# Service Selection in Business Service Ecosystem

Sujoy Basu, Sven Graupner, Kivanc Ozonat, Sharad Singhal, Donald Young
Hewlett-Packard Laboratories, 1501 Page Mill Road
Palo Alto, CA 94304, USA
{sujoy.basu, sven.graupner, kivanc.ozonat, sharad.singhal}@hp.com

**Abstract.** A world-wide community of service providers has a presence on the web, and people seeking services typically go to the web as an initial place to search for them. Service selection is comprised of two steps: finding service candidates using search engines and selecting those which meet desired service properties best. Within the context of Web Services, the service selection problem has been solved through common description frameworks that make use of ontologies and service registries. However, the majority of service providers on the web does not use such frameworks and rather make service descriptions available on their web sites that provide human targeted content.

This paper addresses the service selection problem under the assumption that a common service description framework *does not* exist, and services have to be selected using the more unstructured information available on the web.

The approach described in this paper has the following steps. Search engines are employed to find service candidates from dense requirement formulations extracted from user input. Text classification techniques are used to identify services and service properties from web content retrieved from search links. Service candidates are then ranked based on how well they support desired properties. Initial experiments have been conducted to validate the approach.

**Keywords:** service, selection, unstructured data, service discovery, service matchmaking, text analysis, service ontology, service description framework.

## 1 Introduction

The service selection problem is how to *find and select services* offered by service providers that best meet the requirements described by a service seeker. On one side, the service seeker provides a description of requirements or desired service properties. On the other side, service providers describe their capabilities. In general, this is a two-step process. In the first step, a set of service provider candidates, which generally meet requirements (not necessarily all requirements), are identified. Requirements are described as a set of desired service properties. In the second step, the "best" service provider is selected from the candidate set based on a "best match" between the properties exposed by the service provider and those requested by the service seeker. Typically, in order to solve the service selection problem, a common base must exist or must be established between the descriptions of desired service properties on one side and descriptions of service capabilities on the other. This common base can be strong and formal, as it has traditionally been defined by *service description frameworks* with pre-defined syntactic and semantic constructs for expressing service properties and query and matchmaking capabilities built on top of

it. A service registry is an example of a system which defines a service description framework.

While the service selection problem has been explored and has largely been solved under the assumption that a formal service description framework exists, this is not true of the majority of service offerings, which are described in natural language in marketing collateral, web content or advertisements. The same applies at the service seeker's side: most documents that describe what a person is looking for are expressed in (largely) unstructured formats. And this is particularly true for the web as the most significant medium through which service offerings are promoted and advertised as well as sought and found. Consequently, the service selection problem must be considered for an unstructured environment such as the web.

We see a need to enable service technology to better deal with unstructured data. It will be a central challenge for the comings years. In this paper, we present an approach to the service selection problem using unstructured data for service properties. We address the problem under the assumption that no common description framework exists. In the proposed approach, first, search engine is employed to find service candidates from dense requirement formulations extracted from user input. Next, text classification techniques are used to identify services and service properties from web content retrieved from returned search links. Service candidates are then ranked based on how well they supported desired properties. We have evaluated the approach through conducting experiments on a sampling of real requirements documents of an internal marketing department of a large corporation. The experiments show encouraging results.

The paper is structured as follows. First, we discuss related work in Section 2. Then we discuss assumptions underlying the work in Section 3. Next, we present the problem statement in Section 4. Section 5 gives an overview of the approach of the proposed solution. Details about the techniques and experiments are presented in Section 6. Finally, we discuss future work and conclude the paper in Section 7.

## 2 Related Work

Service technology from the Web 1.0 era mainly relied on established service description frameworks in order to find and select services. Prominent representatives are service registries, which are a fundamental building block of Service-Oriented Architecture (SOA) [1] today. An example is the Universal Description, Discovery and Integration registry (UDDI) [2]. Service registries have not only been defined for technical (web-) services, but also for businesses services (e.g. OASIS ebXML Registry [3]). Registries define syntax and semantics in which service descriptions must be expressed. A common pattern is the use of attribute-value pair sets. Query techniques in registries rely on either a priori known or discoverable attribute sets for finding and selecting services. Query languages are based on Boolean expressions or first-order logic upon property sets. They lead consistently to defined, repeatable search results in a given search space. More advanced techniques use extendable service vocabularies and ontologies for expressing and matching service properties. Service technology of this kind has been targeted towards machines.

Approaches for discovering Web services, i.e., those that are described using WSDL [19] interfaces, are extensively investigated in the literature (e.g., [20], [21], [22]). There exist also Web services repositories and portals where such services are indexed and can be explored and discovered (e.g., XMethods [23] and Seekda [24]). in the focus of this paper is rather on general *services over the Web*, most of which are described in natural language. Indeed, more recent service technology is, in contrast, more targeted towards both people as service seekers and providers. The common base here is a common natural language with all its ambiguity making it hard for machines to identify good service selections.

The Web plays a central role today for bringing service seekers and service providers together, as individuals or as businesses. An indicator for this is the significant revenue search engines harvest from searches on the web. And yet, support for service selection in search engines and on the web in general remains rather limited. More advanced techniques are needed which can deal with the ambiguity of the web.

Ideally, the web should "understand" human natural language, and there are efforts towards this [4, 5]. Similarly, efforts are underway in the Semantic Web [6] to structure web content. However, neither has made the necessary progress. Rather, text analysis and categorization techniques appear more practical and promising and have widely been explored on general web content [7, 8, 9], but not specifically for the service selection problem. This paper specifically considers the service selection problem and outlines an approach that uses text analysis techniques for finding and selecting services offered over the web.

## 3 Assumptions

The paper addresses the service selection problem from the perspective of a seeker of services in a corporate or private environment. It is assumed that someone (a person) in the service seeker's role informally knows about a task to be given to a service provider and that a textual description of desired service properties exists. One form of expressing these properties is in form of documents such as a statement-of-work (SOW), a request-for-proposal (RFP) or a request-for-quote (RFQ) document that corporations use to procure external services. We use a sample set of these documents for identifying desired service properties.

It is furthermore assumed that service providers have presences on the web (web sites) where they informally describe and advertise their capabilities and that those web pages can be found through search engines. While it is assumed that informal descriptions about service providers can be obtained from the web, it is *not* assumed that the actual business interaction also is mediated over the web. For example, a legal counseling agency may be found on the web, but actual counseling then occurs in person. We explore service providers' web content and classify its properties.

Another assumption is that a common language (English) is used to describe sought and offered service properties; and that the same words or phrases are used for same concepts. WordNet [10] provides a rich set of English language words, terms, phrases and defining semantics.

## 4 Problem Statement

Based on these assumptions, the paper addresses the following problems:

1. Can sought service properties (requirements) be gathered informally from a business person in a service seeker role and represented in a condensed descriptive vector of meaningful terms?
2. Can these terms then be used in search engines to find service provider candidates? This includes that service providers must be distinguished from other content returned from search.
3. Can service properties be extracted and classified from service providers' web content (their web sites)?
4. Can properties from service seeker's requirements and service provider's capabilities be correlated such that providers can be ranked based on how well they support requirement properties?

The first two problems relate to "how to find service candidates"; the last two address the matchmaking aspect of the service selection problem.

## 5 Approach

People seeking services typically go to the web and search for them and then select what they like. This process is inherently manual since most data based on which those decisions are made exists as unstructured data, e.g. content of web pages. In this paper, an approach has been developed which allows automating the process.

A search engine is employed to find service candidates. It is fed with key words which are extracted from informal user input using a text classification technique. The result page returned from the search engine contains links to potential service providers. Those links are followed programmatically to fetch content which may or may not represent a service provider. A machine learning technique is employed to automatically make a judgment whether content represents a service provider or not. Furthermore, if content was classified as representing a service provider, more pages are fetched from this site to accumulate enough material to again employ a text analysis technique to determine how well desired service properties are supported. The produced result is factored into a final score of all identified service providers to rank order them.

The approach consists of four steps, each addressing one of the above problems:

The first step aims at condensing the information business people use informally when communicating service requirements. The goal is to identify the essential terms from documents which describe required service properties. Forms and questionnaires are familiar to a business audience and a good way to produce dense information for procuring external services. An example illustrates this approach. For a marketing campaign at a larger customer event, a service provider may be sought which can *"print quantity 2,000 colored 8.5x11 z-fold brochures, 100 lb. gloss within 10 days with maximum budget of $1,000"*. This string represents a condensed form of a statement of work document and the essence of desired service properties. Latent Semantic Indexing (LSI) [11] is used to extract these terms from a representative set

of documents. This step results in a descriptive vector of meaningful words representing the essence of required service properties.

The second step is to use these meaningful words in search engines and to obtain a pool of links to potential service candidates. Since links returned from search may refer to any content, which may or may not be service providers, links must be followed and content obtained from links in order to determine whether or not content represents a service provider. If content could be successfully probed and classified as representing a service provider, the service provider is added the pool of potential service provider candidates. For example, when the string above is typed into the Google search engine, it returns 11 sponsored links of print service providers (in the US) and a number of random entries, which are more or less related to printing brochures, but may or may not be service providers. Typically, in Google, it is sufficient to consider content returned with the first page. Other search engines such as Yahoo! and Microsoft may return different content.

Further sources of information about service providers can be involved such as established service catalogs such as Hoovers [12], Dun and Bradstreet [13] or ThomasNet [14] in order to obtain a larger candidate pool of service providers. These service catalogs have been collecting structured information about businesses in the US and worldwide and make this information available over the web.

However, the answer from search engines or service catalogs can only be used as a starting point to further explore whether or not a returned link represents a service provider. The second problem hence addresses whether or not a site behind a link can be identified as a service provider. The approach here is to look for FORM pages encouraging users to engage with the service. This step results in a pool of potential service provider candidates.

Furthermore, in preparation of comparison, service properties must be identified for candidates from their web content. The approach here relies on meta-tags and content of online service engagement forms. Thus, this step also provides a set of service properties identified for each service.

The service candidates are ready for comparison after their service properties have been extracted. Furthermore, they must be correlated with service properties from condensed requirements derived in the first step. Since LSI has been used in the first step, we repeat the process with the service provider pages to generate their condensed set of properties as a vector of terms. Then we use cosine similarity as the metric to rank the list of service provider candidates that support the desired service properties best.

While this process does not guarantee that service providers found as a result will support the requirements and that the top-ranked service candidate is indeed the best to engage, it mirrors the process a person would follow when asked to find a product or a service on the web. The advantage here is that we can automate the manual steps to potentially explore a much larger set of service provider candidates than would be feasible manually. In the next section, we describe a number of initial experiments that have been conducted to highlight the potential of this approach.

## 6 Techniques and Experiments

This section describes experiments that have been conducted for the four steps. We consider these experiments as an initial set of experiments and are currently scaling up our experiments to larger data sets. We do not claim that these experiments represent the best choices of techniques that could have been made, nor do we claim to evaluate and compare different techniques in this paper. Our goal is to demonstrate a feasible implementation of the four steps of the service selection problem in an unstructured web environment. Others researchers such as Forman [15] have done extensive empirical study of feature selection for text classification.

### 5.1 Extracting Significant Words for Service Requirements

*Input.* We use 24 PDF documents from an internal marketing department, which are primarily request for quotes (RFQ) for various printing projects undertaken by a large enterprise typically through an entity known as the print concierge. The RFQs are normally sent to approved vendors.

*Objective.* We seek an automated method, which is not based on domain-specific knowledge, which can identify the list of terms representing the essence of required service properties and can handle synonymy and redundancy inherent in natural language documents. We do not expect the number of RFQ documents to grow as fast as documents on the Web. We are currently using Latent Semantic Indexing (LSI) to identify the list of terms in this environment.

*Technique.* The LSI method in [11] uses Singular Value Decomposition (SVD), which belongs to the class of matrix decomposition techniques in linear algebra. To begin, we create a matrix where the rows represent the terms and its columns represent the documents. An element of the matrix represents the frequency of a term in a document. SVD expresses this matrix X as the product of 3 matrices, T, S and $D^t$, where S is a diagonal matrix of singular values ordered in descending order, and T and D are the matrices of eigenvectors of the square symmetric matrices $XX^t$ and $X^tX$ respectively. Furthermore, the square of the singular values are the eigenvalues for both $XX^t$ and $X^tX$. The dimension of X is *t* (number of terms) by *d* (number of documents), while that of T is *t* by *m*, where *m* is the rank of X and is at most the minimum of *t* and *d*. S is an *m* by *m* matrix. Intuitively, SVD transforms the documents (columns of X) and the terms (rows of X) into a common space referred to as the *factor space*. The singular values in S are weights that are applied to scale the orthogonal, unit-length columns vectors of T and D and determine where the corresponding term or document is placed in the factor space. Similarity between documents or the likelihood of finding a term in a document can be estimated by computing distances between the coordinates of the corresponding terms and documents in this factor space.

The eigenvectors corresponding to the highest eigenvalues represent principal components that capture the most important characteristics of the data. The contributions keep diminishing for descending eigenvalues. By dropping some of the lower eigenvalues and corresponding eigenvectors, we lose some information, but can reduce the dimensionality of the data. This is useful when the number of documents is very large. We can retain the *k* highest eigenvalues, and the corresponding

eigenvectors in the T and D matrices. The product $T_{txk}\ S_{\underline{kxk}}\ D^t_{kxd}$ gives the unique matrix of rank *k* closest to X based on a least-square distance metric. LSI is the process of using this matrix of lower rank to answer similarity queries such as which terms are strongly related and given query terms, and what are the related documents. LSI has been shown to return query matches with higher precision when synonyms or multiple word senses would have prevented syntactic matching.

*Experiment.* We use LSI on the term by document matrix obtained from our document set. The terms were single words, bigrams and trigrams. We filtered out stopwords and the long tail of words that occurred only once. We reduced the rank of the matrix to *k* chosen such that 99% of the sum of squares of the singular values, which is the sum of eigenvalues, is retained. Next, we take the product $T_{txk}\ S_{kxk}$ which consists of the eigenvectors weighted by their singular values. In [17], Deerwester et.al. show that the comparison of two terms can be done in the factor space by taking inner product of corresponding rows in $T_{txk}\ S_{kxk}$. However, we want to extract the important terms. So we take the maximum absolute value in each row as the importance of that term, and sort based on this metric to order the terms by their descending importance. Given a threshold, our algorithm outputs all terms for which the metric, normalized to its maximum value, exceeds this threshold. When a new requirement document arrives in our system, LSI allows us to map it into a vector in the factor space by a simple matrix multiplication, and extract its important terms using this threshold technique.

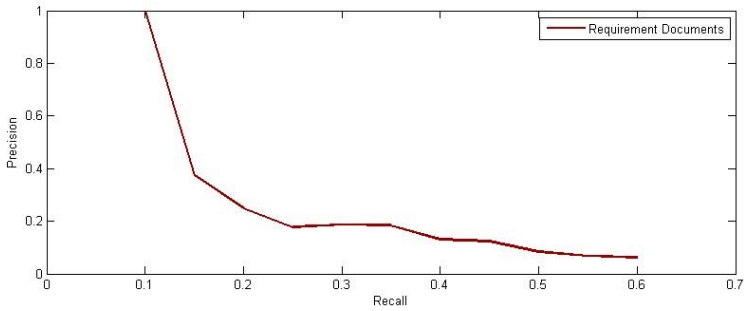


**Figure 1:** Variation of Precision with Recall as threshold is varied.

Since we need the ground truth to determine how well this approach works, we asked a human participant in our experiment to read the documents and identify important terms. To eliminate bias, this person had no a priori knowledge of the terms selected by our automated approach based on SVD and limited himself to manual analysis of the documents. He created a list of the top 20 important terms, henceforth referred to as the *ground truth* for this experiment. We started with the top 20 terms from the sorted list created by SVD, and progressively relaxed the threshold. At each stage, we calculated *precision* as the fraction of SVD's output that is present in the ground truth. We also calculated *recall* as the fraction of the ground truth that is present in the output of SVD. Our recall suffers due to our strategy of including all 2 and 3 letter words as stopwords. Thus the word 'ink', which is included in the ground truth, is absent from our term by document matrix. The same is true for terms such as

"80# gloss" since we did not process any token such as "80#" that does not have at least one alpha character. Figure 1 shows the variation of precision with recall as the threshold is varied. Precision drops from 1 to 0.25 when we relax our threshold progressively from 1 to 0.28. During this period, we observed only a small increase in recall to 0.2. The threshold was then relaxed progressively to 0.07. During this period, recall increased to 0.6, while precision dropped to 0.06.

### 5.2 Finding Service Candidates

We use a search engine (in particular, AltaVista) to find a pool of service providers through which the customer can engage in a business interaction over the web to meet its service needs. We use phrases such as "telemarketing service", "printing service" or "copyright litigation service" as input to retrieve the service provider pages.

In order to populate our data set (of service providers), we randomly selected words under the Wikipedia descriptions of the three services (telemarketing, printing and copyright litigation), and used them as input phrases to the AltaVista search engine. The choice is based on the fact that we were successful in accessing it programmatically. Recently, we have been made aware of BOSS [18], which is Yahoo!'s open search web services platform. We intend to redo our experiments using this API. Our understanding is that Google has stopped issuing new keys for its SOAP API that would have allowed programmatic search. Instead the new AJAX API must be used. It is geared towards browsers that need to tweak the display of search results, and will most likely not be suitable for our needs.

The pages retrieved by the search engine needed to be filtered as we seek only web forms and only web forms that contain the properties and attributes of the offered services to initiate a business engagement. Thus, for each retrieved web page by AltaVista, we retrieved the HTML source of the page to filter the non-form pages (or non-form sections of the form pages). We used the standard HTML tags that denote HTML form tags in order to filter out the non-form pages or non-form sections.

### 5.3 Identify Service Properties of Service Candidates

Once the pool of service providers (or, alternatively, the pool of forms, since there is a one-to-one mapping between forms and service providers) is determined by the methods of section 5.2, we seek to find the properties of each service type represented in the pool. For the experiments we conducted, there are three service types: telemarketing, printing and copyright litigation.

Throughout the remainder of the discussion, we denote each service type by $m$, each service providers (or form) by $n$, and each word used in the forms by $w$. We denote the number of service types, number of service providers, and the number of distinct words used in the forms by M, N and W, respectively. We note that the parameters N and W are pre-determined, since the pool of forms is already discovered in section 5.2. We assume that the service types $m$ (and consequently the number of service types M) are known.

We use statistical learning techniques to identify properties of services based on the pool of service providers (or forms) retrieved by the methods discussed in section

5.2. We employ both supervised and unsupervised learning techniques. In supervised learning, the service type of each service provider *n* is assumed to be known, while in unsupervised learning this information is missing.

Data representation: We model each service provider (or form) *n* by a sequence of W bits (sequence of 0's and 1's), where each bit represents whether a word is present in the form. If the bit is 0, the word is absent, and if it is 1, the word is present.

*Unsupervised learning:* Clustering is an unsupervised learning technique, where objects that are close under some distortion criterion are clustered in the same group.

*Supervised learning:* When the cluster labels of the objects are already available (i.e., the service type of each service provider is known), one can use a supervised learning technique. We model the data as a mixture of M (where M=3) W-dimensional Gaussians, and estimated the parameters each Gaussian using the sample averages of the forms in that cluster. We use the k-means clustering algorithm for supervised classification with the squared-error distortion measure to cluster forms into M=3 groups. Each object to be clustered is a vector of 0's and 1's of length W, representing a form.

In total, 600 pages have been classified for this experiment for the three service categories with ~200 for each category. 122 of those pages have been form pages.

*Results:* We did not obtain encouraging results through unsupervised learning; however, supervised learning led to keywords that describe each service reasonably well. These words were selected based upon the probability of 50% or greater that the word will be found in all documents in the cluster. Keywords are shown in Table 1.

| **Service category** | **Significant keywords** |
|---|---|
| Telemarketing | inbound, outbound, call, center, **telemarketing**, outsourcing, marketing, company, phone, address, zip, list. |
| Printing | business, card, **printing**, format, color, folder, sample, presentation, text, quote. |
| Litigation | intellectual, property, dispute, client, copyright, **litigation**, client, business, infringement, attorney, law, trial, website, competitor. |

**Table 1:** Keywords describing service properties for three service categories.

### 5.4 Rank Service Candidates

*Input.* In this step, we start with web pages of services identified from the web similar to step 5.2.

*Objective.* We seek an automated method, not based on domain-specific knowledge, which can identify the subset of the input services that match the required service properties of one of the requirement documents described in section 5.1. Since multiple matches are expected, a rank-ordered list of the services should be produced.

*Technique.* Singular Value Decomposition (SVD) was introduced in Section 5.1. We use SVD to index the service descriptions and treat one of the requirement documents as a term vector with which a query is performed.

*Experiment.* We use SVD on the term by document matrix obtained from the service web pages treated as documents. The HTML tags were removed. Again, we use single words, bigrams and trigrams as the terms. We reduced the rank of the matrix to *k* chosen such that 99% of the sum of squares of the singular values, which is the sum of eigenvalues, is retained. The term vector for the query is transformed into the factor space obtained by SVD. This involves multiplying the transpose of the term vector by the SVD term matrix $T_{txk}$. The coordinates of the transform may be compared in this space to the other documents representing the services by accessing individual rows of the matrix product $D_{dxk}\, S_{kxk}$. For each row of this matrix product, we compute the inner product with the transform of the query term vector. Then we compensate for variable document sizes by normalizing the result by the product of the Euclidean length of the two vectors. The result is their *cosine similarity*, a standard measure for quantifying similarity of two documents in a vector space.

For our data, the human participant again established the ground truth without a priori knowledge of how SVD ordered the services. He did a manual evaluation of the requirement document used as query term vector. Based on his qualitative judgment, he provided us with a rank ordering of the services documents (HTML pages) in terms of how well they will meet the requirements specified in the query document.

We ranked the services based on cosine similarity of the service documents to the requirement document used for query. The correlation of two rank orders can be quantified by the Spearman rank correlation coefficient, which ranges between +1 when both rank orders are identical to -1 when one rank order is 1, 2,3,…,n and the other one is n,n-1,n-2,…,1. The results are presented in Table 2. In the absence of shared ranks, the Spearman coefficient is simply $1 - (6\sum d^2 / n\,(n^2 - 1))$. From this table, $\sum d^2$ is 106 and n = 17. So the Spearman coefficient is 0.87 for our experiment, indicating very high correlation.

| **Anonymized Service Names** | Cosine Similarity | Cosine Rank | Manual Rank | d-squared |
|---|---|---|---|---|
| pitn-prt | 0.4848 | 7 | 1 | 36 |
| mpit-prt | 0.7469 | 2 | 2 | 0 |
| pitl-prt | 0.6404 | 3 | 3 | 0 |
| ppit-prt | 0.8149 | 1 | 4 | 9 |
| pitu-prt | 0.5275 | 5 | 5 | 0 |
| mkid-mlg | 0.51 | 6 | 6 | 0 |
| jsad-mlg | 0.5665 | 4 | 7 | 9 |
| pitn-mlg | 0.4564 | 9 | 8 | 1 |
| byro-tlm | 0.2363 | 12 | 9 | 9 |
| bbec-tlm | 0.182 | 13 | 10 | 9 |
| gtqo-tlm | 0.3634 | 10 | 11 | 1 |
| vnes-tlm | 0.4821 | 8 | 12 | 16 |
| spra-tlm | 0.3045 | 11 | 13 | 4 |
| il_o-lwr | 0.0505 | 17 | 14 | 9 |
| lwno-lwr | 0.1524 | 14 | 15 | 1 |
| cbrr-lwr | 0.134 | 15 | 16 | 1 |
| cmue-lwr | 0.0709 | 16 | 17 | 1 |

**Table 2:** Comparison of service ranks obtained manually and through SVD for three service categories: printing (prt), telemarketing (tlm) and legal services (lwr).

In Table 2 we compare the service ranks obtained manually and through SVD for three service categories: printing, telemarketing and legal services.

This is further analyzed in Figure 2 where we plot the precision versus recall, assuming that the top 10 services in the manual rank column are relevant to the needs expressed in the requirement document. This is likely to be over-optimistic since an enterprise is likely to take the list of 17 services and put them through a qualification process and create a shortlist of vendors to whom requirements will be sent in future as part of a request for quote. Ideally, that shortlist should be the ground truth for calculating precision and recall. We assume that the top 10 services in the manual rank column will be fairly representative of the shortlist that an enterprise may generate if provided with these 17 services in response to requirement documents that are related to marketing campaigns. We observe from this graph that we can obtain a recall of 0.7 without losing any precision.

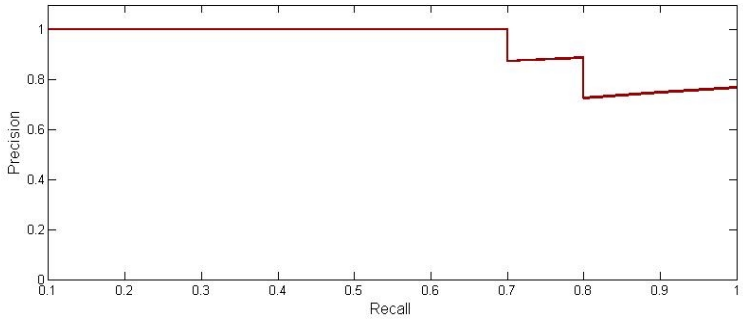


**Figure 2:** Precision versus recall for ground truth consisting of top 10 services from manual ranking.

### 5.5 Discussion

We consider the chosen techniques and experiments as an initial set which led to an initial set of answers. There is no doubt that the service selection problem remains difficult in an environment with largely unstructured data.

From the experiments in the first part, we can conclude that SVD provides a reasonable method for extracting important terms. As we have stated at the beginning of Section 5, our current goal was not to find the best method for each part, but to validate the approach. Since some of the terms provided as ground truth are 3-letter words, we conclude our policy of eliminating all 3-letter words as stopwords should be replaced by a more specific one based on lexical databases like WordNet.

For the second part, the identification of service providers from content returned from search using form tags on the web site, we can conclude that search engines such as Google, Yahoo and AltaVista retrieve not only the form pages that typically include attributes/capabilities for the services, but all pages that are related to the service being searched. Using the standard HTML tags that denote forms in order to filter out the non-form pages or non-form sections is a simple, but powerful technique in filtering out irrelevant text.

For the third part, the identification of service properties from form tags, we can conclude that we do not have a quantitative measure of accuracy. We note that the words in Table 2 reflect the major attributes of the services to which they correspond. For instance, for telemarketing services, the key attributes include the call volumes as well as numbers of inbound and outbound calls per minute. These are extracted by our algorithm and included as inbound, outbound and call. The keywords extracted for printing and copyright litigation also include major attributes of those services. We should point out, however, that not all keywords for each service have been extracted. For instance, for printing services, keywords that describe products to be printed (e.g., brochures, posters, etc.) are not extracted.

For the fourth part, the ranking of service provider candidates against a set of requirements using the SVD method, we compared against manual ranking and obtained high positive correlation. We can conclude that SVD was an effective means of ranking the services in the small set used for this experiment.

## 7 Conclusion and Future Work

The paper has presented an approach to the service selection problem using unstructured data for service properties. The problem has been addressed under the assumption that no common description framework exists. Four essential steps have been identified to address the problem. A search engine was employed to find service candidates from dense requirement formulations extracted from user input. A text classification technique was used to identify services and service properties from web content retrieved from returned search links. Service candidates were then ranked based on how well they supported desired properties using a Single Value Decomposition method.

An initial set of experiments has been conducted using a sampling of real requirements documents of an internal marketing department of a large corporation to procure external services for marketing campaigns. Descriptions of real service providers found on the web were used to extract service capabilities. Initial experiments showed encouraging results, but also exposed shortcomings that need to be addressed by further research.

A number of challenges have been uncovered by this research. We will address some of them in future work. One direction is to improve the techniques used for the service selection problem. Another direction aims at expanding from selecting services to also engaging them.

For the first direction, we will explore and develop stronger statistical learning techniques to improve accuracy for identifying service providers and extracting their properties from web content. We are also looking at incorporating structured data sources available on the web such as service catalogs or clustered information sources to improve results. The size of data sets (numbers of documents and services) we have considered was relatively small (10's of documents, 100's of web page from services). We need to explore how well the techniques scale over larger numbers of documents and services.

The second direction is to also actually engage services if they provide means of online engagement. It will open a whole new set of problems that needs to be

explored such as what different types of engagements can exist, how business relationships and service contracts are represented, and how engagement stages, steps and processes are described and executed, again under the assumption that no common formally defined framework can be assumed such as, for instance, RosettaNet's Partner Interface Processes (PIP) or Trading Partner Implementation Requirements (TPIR) [16] definitions or conventional Business Processes.

## Acknowledgement

We would like to acknowledge the advice of Hamid Motahari for bringing the paper into its final shape.

## References

1. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service Oriented Architecture Best Practices. Prentice Hall (2005).
2. Universal Description, Discovery and Integration (UDDI). http://uddi.xml.org.
3. OASIS ebXML Registry. http://www.oasis-open.org/committees/regrep.
4. Lapata, M., Keller, F.: Web-based Models for Natural Language Processing. ACM Transactions of Speech and Language Processing, Vol. 2, No. 1., Pages: 1-30 (2005). http://homepages.inf.ed.ac.uk/mlap/Papers/tslp05.pdf
5. Powerset. Discover Facts. Unlock Meaning. Scan Summaries. http://www.powerset.com.
6. W3C Semantic Web. http://www.w3.org/2001/sw.
7. Soumen Chakrabarti: Mining the Web: Analysis of Hypertext and Semi Structured Data, Morgan Kaufmann (2002).
8. Bing Liu: Web Data Mining: Exploring Hyperlinks, Contents and Usage Data, Springer Verlag, (2007).
9. Olfa Nasraoui, Osmar Zaiane, Myra Spiliopoulou, Bamshad Mobasher, Philip Yu, Brij Masand, Eds.: Advances in Web Mining and Web Usage Analysis 2005 - revised papers from 7th Workshop on Knowledge Discovery on the Web, Springer Lecture Notes in Artificial Intelligence, LNAI 4198, (2006).
10. WordNet. Cognitive Science Laboratory. Princeton University. http://wordnet.princeton.edu
11. Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S. (1988), "Using latent semantic analysis to improve information retrieval." In *Proceedings of CHI'88: Conference on Human Factors in Computing*, New York: ACM, 281-285.
12. Hoovers. Online business registry. http://www.hoovers.com
13. Dun and Bradstreet. Provider of international and US business credit information and credit reports. http://www.dnb.com
14. ThomasNet. Provider of information about Industrial Manufacturers. http://www.thomasnet.com
15. Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification, Journal of Machine Learning Research 3, pages 1289-1305 (2003).
16. RosettaNet: Trading Partner Implementation Requirements (TPIR) Partner Interface Process (PIP) Maintenance Service. http://www.rosettanet.org/shop/store/catalogs/publications.html

17. Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A., "Indexing by latent semantic analysis." Journal of the Society for Information Science, 41(6), 391-407, 1990
18. Yahoo! Search BOSS (Build your Own Search Service) http://developer.yahoo.com/search/boss/
19. W3C, Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl
20. Dong, X., Halevy, A., Madhavan, J., Nemes, E., and Zhang, J. 2004. Similarity search for web services. In *Proceedings of* VLDB 2004, pp. 372-383.
21. Ma, J., Zhang, Y., and He, J. 2008. Efficiently finding web services using a clustering semantic approach. In *Proceedings of WWW 2008.*
22. Wang, Y. and Stroulia, E. 2003. Flexible Interface Matching for Web-Service Discovery. In *Proceedings of the Fourth international Conference on Web information Systems Engineering,* 2003
23. XMethods. http://www.xmethods.com
24. Seekda, http://seekda.com/