposed a method to improve the Web service discovery process using the Jaccard coefficient to calculate the similarity between Web services. Multidimensional Angle is an efficient measure of the proximity of two vectors. It is used in various clustering approaches [20]. This proximity measure applies cosine of the angle between two vectors. It reaches from the origin rather than the distance between the absolute position of the two points in vector space.

Ma et al. [16] proposed an approach similar to the previously discussed approaches [9], [1], [18] where the keywords are used first to retrieve Web services, and then to extract semantic concepts from the natural language descriptions in Web services. Ma et al. presented a service discovery mechanism called CPLSA which uses Probabilistic Latent Semantic

Analysis (PLSA) to extract latent factors from WSDL service descriptions after the search is narrowed down to a small cluster using a K-Means algorithm. The PLSA model represents a significant step towards probabilistic modelling of text, it is incomplete in that it provides no probabilistic model at the level of documents [4]. The Latent Dirichlet Allocation (LDA) [4] is an attempt to improve the PLSA by introducing a Dirichlet prior on document-topic distribution.

Cassar et al. [6], [7] investigated the use of probabilistic machine-learning techniques (PLSA and LDA) to extract latent factors from semantically enriched service descriptions. These latent factors provide a model which represents any type of service's descriptions in a vector form. In their approach, the authors assumed all service descriptions were written in the OWL-S. The results obtained from comparing the two methods (PLSA and LDA) showed that the LDA model provides a scalable and interoperable solution for automated service discovery in large service repositories. The LDA model assumes that the words of each document arise from a mixture of topics, each of which is a distribution over the vocabulary. A limitation of LDA is the inability to model topic correlation [5]. This limitation stems from the use of the Dirichlet distribution to model the variability among the topic proportions.

The Correlated Topic Model (CTM) has been developed to address the limitation of LDA [5]. In CTM, topic proportions exhibit correlation via the logistic normal distribution. One key difference between LDA and CTM is the independence assumption between topics in LDA, due to the Dirichlet prior on the distribution of topics (under a Dirichlet prior, the components of the distribution are independent whereas the logistic normal models correlation between the components through the covariance matrix of the normal distribution). However, in the CTM model, a topic may be consistent with the presence of other topics. In this paper, we exploit the advantages of CTM to propose an approach for web service discovery and ranking. In our approach, we utilized CTM to capture the semantics hidden behind the words in a query, and the descriptions of the services. Then, we extracted latent factors from web service descriptions. The latent factors can then be used to provide an efficient discovery and ranking mechanism for web services.

V. Conclusion

In this paper, we have used several probabilistic topic models (i.e. PLSA, LDA and CTM) to extract latent factors from web service descriptions. The learned latent factors are then used to provide an efficient Service Discovery and Ranking. We evaluated our Service Discovery and Ranking approach by calculating the precision ($Pecision@n$) and normalized discounted cumulative gain ($NDCG_n$). The comparison of Precision@n and $NDCG_n$ show that the CTM performs better than the other search methods (i.e. LDA, PLSA and Text-Search). This reflects the accuracy of the ranking mechanism used by our method. The probabilistic methods based on CTM used the information captured in the latent factors to match web services based on the conditional probability of the user query.

Future work will focus on developing a new probabilistic topic model which will able to tag web services automatically.

References

[1] Abramowicz, W., Haniewicz, K., Kaczmarek, M. and Zyskowski, D.: Architecture for Web services filtering and clustering. In ICIW'2007.

[2] Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services - Concepts, Architectures and Applications. Springer Verlag, Berlin Heidelberg, 2004.

[3] Atkinson, C., Bostan, P., Hummel O. and Stoll, D.: A Practical Approach to Web service Discovery and Retrieval. In ICWS'2007.

[4] Blei, D., Ng, A. Y. and Jordan, M. I.: Latent dirichlet allocation. J. Mach. Learn. Res., 3:993-1022, 2003.

[5] Blei, D., and Lafferty, John D.: A Correlated Topic model of Science, In AAS 2007. pp. 17-35.

[6] Cassar, G., Barnaghi, P. and Moessner, K.: Probabilistic methods for service clustering. In Proceeding of the 4th International Workshop on Semantic Web Service Matchmaking and Resource Retrieval, Organised in conjonction the ISWC'2010.

[7] Cassar, G.; Barnaghi, P.; Moessner, K.: A Probabilistic Latent Factor approach to service ranking. In ICCP'2011, pp.103-109.

[8] Elgazzar, K., Hassan A., Martin, P.: Clustering WSDL Documents to Bootstrap the Discovery of Web Services. In ICWS'2010, pp. 147-154.

[9] Dong, X., Halevy, A., Madhavan, J., Nemes, E., Zhang, J.: Similarity Search for Web Services. In VLDB Conference, Toronto, Canada, pp. 372-383, 2004.

[10] Hess, A. and Kushmerick, N.: Learning to Attach Semantic Metadata to Web services. In ISWC'2003, Sanibel Island, Florida, USA, 2003

[11] Hofmann, T.: Probabilistic Latent Semantic Analysis. In UAI(1999), pp. 289-296.

[12] Kokash, N.: A Comparison of Web Service Interface Similarity Measures. Frontiers in Artificial Intelligence and Applications, Vol. 142, pp.220-231, 2006.

[13] Jurman, G., Riccadonna, S., Visintainer, R., Furlanello, C., Canberra Distance on Ranked Lists. In Proceedings of Advances in Ranking NIPS'2009 Workshop, 22-27.

[14] Lausen, H. and Haselwanter, T.: Finding Web services. In European Semantic Technology Conference, Vienna, Austria,2007

[15] Liu, Wei., Wong, W.: Web service clustering using text mining techniques. In IJAOSE'2009, Vol. 3, No. 1, pp. 6-26.

[16] Ma, J., Zhang, Y. and He, .J.: Efficiently finding web services using a clustering semantic approach. In CSSSIA'08, pp 1-8. ACM, New York, NY, USA.

[17] Maximilien, E.M., Singh, M.P.: Toward Autonomic Web Services Trust and Selection. In ICSOC'2004, pp. 212-221

[18] Nayak, R. and Lee, B.: Web service Discovery with Additional Semantics and Clustering. In IEEE/WIC/ACM 2007

[19] Porter, M. F.: An Algorithm for Suffix Stripping, In: Program 1980, Vol. 14, No. 3, pp. 130-137.

[20] Platzer, C., Rosenberg F. and Dustdar, S.: Web service clustering using multidimentional angles as proximity measures. ACM Trans. Internet Technol. 9(3), pp. 1-26 (2009).

[21] Rajendran, T., Balasubramanie, P.: An Optimal Broker-Based Architecture for Web Service Discovery with QoS Characteristics. In IJWSP'2010, Vol. 5, No. 1.

[22] Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA (1989).

[23] Sivashanmugam, K., Verma, A.P and Miller, J.A.: Adding Semantics to Web services Standards. In ICWS'2003, pp: 395-401.

[24] Steyvers, M. and Griffiths, T.: Probabilistic topic models. In Latent Semantic Analysis: A Road to Meaning, T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, Eds. Laurence Erlbaum, 2007.

[25] W3C (2004). Web services architecture. Technical report, W3C Working Group Note 11 February 2004.

[26] Xu, Z., Martin, P., Powley, W. and Zulkernine, F.: Reputation Enhanced QoS-based Web services Discovery. In ICWS'2007.

[27] Yu, Q.: Place Semantics into Context: Service Community Discovery from the WSDL Corpus. In ICSOC 2011, LNCS 7084, pp. 188-203.