HAL
open science

Applying a family of IR models to text description-based service retrieval

Isaac Caicedo-Castro, Marie-Christine Fauvet, Ahmed Lbath, Helga Duarte-Amaya

To cite this version:

Isaac Caicedo-Castro, Marie-Christine Fauvet, Ahmed Lbath, Helga Duarte-Amaya. Applying a family of IR models to text description-based service retrieval. CORIA, 2015, PARIS, France. hal-01888342

HAL Id: hal-01888342

https://hal.science/hal-01888342v1

Submitted on 5 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Toward the Highest Effectiveness in Text Description-based Service Retrieval

Isaac B. Caicedo-Castro***** — Marie-Christine Fauvet* — Ahmed Lbath* — Helga Duarte-Amaya****

* *Univ. Grenoble Alpes, LIG lab, MRIM Team, F-38000 Grenoble, France*
CNRS, LIG, F-38000 Grenoble, France
** *Univ. Nacional de Colombia, ColSWE Team, Bogotá D.C., Colombia*
*** *Univ. de Córdoba, SOCRATES Team, Montería, Colombia*
{*isaac-bernardo.caicedo-castro, marie-christine.fauvet*}*@imag.fr*

ABSTRACT. *In the study reported in this paper, we apply a family of Information Retrieval (IR) models to overcome the problem of retrieving services, whose descriptions match users' queries given in a free text style. This family is composed by four models which have not been applied in prior research on IR-based service discovery. The two first models are based on matrix factorisation models applied to Latent Semantic Indexing (LSI). The third one expands the query with terms retrieved from WordNet. The fourth model also expands the query, but with terms extracted from an automatically generated co-occurrence thesaurus. The results of the experiments suggest that the last model outperforms those most recent and prominent in the state-of-the-art on IR-based service discovery.*

RÉSUMÉ. *Dans l'étude rapportée dans cet article, nous appliquons et étudions une famille de modèles de Recherche d'Information (RI) afin de traiter le problème de la recherche de services, dont la description correspond aux requêtes des utilisateurs exprimées sous forme libre. Ainsi, nous appliquons quatre modèles qui, au meilleur de notre connaissance, n'ont été appliqués dans aucune des approches existantes de RI pour la découverte de services. Les deux premiere sont basés sur des modèles à base de factorisation de matrices appliquée à l'indexation sémantique latente (Latent Semantic Indexing, LSI). Le troisième étend la recherche avec les termes extraits du lexique WordNet. Le dernier modèle étend également la requête, mais avec des termes extraits d'un thesaurus de co-occurrence généré automatiquement. Les résultats expérimentaux que nous avons obtenus montrent que le dernier modèle étudié surpasse les autres que ceux de l'état de l'art sur la découverte de services à base de RI.*

KEYWORDS: *IR-based Service Discovery, Query Expansion, Matrix factorisation.*

MOTS-CLÉS : *Découverte de services basée sur des techniques de RI, Expansion de requêtes, Factorisation de matrices.*

1. Introduction

In this paper we consider the problem of finding Web services that fulfil users' requirements expressed in free text queries (i.e., queries composed by one or several terms instead of a specific language with operators). Examples of such requirements might be booking a hotel room, or reserving a table in a restaurant located in a certain city, etc.

Given a query, discovering services have been addressed by Information Retrieval (IR) models. In this context, the corpuses are composed by collection of WSDL[1] documents. Such kind of documents contains syntactically-based description including service name, operations name and signature, and sometimes descriptions in natural language are also given.

Despite that WSDL is the standard for service description, we adopted OWL-S (Burstein *et al.*, 2004) which is an OWL[2] ontology for describing, discovering, composing, enacting, and monitoring services. With OWL-S it is possible to describe WSDL-based services as well as those based on REST-ful[3] architecture.

In this work, we apply two Latent Semantic Indexing (LSI) models based on matrix factorisation techniques. To the best of our knowledge, none of them have been used in prior research on service discovery. Moreover, we apply other two models based on query expansion. Furthermore, we carried out experiments for comparing the effectiveness of these models with others proposed in prior research.

Our contribution is many fold: 1) the results of our experiments suggest extending queries with terms extracted from a co-occurrence thesaurus is the model which outperforms all the others, also studied in this work, 2) we show as well, that in retrieving services models based on Latent Dirichlet Allocation (LDA) have a lesser effectiveness than all the LSI-based models considered in this paper, 3) our results reveal that there is no difference which is statistically significative in the effectiveness among the LSI-based models assessed in this work. Furthermore, 4) LSI-based models have a similar effectiveness than the expansion of queries with terms extracted from WordNet.

The remainder of this paper is outlined as follows: while next Section (Section 2) discusses which issues affect the effectiveness of IR models used for the retrieval of text-based service, Section 3 presents a literature survey about IR-based service discovery. Section 4 details the family of models proposed and studied in this work. Section 5 describes the experimental setting used in this study, presents and discusses the results of our experiments. Finally, in Section 6, we conclude the paper.

1. *Web Service Description Language*, see `www.w3.org/TR/wsdl20/`
2. *Web Ontology Language*, a/k/a/ DAML-S, see `www.w3.org/TR/owl-features/`
3. Representational State Transfer, REST, see `www.w3.org/2012/ldp/charter`

```
1  <profile:Profile>
2  ...
3  <profile:serviceName>
4    WorldwideHotelInfoService
5  </profile:serviceName>
6  <profile:textDescription>
7    This service returns information of all famous
8    hotels in the world.
9  </profile:textDescription>
10 ...
11 </profile:Profile>
```

Figure 1. *OWL-S profile of a web service from the collection called OWLS-TC4 (currently available in: http://projects.semwebcentral.org/projects/owls-tc/).*

2. What does affect the effectiveness of IR-based service discovery systems?

The structure of an OWL-S ontology is designed to provide the knowledge about three aspects of a Web service, i.e., 1) What does it provide? 2) How is it used? 3) How to interact with it? The functional description of a service is given in the service profile in a way that it is suitable for a software agent searching for services (Burstein *et al.*, 2004). Figure 1 depicts a chunk of a service profile described in a OWL-S document. Tags <profile:serviceName> and <profile:textDescription> introduce the name of the service and its description in free text, respectively. The problem is to measure the extent to which this information matches a query, which is a mean utilised by users in an attempt to communicate their needs to a system designed for discovering services.

Services descriptions are briefer than usual documents (e.g., books) (see Figure 1), so, in the domain of service retrieval there are term mismatch problems (e.g., synonymy and homonymy problems) because of service descriptions are brief. Service providers use sentences (or sometimes short paragraphs) to describe desired services, hence, when these descriptions are different to the sentences in queries, this causes term mismatch problems in classical models which depend on the observable text features rather than the hidden semantic features (Ponte and Croft, 1998; Zhai and Lafferty, 2004). For instance, suppose a user submits the query: I want to book an apartment, and assuming the description of the desired service is: This service allows users to reserve a flat. In this example, terms such as book and reserve are synonyms as terms apartment and flat either. In this example, IR models based on the matching among terms (e.g., Vector Space Model or Boolean Retrieval Model) are not able to find services whose descriptions match that sort of query. As a consequence, in this context term mismatch problems affect the effectiveness of a text-based retrieval system applied on service discovery. A service retrieval system with high effectiveness increases the satisfaction of users, therefore, this encourages them to use

the service retrieval system. While a system with low effectiveness disappoints users, hence, they might stop to use it permanently.

3. A Review of the Literature on IR-based Service Discovery

In this Section we analyse the prior research on IR-based service discovery, and how synonymy problems have been handled in the state-of-the-art.

The IR models applied in prior research on service discovery are as follows:

1) Vector Space Model (VSM) (Salton *et al.*, 1975),

2) Latent Semantic Indexing (LSI) (Deerwester *et al.*, 1990),

3) Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003), or

4) Hybrid models based on ontologies and LSI.

The VSM has been applied in many approaches (see for example: (Wang and Stroulia, 2003; Platzer and Dustdar, 2005; Kokash *et al.*, 2006; Lee *et al.*, 2007; Crasso *et al.*, 2008; Wu, 2012)). In these works, a set of WSDL documents composes the corpus. Some of these approaches do not tackle the synonymy problems (Platzer and Dustdar, 2005; Lee *et al.*, 2007; Crasso *et al.*, 2008; Wu, 2012). Whereas, these problems have been addressed by expanding queries and WSDL documents with synonyms of their terms (Wang and Stroulia, 2003; Kokash *et al.*, 2006). Synonyms are extracted from WordNet lexicon (Miller, 1995). Nevertheless, the query expansion based on the injection of synonyms significantly decreases precision because a term may have synonyms with different meanings depending of the context of the term in the query.

In other approaches, researchers applied LSI to cope synonymy problems in the context of service discovery (Sajjanhar *et al.*, 2004). However, factorising a matrix through SVD causes scalability issues in LSI. Therefore, other works handled this shortcoming instead of aiming to increase the effectiveness of LSI (Ma *et al.*, 2008; Wu *et al.*, 2009). Nevertheless, dealing with scalability is out the scope of this work.

LDA is another model based on latent factors in text documents. In this model the latent factors are topics, and their distribution is assumed to have Dirichlet prior. This model was applied to discover the latent topics from concepts contained in service descriptions written in OWL-S (Cassar *et al.*, 2011; Cassar *et al.*, 2013). According to the results obtained in (Cassar *et al.*, 2011; Cassar *et al.*, 2013), LDA outperforms *Probabilistic LSI* (PLSI) (Hofmann, 1999). Nonetheless, in (Cassar *et al.*, 2011; Cassar *et al.*, 2013) did not compare LDA with other models used in prior research (e.g., LSI).

Another direction to deal with the synonymy problem is to use ontologies. Therefore, several works combine LSI and ontologies (Pan and Zhang, 2009; Paliwal *et al.*, 2012). An ontology is used as a vocabulary to expand the query (Paliwal *et al.*, 2012). Another hybrid approach where K–means algorithm is used to divide

the corpus in several clusters of documents is proposed (see (Pan and Zhang, 2009)). Given a query, SVD is applied on the most similar cluster (see (Ma *et al.*, 2008)). However, this technique is complemented with a semantic–based matching, which is implemented on an ontology of services, by computing the similarity between service input and output parameters. At the end of the procedure, services are ranked according two both techniques.

The drawback of such ontology-based approach is that the human intervention is necessary, as ontologies must be built with the assistance of human experts of the domain. Therefore, the creation of ontologies is an expensive, time-consuming, tedious, and error-prone task (Gomez-Perez *et al.*, 2003; Shamsfard and Barforoush, 2004). This is why we have decided not to use any ontology.

Most of the prior research on IR-based service discovery has been evaluated with different test suites. Some of them have common data sources, such as XMethods[4], which has been a source of WSDL documents for other researches (Kokash *et al.*, 2006; Paliwal *et al.*, 2007; Lee *et al.*, 2007; Ma *et al.*, 2008; Wu *et al.*, 2009; Hao *et al.*, 2010; Paliwal *et al.*, 2012). Moreover, in other works are carried out experiments on the collection of WSDL service descriptions used in (Hess *et al.*, 2004) (see for instance (Paliwal *et al.*, 2007; Crasso *et al.*, 2008; Ma *et al.*, 2008; Wu *et al.*, 2009; Paliwal *et al.*, 2012)). Nevertheless, this collection does not include neither queries nor relevance judgments because of it is designed for machine learning applications.

In other research works are carried out experiments over the same test collection. In (Kokash *et al.*, 2006), researchers collected 40 Web services from XMethods, and they reused part of the WSDL corpus collected for matching services research conducted in (Stroulia and Wang, 2005). In (Kokash *et al.*, 2006), authors have used 447 services from the original corpus composed by 814 services, excluding a group of 366 unclassified WSDL documents. Whereas in (Lee *et al.*, 2007) the same dataset utilised in (Kokash *et al.*, 2006) have been used. In (Sajjanhar *et al.*, 2004) authors collected 47 services, 22 with description, and in the rest were assigned default descriptions based on the topic from IBM UDDI[5] registry. In (Wu, 2012), the dataset collected in the study conducted in (Klusch and Kapahnke, 2008) is used. In (Platzer and Dustdar, 2005), researchers did not carry out any evaluation. Finally, in (Cassar *et al.*, 2011; Cassar *et al.*, 2013) authors used the collection named OWLS-TC v3.0[6]. It is composed by 1007 service descriptions written in OWL-S, 29 queries, and a relevant answer set for each query.

Nowadays, there not exist any work which has completely compared all prior research works. This might be because there is no standard test collection for IR-based service discovery. This literature survey raises the following questions which are addressed in this paper: 1) Which model has the best effectiveness between LDA and

4. XMethods, see `www.xmethods.org`
5. Universal Description Discovery and Integration, see `www.uddi.org`
6. OWL-S Service Retrieval Test Collection, see `projects.semwebcentral.org/projects/owls-tc/`

LSI based on SVD? 2) Is it possible to increase the effectiveness of LSI with other matrix factorisation models? 3) Which kind of model has the best effectiveness between LSI-based models or query expansion?

4. Proposed IR models for indexing and retrieving services

4.1. *Preprocessing*

Preprocessing of a corpus involves the following steps: first, all terms from tags <profile:serviceName> and <profile:textDescription> are extracted. In this study the first tag was assumed to be codified according to camel case convention, i.e., the practice of writing identifier composed by several terms that start with a capital letter (e.g., NonNegativeMatrixfactorisation). Besides, it was assumed the terms in first tag to be separated by spaces or underscore character.

After that, either punctuations and symbols are removed. All terms are changed to lowercase and lemmatised. Stemming increases recall but decreases precision, hence, we adopted lemmatisation instead of stemming in order to get the base of dictionary form (a.k.a., lemma) of each term. We used the Northwestern University lemmatizer called MorphAdorner[7]. Finally, stop words are removed and the *Term Frequency* and *Inverse Document Frequency* (TF-IDF) (Salton *et al.*, 1975) are used to compute the term-document matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$, where m and n are the number of terms and documents, respectively.

4.2. *Latent Semantic Indexing via Minimising the Squared Error*

With *Latent Semantic Indexing* (LSI), a set of r latent (hidden) factors are inferred from patterns found in the occurrence of terms for each document. The number of factors is less than the number of either terms or documents. These factors explain these occurrences by characterising documents and terms. In case of documents, latent factors may measure dimensions, which might be uninterpretable but meaningful. On the other side, for a term, latent factors measure its occurrence in documents related to the corresponding hidden factors.

Documents and terms are represented in a joint latent semantic space, where their relationship is computed by using the inner product between their vector representation. Let $\mathbf{x}_d \in \mathbb{R}^r$ be the representation of the document d, and $\mathbf{w}_t \in \mathbb{R}^r$ be the representation of the term t, both in the latent semantic space. Components of the vector $\mathbf{x}_d$ measure the extent to which the document d expresses latent factors, as well as the components of the vector $\mathbf{w}_t$ measure the extent to the term t appears in documents related to the corresponding factors. The inner product among both vectors

7. MorphAdorner, `devadorner.northwestern.edu/maserver/lemmatizer`

yields the TF-IDF feature of the term t into the document d. This can be written in a matrix form as follows:

$$\mathbf{Y} = \mathbf{W}^T\mathbf{X} \qquad [1]$$

where $\mathbf{W} \in \mathbb{R}^{r\times m}$, $\mathbf{X} \in \mathbb{R}^{r\times n}$, $\mathbf{Y} \in \mathbb{R}^{m\times n}$, m is the number of terms into the documents, n is the number of documents that compose the corpus, r is the number of latent factors, and $\mathbf{W}^T$ is the transposed matrix of $\mathbf{W}$. Therefore, behind LSI there is a matrix factorisation problem, which is solved by approximating a target term-document matrix $\mathbf{Y}$ as a product of two lower dimensional factor matrices ($\mathbf{W}$ and $\mathbf{X}$), where the common dimension r is smaller than m and n.

This problem could be fixed by using *Singular Value Decomposition* (SVD, see (Deerwester *et al.*, 1990)). The strategy consists of choosing $\mathbf{W}^T = \mathbf{U}_r\mathbf{D}_r$ and $\mathbf{X} = \mathbf{V}_r^T$, where $\mathbf{D}_r \in \mathbb{R}^{r\times r}$ is a diagonal matrix of the r largest singular eigenvalues of $\mathbf{Y}\mathbf{Y}^T$ and $\mathbf{Y}^T\mathbf{Y}$, $\mathbf{U}_r \in \mathbb{R}^{m\times r}$ and $\mathbf{V}_r \in \mathbb{R}^{n\times r}$ are orthonormal matrices, whose column vectors are eigenvectors of $\mathbf{Y}\mathbf{Y}^T$ and $\mathbf{Y}^T\mathbf{Y}$, respectively.

In this Section we present how to estimate both factor matrices ($\mathbf{W}$ and $\mathbf{X}$) by minimising the squared Frobenius norm of the matrix of approximation errors, $\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_F^2$, as follows:

$$\min_{\mathbf{X},\mathbf{W}} \frac{1}{2}\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_F^2 + \frac{\lambda}{2}(\|\mathbf{X}\|_F^2 + \|\mathbf{W}\|_F^2) \qquad [2]$$

where λ is a regularisation parameter used to avoid that the Frobenius norm of each factor matrices reaches large magnitudes. All the terms are squared in order to have a minimum global, i.e., to have convex cost function. This is useful because gradient descent may be used to find the minimum global instead of more costly heuristics such as simulated annealing or genetic algorithms. Therefore, to minimise the cost function, the gradient descent is calculated by setting the derivative of the cost function with respect to $\mathbf{W}$ as follows:

$$\mathbf{X}(\mathbf{W}^T\mathbf{X} - \mathbf{Y})^T + \lambda\mathbf{W} \qquad [3]$$

As a result, the gradient descent is:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta(\mathbf{X}(\mathbf{W}^T\mathbf{X} - \mathbf{Y})^T + \lambda\mathbf{W}) \qquad [4]$$

where η is the learning rate. This updating rule is used to estimate $\mathbf{W}$. On the other hand, taking the derivative of the Equation 2 with respect to $\mathbf{X}$ and setting it equal to zero is obtained $\mathbf{X}$ as follows:

$$\mathbf{X} = (\mathbf{W}\mathbf{W}^T + \lambda\mathbf{I})^{-1}\mathbf{W}\mathbf{Y} \qquad [5]$$

This Equation is used to compute $\mathbf{X}$, whereas $\mathbf{W}$ is updated with rule in Equation 4.

Algorithm 1 Minimising squared error-based matrix factorisation algorithm

Input: $\mathbf{Y}$, r, η_0, λ, maxIter
Output: $\mathbf{W}$ and $\mathbf{X}$

1) Initialise the $r \times m$ matrix $\mathbf{W}$ to small random values
2) **for** (i **in** 1:maxIter)
   a) $\mathbf{X} \leftarrow (\mathbf{W}\mathbf{W}^T + \lambda\mathbf{I})^{-1}\mathbf{W}\mathbf{Y}$
   b) $\eta \leftarrow \eta_0/(1 + \eta_0\lambda i)$
   c) $\mathbf{W} \leftarrow \mathbf{W} - \eta(\mathbf{X}(\mathbf{W}^T\mathbf{X} - \mathbf{Y})^T + \lambda\mathbf{W})$
3) **Return**: $\mathbf{W}$ and $\mathbf{X}$

The algorithm 1 estimates the factor matrices. The input of the algorithm is composed by the target matrix $\mathbf{Y}$, the number of latent factors r to be estimated, the number of iterations $maxIter$ used to find the minimum global, the initial learning rate η_0, and the regularisation parameter λ. The algorithm starts initialising $\mathbf{W}$ to random values (see step 1.). In the step 2.a. inside the loop, the latent factor representation for the documents ($\mathbf{X}$) is computed by applying the analytic solution of Equation 5, with $\mathbf{W}$ fixed at the version known in the current iteration given by the variable called i. The learning rate η is updated in the step 2.b. in order to go faster toward the direction of the gradient in the beginning, but it is smaller with each iteration to avoid oscillations or divergence. The learning rate that is used in this approach, is a decreasing rate which depends on the number of iterations, the regularisation parameter and the initial learning factor (Bottou, 2010). In the final step of this loop (see step 2.c.), $\mathbf{W}$ is updated according to the rule presented in Equation 4, with $\mathbf{X}$ fixed at the last known version. Finally, the output of this algorithm is composed by the factor matrices as is shown in the step 3.

Once the index is created, the latent factor representation of a query $\mathbf{q} \in \mathbb{R}^m$ is computed as follows:

$$\mathbf{x} = (\mathbf{W}\mathbf{W}^T + \lambda\mathbf{I})^{-1}\mathbf{W}\mathbf{q} \qquad [6]$$

With this latent factor representation $\mathbf{x} \in \mathbb{R}^r$, the similarity between the documents and the query projected onto the latent semantic space is computed by using similarity cosine (see Equation7) as follows:

$$sim(\mathbf{x}, \mathbf{x}_i) = \frac{\mathbf{x}^T\mathbf{x}_i}{\|\mathbf{x}\|\|\mathbf{x}_i\|} \qquad [7]$$

where the vector $\mathbf{x}_i \in \mathbb{R}^r$ is the column ith in the matrix $\mathbf{X} = (\mathbf{x}_1 \ldots \mathbf{x}_n)$.

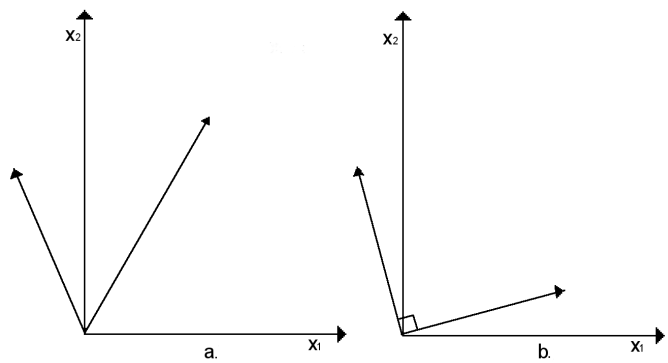

Figure 2. ***a.*** *Directions found by MSE.* ***b.*** *Directions found by SVD*

Figure 2 illustrates the geometrical differences between both matrix factorisation models. We expect that by factorising matrixes by minimising the square error is more likely to capture the latent semantic factors which better describe services than through classical LSI, because vectors in the latent semantic space might not be constrained to be orthogonal.

4.3. *Latent Semantic Indexing via NMF*

Another method to factorise $\mathbf{Y}$ is by estimating the factor matrices ($\mathbf{X}$ and $\mathbf{W}$) as two non-negative matrices which minimise the cost function as given below:

$$\min_{\mathbf{X},\mathbf{W}} \frac{1}{2}\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_F^2 \qquad [8]$$

This cost function is minimised when applying the rule below and the convergence is guaranteed (for a proof, see (Lee and Seung, 2001)):

$$W_{ij} \leftarrow W_{ij}\frac{(\mathbf{X}\mathbf{Y}^T)_{ij}}{(\mathbf{X}\mathbf{X}^T\mathbf{W})_{ij}} \qquad [9]$$

$$X_{ij} \leftarrow X_{ij}\frac{(\mathbf{W}\mathbf{Y})_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{X})_{ij}} \qquad [10]$$

This approach is known as the *Non-negative matrix factorisation* (NMF). The algorithm 2 carries out NMF. The algorithm has an input composed by the target matrix $\mathbf{Y}$, the number of latent factors r to be estimated, and the number of iterations used

Algorithm 2 Non-negative matrix factorisation algorithm

Input: $\mathbf{Y}$, r, maxIter
Output: $\mathbf{W}$ and $\mathbf{X}$

1) Initialise the $r \times m$ matrix $\mathbf{W}$ to small non-negative random values
2) Initialise the $r \times n$ matrix $\mathbf{X}$ to small non-negative random values
3) **for** (k **in** 1:maxIter)
   a) $W_{ij} \leftarrow W_{ij} \frac{(\mathbf{X}\mathbf{Y}^T)_{ij}}{(\mathbf{X}\mathbf{X}^T\mathbf{W})_{ij}}$, $\forall_{i,j \in \mathbb{N}}$ such that $1 \leq i \leq r$ and $1 \leq j \leq m$
   b) $X_{ij} \leftarrow X_{ij} \frac{(\mathbf{W}\mathbf{Y})_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{X})_{ij}}$, $\forall_{i,j \in \mathbb{N}}$ such that $1 \leq i \leq r$ and $1 \leq j \leq n$
4) **Return**: $\mathbf{W}$ and $\mathbf{X}$

to converge into the minimum global. In the steps 1 and 2 the factor matrices are initialised with non-negative random values. In the loop the updating rules in Equations 9 and 11 are used to estimate the factor matrices. Finally, the output of this algorithm is composed by the factor matrices as shown in the step 4.

The projection of a query $\mathbf{q} \in \mathbb{R}^m$ on the latent semantic space, $\mathbf{x} \in \mathbb{R}^r$, is found by using the following updating rule:

$$x_i \leftarrow x_i \frac{(\mathbf{W}\mathbf{q})_i}{(\mathbf{W}\mathbf{W}^T\mathbf{x})_i} \qquad [11]$$

With the query and documents projected onto the same latent semantic space, the cosine similarity (see Equation 7) is applied to identify relevant results.

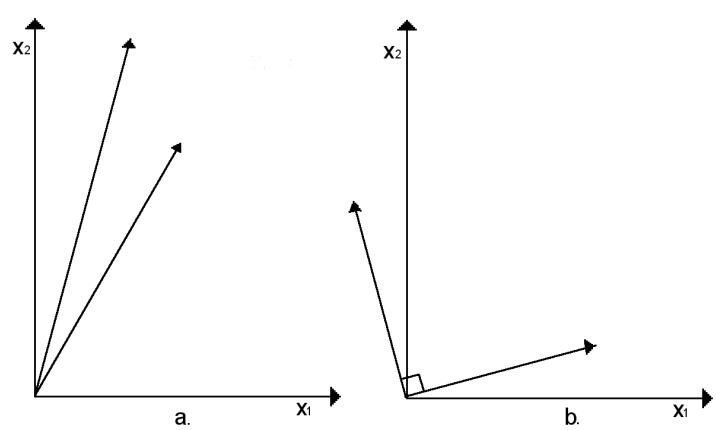

Figure 3. ***a.*** *Directions found by NMF.* ***b.*** *Directions found by SVD*

In the same way as the model presented in Section 4.2, we expect that by factorising matrixes through NMF is more likely to capture the latent semantic factors which better describe services than by mean of classical LSI, because vectors in the latent semantic space might not be constrained to be orthogonal, besides, with NMF is guaranteed that each vector in latent semantic space has only non-negative values in every direction (see Figure 3).

4.4. *Query Expansion via Wordnet*

The model presented in this section is based on VSM rather than LSI. However, the query is expanded in order to cope synonymy problems. Nevertheless, the problem with models based on query expansion, such as those used in (Wang and Stroulia, 2003; Kokash *et al.*, 2006) (see Section 3) is that this kind of models might significantly decrease precision. For instance, suppose a user issues a query like book a room, if the term *book* is considered as a noun it may be expanded with WordNet with synonyms likewise: record, ledger, lever, Word of God, account book, Bible, al-Qur'an, etc. If all these synonyms are included into the query, the system will retrieve services which are not related to reserve or book a room. However, if the same term is correctly identified as a verb, it may be expanded with WordNet with the following synonyms: hold and reserve. In this latter case, the system will retrieve services for booking a room and thanks to these synonyms the precision is increased.

Therefore, the problem in this context consists of tagging the parts of speech of the query in order to look for the synonyms in the thesaurus that will be injected into the query. In this study this problem has been tackled by using Apache OpenNLP[8] library to tag the parts of speech of a query. This library uses a probability model to predict the tag of a term based on its corresponding word type and its context in the query sentence. Thereafter, the synonyms associated with the term are sought in WordNet but considering if the term is an adverb, verb, noun or adjective.

Once the query is expanded, its vector representation is computed with the TF-IDF. Finally, services descriptions are ranked according their cosine similarity (see Equation 7) with the vector which represents the query.

4.5. *Query Expansion via a Co-Occurrence Thesaurus*

Another alternative to WordNet consists of automatically generating a thesaurus by computing the *Terms Similarity Matrix* $\mathbf{C} = \mathbf{Y}\mathbf{Y}^T$, where $\mathbf{C} \in \mathbb{R}^{m \times m}$ and each component C_{ij} represents the similarity score between terms t_i and t_j. The similarity between two syntactically different terms depends on the extent to which both terms appear together (or co-occur) in several service descriptions. In this model the aim is to analyse the latent factors hidden in these co-occurrences of terms. Thereafter,

8. Apache OpenNLP, `opennlp.apache.org/`

the latent factors of each column vector of this matrix are computed, by factorising it through the above mentioned methods in order to obtain $\mathbf{W} \in \mathbb{R}^{r \times m}$ and $\mathbf{X} \in \mathbb{R}^{r \times m}$ such that $\mathbf{C} = \mathbf{W}^T\mathbf{X}$. Let Q be a set of terms used in a query, and let V be a set of terms used to describe services (a.k.a. vocabulary), each term $t_i \in V - Q$ of the thesaurus is added to the query if $sim(\mathbf{x}_i, \mathbf{x}_j) > \theta$ (see Equation 7), where $t_j \in Q$. The parameter θ is estimated by means of experiments.

Algorithm 3 Query expansion algorithm via co-occurrence thesaurus

Input: $\mathbf{X}$, Q, V, θ
Output: Q_e

1) Initialise the expanded query $Q_e \leftarrow Q$
2) **for** ($t_j \in Q \cap V$)
   a) **for** ($t_i \in V - Q$) **If** $cos(\mathbf{x}_i, \mathbf{x}_j) > \theta$ **then** $Q_e \leftarrow Q_e \cup \{t_i\}$
3) **Return**: Q_e

The algorithm 3 carries out the expansion of queries. The input of the algorithm includes the factor matrix $\mathbf{X}$, the terms of the query Q and the vocabulary V, and the parameter θ. The algorithm returns the expanded query Q_e.

Once the query has been expanded, its vector representation is computed. Thus, given this vector, cosine similarity is applied to rank each vector which represents a service description.

5. Evaluation

5.1. *Experimental Setting*

We carried out experiments using the fourth version of the OWL-S service retrieval test collection named OWLS-TC4[9] which contains the descriptions of 1083 Web services from 9 domains (i.e., education, medical care, food, travel, communication, economy, weapon, geography, and simulation). Each description is given in OWL-S 1.1. This collection includes 42 queries associated with their relevance judgment provided by several users. A pooling strategy (as used in TREC[10]) was conducted to collect the relevance judgment set which was obtained from the top-100 results of participants of the S3 contest[11] in 2008. The judgment relevance has been graded in four different levels, i.e., highly relevant (value 3), relevant (value 2), potentially relevant (value 1), and non-relevant (value 0). Therefore, during the experiments the *Normalised Discounted Cumulative Gain at 10* (NDCG@10) has been used instead

9. OWL-S Service Retrieval Test Collection, `projects.semwebcentral.org/projects/owls-tc/`
10. Text Retrieval Conference, `trec.nist.gov/`
11. Semantic Service Selection, `www-ags.dfki.uni-sb.de/~klusch/s3/`

of the *Mean Average Precision* (MAP) to measure the overall ranking effectiveness of each approach.

As we have mentioned in Section 1, service descriptions are sentences or paragraphs, which are briefer than usual documents. Indeed, after removing all stop words in the collection called OWLS-TC4[12], it has:

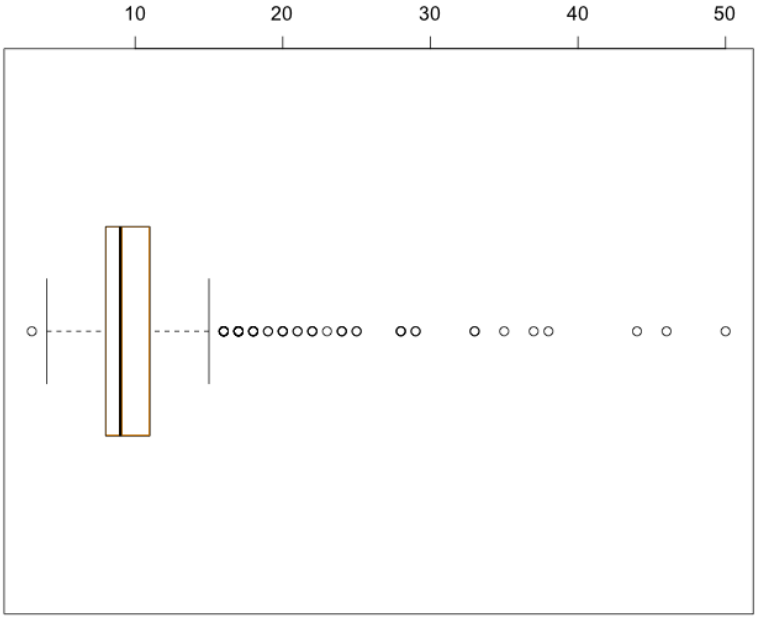

Figure 4. *Number of terms frequently used to describe services*

– minimum 3 terms per description,
– 8 terms per description in the first quartile,
– a median of 9 terms per description,
– 11 terms per description in the third quartile
– a mean of 9.99 terms per description,
– with a standard deviation of 4.07 terms, and
– maximum 50 terms per descriptions.

Figure 4 depicts a box plot which shows that most of the services are described roughly with 10 terms. Only few services are described with more than 30 terms, and these services correspond with outliers in the box plot.

12. The OWL-S profiles from the collection named OWLS-TC4 (`http://projects.semwebcentral.org/projects/owls-tc/`, retrieved on July of 2014)

This collection is the unique one which exists in service retrieval domain which has judgment relevance. Previous versions of this collection were used for carrying out experiments in related recent works (Cassar *et al.*, 2011; Cassar *et al.*, 2013).

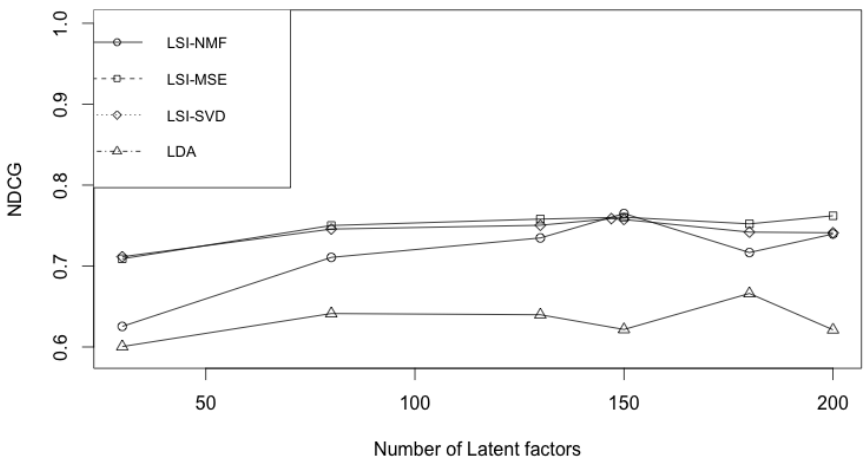

Figure 5. *Effectiveness (measured through NDCG@10) of LSI models and LDA given the number of latent factors (or topics)*

The best settings for LSI via SVD, MSE, and NMF are achieved when the number of latent factors are 147, 200, and 150, respectively (see Figure 5). The best setting for LDA is achieved with 180 latent topics (see Figure 5). Besides, we assessed LDA with several value for α. We found the best setting for α is the same suggested in (Griffiths and Steyvers, 2004). Moreover, to factorise matrixes by minimising the squared error, the best setting is obtained for the initial η and λ, when their values are 0.2 and 0.001, respectively. Furthermore, the co-occurrence thesaurus is generated with the same three matrix factorisation models above mentioned. Several values of θ are set for this approach (0.85, 0.90, 0.95, and 0.99). The greatest effectiveness of this models is obtained by generating the thesaurus and factorising the term similarity matrix via the technique to minimise the squared error with 200 latent factors, and θ equal to 0.95. With SVD the best performance is achieved with 220 latent factors and θ equal to 0.90. Finally, with NMF the greatest effectiveness is obtained with 130 latent factors and θ equal to 0.90.

5.2. *Results*

Table 1 presents the results we obtained from the experiments. The three first rows show the retrieval effectiveness we obtained for existing techniques (VSM, LDA and

LSI). Then the six last rows show results for our model extensions: *Query Expansion via a Co-Occurrence Thesaurus* (QECOT) automatically generated through SVD is called QECOT-SVD, QECOT generated using the method MSE, QECOT-MSE, QECOT generated through NMF, QECOT-NMF. Similarly, in the rest of the paper we denote LSI-SVD the technique LSI via SVD, LSI-MSE LSI via MSE, LSI-NMF LSI via NMF. Eventually, *WordNet-based Query Expansion* is shortened to WN-QE. The outcomes of the experiments suggest that QECOT-MSE outperforms all the above mentioned models in terms of effectiveness (see Table 1).

Table 1. *Retrieval effectiveness.*

Model	NDCG@10	Gain (%)
Models applied in prior research on IR-based service discovery		
VSM (baseline)	0.5435	N/A
LDA	0.6661	22.55
LSI-SVD	0.7586	39.57
Proposed family of models for text-based service retrieval		
LSI-MSE	0.7621	40.22
WN-QE	0.7645	40.66
LSI-NMF	0.7649	40.73
QECOT-NMF	0.7792	43.37
QECOT-SVD	0.7804	43.59
QECOT-MSE	**0.7897**	45.29

Table 2. *Student's paired t-test on NDCG@10 to compare QECOT-MSE with other models applied in prior research on IR-based service discovery*

Model QECOT-MSE	NDCG@10 **0.7897**	p-value	is statistically significant?
VSM	0.5435	3.47×10^{-7}	Yes
LSI-SVD	0.7586	0.08039	No
LDA	0.6661	7.613×10^{-5}	Yes

5.3. *Discussion*

The results we got suggest that QECOT-MSE outperforms all the models studied in the paper. Indeed, Table 2 shows that the effectiveness of QECOT-MSE is better than LDA and VSM, which are the models applied for service retrieval in (Wang and Stroulia, 2003; Platzer and Dustdar, 2005; Kokash *et al.*, 2006; Lee *et al.*, 2007; Crasso *et al.*, 2008; Wu, 2012; Cassar *et al.*, 2011; Cassar *et al.*, 2013), and the difference is statistically significant. Despite the difference between the effectiveness of QECOT-MSE and LSI-SVD is not statistically significant, the first model outperformed the

second one in more queries. Indeed, in 5 queries both models had the same effectiveness, in 24 queries QECOT-MSE outperformed LSI-SVD, and only in 13 queries LSI-SVD has better effectiveness than QECOT-MSE (see Figure 6). Figure 6 depicts a comparison of the effectiveness of both models regarding each query used in the experiments. Points below the diagonal line correspond with queries where LSI-SVD outperformed QECOT-MSE (13 points). Whereas points above the line correspond with queries where QECOT-MSE outperformed LSI-SVD (24 points). Finally, points in the diagonal line correspond with queries where both models had the same effectiveness (5 points).

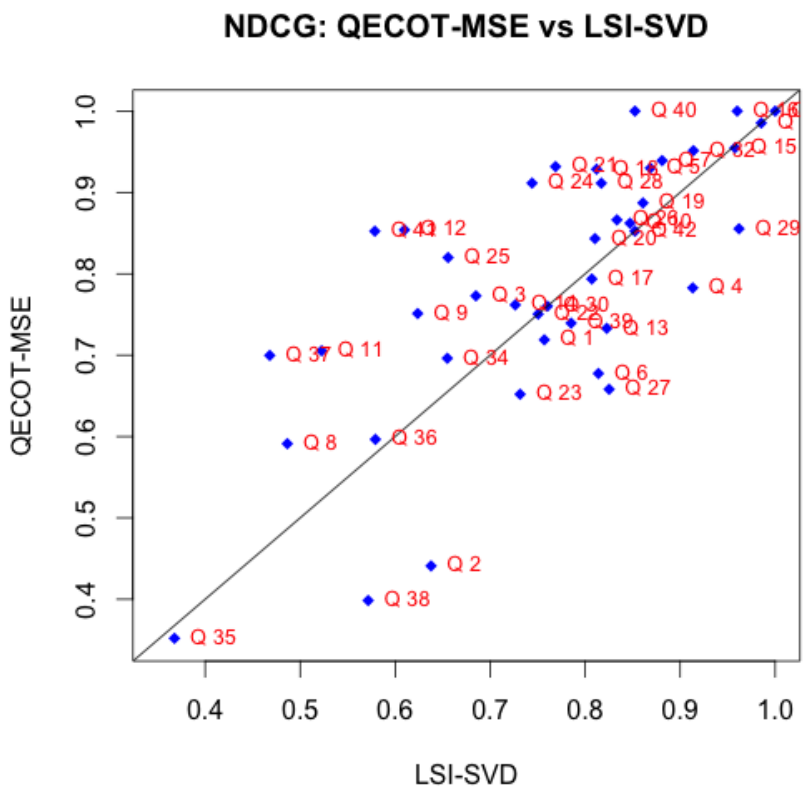

Figure 6. *Comparison between QECOT-MSE and LSI-SVD*

Another remarkable result suggest that LDA is outperformed by LSI-SVD (see Table 3), although the first models has been proposed to improve LSI. Perhaps the low effectiveness of LDA is due to the low term frequencies in service descriptions. This answers the first question stated in Section 3.

LSI-NMF outperformed LSI-MSE, likewise the latter has a NDCG@10 greater than the one achieved with the classical LSI-SVD. This suggests that vectors in the latent semantic space might not be constrained to be orthogonal. Besides, with NMF is guaranteed that each vector in this space has only non-negative values in every direction. Nevertheless, the difference between NDCG@10 values is not statistically significant as it is shown in Table 3. This fact answers the second question stated in Section 3.

Furthermore, the evaluation reveals that LSI-NMF had a similar performance than WN-QE (see Table 3). This might be the reason why LSI and query expansion have been the two main streams in the state-of-the-art. Nevertheless, QECOT-MSE had the best effectiveness, this suggests that query expansion might outperform LSI-models. This fact answers the third question stated in Section 3.

Table 3. *Student's paired t-tests on NDCG@10 to compare LSI-NMF with other LSI-based models, WN-QE, and LDA*

Model	NDCG@10	p-value	is statistically significant?
LSI-NMF	0.7649		
Models applied in prior research on IR-based service discovery			
LSI-SVD	0.7586	0.6855	No
LDA	0.6661	8.37×10^{-4}	Yes
Proposed family of models for text-based service retrieval			
LSI-MSE	0.7621	0.8466	No
WN-QE	0.7645	0.9718	No

6. Conclusion

In this work we have studied four models which have not been already applied on the retrieval of text-based services. Two of them are Latent Semantic Indexing models via Non-negative Matrix Factorisation and Minimising the Squared Error. The other two are based on query expansion: the first one expands the query by injecting synonyms extracted from WordNet, and taking into account the parts of speech of the query; the second one expands the query with terms extracted from a co-occurrence thesaurus.

The contribution of the study reported in this paper is manyfold: 1) The outcomes of the experiments suggest the expansion of queries via co-occurrence thesaurus outperforms the other models studied in this work. 2) The results suggest that LDA has a lesser effectiveness than all the LSI-models evaluated in this work. However, for further research, this model might be used for expanding queries. 3) The results reveal the difference in the effectiveness between the underlying factorisation technique used in this study to implement LSI models statistically significant. Moreover, LSI-based models had a similar performance than the injection of synonyms from WordNet taking into account the parts of speech of the query.

For future work, the co-occurrence thesaurus may be made with the corpus of another source of knowledge, such as Wikipedia. Besides, LSI might be implemented with matrix factorisation based on the kernel trick.

Acknowledgements: Authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

7. References

Blei D. M., Ng A. Y., Jordan M. I., "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, vol. 3, p. 993-1022, March, 2003.

Bottou L., "Large-Scale Machine Learning with Stochastic Gradient Descent", *in* Y. Lechevallier, G. Saporta (eds), *Proc. of the 19th International Conference on Computational Statistics*, Springer, p. 177-187, August, 2010.

Burstein M., Hobbs J., Lassila O., Mcdermott D., Mcilraith S., Narayanan S., Paolucci M., Parsia B., Payne T., Sirin E., Srinivasan N., Sycara K., OWL-S: Semantic Markup for Web Services, Technical report, World Wide Web Consortium, 2004.

Cassar G., Barnaghi P., Moessner K., "A Probabilistic Latent Factor approach to service ranking", *Proc. of the International Conference on Intelligent Computer Communication and Processing*, p. 103-109, Aug, 2011.

Cassar G., Barnaghi P., Moessner K., "Probabilistic Matchmaking Methods for Automated Service Discovery", *IEEE Transactions on Services Computing*, vol. 7, n° 4, p. 1-1, May, 2013.

Crasso M., Zunino A., Campo M., "Easy Web Service Discovery: A Query-by-example Approach", *Science of Computer Programming*, vol. 71, n° 2, p. 144-164, April, 2008.

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R., "Indexing by latent semantic analysis", *Journal of the American Society for Information Science*, vol. 41, n° 6, p. 391-407, 1990.

Gomez-Perez A., Corcho-Garcia O., Fernandez-Lopez M., *Ontological Engineering*, Springer-Verlag New York, Inc., 2003.

Griffiths T. L., Steyvers M., "Finding scientific topics", *Proc. of the National Academy of Sciences*, vol. 101, n° 1, p. 5228-5235, April, 2004.

Hao Y., Zhang Y., Cao J., "Web services discovery and rank: An information retrieval approach.", *Future Generation Computer Systems*, vol. 26, n° 8, p. 1053-1062, 2010.

Hess A., Johnston E., Kushmerick N., "ASSAM: A tool for semi-automatically annotating Web Services with semantic metadata", *Proc. of the International Semantic Web Conference*, Springer, 2004.

Hofmann T., "Probabilistic Latent Semantic Indexing", *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 50-57, 1999.

Klusch M., Kapahnke P., "Semantic Web Service Selection with SAWSDL-MX", *Proc. of the 2nd International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web*, CEUR-WS.org, 2008.

Kokash N., van den Heuvel W.-J., D'Andrea V., "Leveraging Web Services Discovery with Customizable Hybrid Matching", *Proc. of the 4th International Conference on Service Oriented Computing*, p. 522-528, 2006.

Lee D. D., Seung H. S., "Algorithms for Non-negative Matrix Factorization", *Advances In Neural Information Processing Systems*, MIT Press, p. 556-562, 2001.

Lee K.-H., Lee M.-Y., Hwang Y.-Y., Lee K.-C., "A Framework for XML Web Services Retrieval with Ranking.", *Proc. of the International Conference on Multimedia and Ubiquitous Engineering, 2007.*, IEEE Computer Society, p. 773-778, 2007.

Ma J., Zhang Y., He J., "Web Services Discovery Based on Latent Semantic Approach.", *Proc. of the International Conference on Web Services*, IEEE Computer Society, p. 740-747, 2008.

Miller G. A., "WordNet: A Lexical Database for English", *Communications of the ACM*, vol. 38, n° 11, p. 39-41, November, 1995.

Paliwal A. V., Adam N. R., Bornhövd C., "Web Service Discovery: Adding Semantics through Service Request Expansion and Latent Semantic Indexing.", *Proc. of the International Conference on Services Computing*, IEEE Computer Society, p. 106-113, 2007.

Paliwal A. V., Shafiq B., Vaidya J., Xiong H., Adam N. R., "Semantics-Based Automated Service Discovery.", *IEEE Transactions of Services Computing*, vol. 5, n° 2, p. 260-275, 2012.

Pan S.-L., Zhang Y.-X., "Ranked Web Service Matching for Service Description Using OWL-S", *Proc. of the International Conference on Web Information Systems and Mining*, p. 427-431, Nov, 2009.

Platzer C., Dustdar S., "A Vector Space Search Engine for Web Services", *Proc. of the 3rd International Conference on Web Services*, IEEE Computer Society, p. 14-16, 2005.

Ponte J. M., Croft W. B., "A Language Modeling Approach to Information Retrieval", *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, p. 275-281, 1998.

Sajjanhar A., Hou J., Zhang Y., "Algorithm for Web Services Matching", *in* J. Yu, X. Lin, H. Lu, Y. Zhang (eds), *Proc. of the 6th Asia-Pacific Web Conference*, vol. LNCS 3007, Springer Berlin Heidelberg, p. 665-670, 2004.

Salton G., Wong A., Yang C. S., "A Vector Space Model for Automatic Indexing", *Communications of the ACM*, vol. 18, n° 11, p. 613-620, November, 1975.

Shamsfard M., Barforoush A. A., "Learning ontologies from natural language texts", *International Journal of Human–Computer Studies*, vol. 60, p. 17-63, 2004.

Stroulia E., Wang Y., "Structural and semantic matching for assessing web-service similarity", *International Journal of Cooperative Information Systems*, vol. 14, p. 407-437, 2005.

Wang Y., Stroulia E., "Semantic Structure Matching for Assessing Web Service Similarity", *Proc. of the 1st International Conference on Service Oriented Computing*, Springer-Verlag, p. 194-207, 2003.

Wu C., "WSDL Term Tokenization Methods for IR-style Web Services Discovery", *Science of Computer Programming*, vol. 77, n° 3, p. 355-374, March, 2012.

Wu C., Potdar V., Chang E., "Latent Semantic Analysis – The Dynamics of Semantics Web Services Discovery", *Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web*, Springer-Verlag, p. 346-373, 2009.

Zhai C., Lafferty J., “A Study of Smoothing Methods for Language Models Applied to Information Retrieval”, *ACM Transactions Information Systems*, vol. 22, n° 2, p. 179-214, April, 2004.