Découverte et recommandation de services web basées sur l'extraction d'information et la réputation symbolique

Mustapha Aznag, Mohamed Quafafou, Nicolas Durand, Zahi Jarir

▶ **To cite this version:**

Mustapha Aznag, Mohamed Quafafou, Nicolas Durand, Zahi Jarir. Découverte et recommandation de services web basées sur l'extraction d'information et la réputation symbolique. Atelier Sources Ouvertes et Services - Données Liées pour un Web de Données (SOS-DLWD'12) de la 12ème Conférence ă Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC 2012), 2012, Bordeaux, France. pp.7-17. hal-01465119

HAL Id: hal-01465119
https://hal.science/hal-01465119

Submitted on 18 May 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.

Découverte et recommandation de services web basées sur l'extraction d'information et la réputation symbolique

Mustapha AZNAG*, Mohamed QUAFAFOU*, Nicolas DURAND*, Zahi JARIR**

*Laboratoire des Sciences de l'Information et des Systèmes (LSIS UMR 6168)
Université Aix-Marseille - Avenue Escadrille Normandie Niemen - 13397 Marseille
{mustapha.aznag, nicolas.durand, mohamed.quafafou}@univmed.fr
** Laboratoire d'Ingénierie des Systèmes Informatiques (LISI)
Université Cadi Ayyad - Boulevard Prince My Abdellah, B.P. 2390 | 40000 Marrakech
zahijarir@ucam.ac.ma

Résumé. Cet article montre que le problème de représentation de services web est crucial et analyse les différents facteurs qui l'influencent. Il discute une représentation classique et en propose deux nouvelles. La première représentation que nous proposons provient du domaine du traitement du langage naturel et est basée sur des règles pour annoter les descriptions de services et ainsi extraire les informations utiles pour l'indexation sémantique de services. La seconde méthode proposée, appelée réputation symbolique, est calculée à partir des relations entre les services considérés et est utilisée pour la recommandation de services web. L'impact de ces représentations pour la découverte et la recommandation est étudié et discuté à la lumière de nos expérimentations utilisant des services web réels.

1 Introduction

Un service web [1](W3C, 2004) est défini comme un programme modulaire mis à disposition sur le web par un fournisseur de services et destiné à supporter l'interaction machine-machine ou application-application dans des environnements distribués. Il est basé sur l'architecture orientée service (SOA). Notons que, dans le cas des services web, le mécanisme de communication est indépendant de tout langage de programmation et de toute plateforme d'exécution. Il existe plusieurs types de services web : (1) Les services web de type **REST**(Representational state transfer) et (2) et les services web de type **WS-***. Les services REST sont basés sur l'architecture du web et ses standards de base HTTP et URI. Les spécifications des services de type WS-* reposent sur les standards W3C SOAP (Simple Object Access Protocol) et WSDL (Web Services Description Language). Dans cet article nous avons étudié les services web de type WS-*. L'accès à ces services est réalisé par l'intermédiaire de messages SOAP. Ils utilisent une syntaxe basée sur la notation XML pour décrire les appels de fonctions distantes et les données échangées, et organisent les mécanismes d'appel et de réponse. Comme exemple de service web, nous pouvons évoquer les services météorologiques permettant à l'utilisateur

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

de connaître la météo selon sa localisation. Les services web sont publiés dans des annuaires comme par exemple les annuaires UDDI (Universal Description Discovery and Integration).

Le nombre de serveurs offrant des services web, grâce à des moteurs spécifiques comme Axis [2], est en constante progression. Il est alors important d'automatiser l'indexation de ces services au niveau des annuaires. Pour cela, les documents WSDL décrivant les services peuvent être collectés et analysés automatiquement afin de générer des représentations structurées (vectorielles) de ces services. Les documents WSDL contiennent des descriptions textuelles, des opérations (au sens fonctions appelables à distance), des types de données simples ou complexes, etc. Les descriptions utilisées sont en langage naturel, multilingues et inter-domaines. Générer des représentations de services web est donc un défi majeur. De plus, la qualité des représentations a une forte incidence sur les annuaires qui offrent des mécanismes de découverte, de composition et de recommandation de services web. Ces différentes tâches ont été intensément étudiées ces dernières années (Maximilien et Singh, 2004; Dong et al., 2004; Xu et al., 2007; Yan et Piao, 2008; Petrova-Antonova et Ilieva, 2009; Omer et Schill, 2009; Elgazzar et al., 2010). Cependant, plusieurs problèmes restent posés : l'information décrivant les services est très hétérogène et ne respecte pas forcément les règles de la langue, de nombreux services souvent des services sans descriptions et enfin très peu de travaux ont porté sur l'impact des représentations utilisées pour la découverte et la recommandation.

Cet article présente une méthode classique de génération de représentations de services web, et en propose deux nouvelles. La première méthode, appelée *REI* (Reconnaissance et Extraction d'Information), se base sur des techniques de traitement du langage naturel et utilise des règles pour annoter les descriptions des services et ainsi extraire les informations pertinentes. La seconde méthode proposée, appelée *RS* (Réputation Symbolique), calcule les représentations à partir des relations entre les différents services web considérés. Nous présentons les algorithmes adéquats de découverte et de recommandation, et nous les utilisons pour confronter les différentes représentations.

La section 2 détaille le traitement appliqué aux documents WSDL. La section 3 introduit les différentes représentations de services web et explique comment chacune affine l'autre ou offre une nouvelle alternative. La section 4 est dédiée aux algorithmes adéquats de découverte et de recommandation de services web. Les expérimentations et les résultats sont présentés dans la section 5 avant de conclure en section 6.

2 Traitement des documents WSDL

Le langage standard WSDL se base sur XML et sert à décrire les services web. Le document WSDL d'un service web permet d'indiquer, grâce à des balises (*binding*, *message*, *operation*, *service*, etc.), le protocole de communication utilisé, le format de messages requis pour communiquer avec ce service, les opérations que le client peut invoquer et la localisation de ce service. Les versions 1.1 et 2.0 de WSDL [3] sont prises en compte dans nos travaux.

Après avoir vérifié la disponibilité d'un service web et validé le contenu de son document WSDL, ce dernier est analysé afin d'extraire toutes les informations potentiellement utiles. Ces dernières sont (a) le nom, la documentation et la version du document, (b) les types simples

2. http ://axis.apache.org/axis/
3. v1.1 http ://www.w3.org/TR/wsdl ; v2.0 http ://www.w3.org/TR/wsdl20/

ou complexes utilisés par les messages pour transmettre des informations, (c) le nom et la documentation du service, et (d) l'interface définissant les opérations disponibles. Pour chacune de ses opérations, le nom, la documentation, les arguments et les valeurs de retour sont récupérés. Remarquons que les "documentations" sont des descriptions textuelles correspondant à des commentaires.

Les informations extraites seront utilisées lors de la construction des représentations. Avant de présenter les méthodes permettant le calcul des représentations, nous évoquons quelques traitements textuels classiques utilisés par la suite. Ces traitements concernent : (1) **Suppression des balises :** les documentations peuvent être écrites en HTML. Nous supprimons donc toutes les balises HTML, les composantes CSS, etc. (2) **Séparation des mots :** certains termes sont composés de plusieurs mots séparés par une lettre majuscule, nous utilisons alors des expressions régulières pour extraire ces mots. Par exemple, l'application de *[A-Z][a-z]+* sur la chaîne *"GetAllCountryCurrenciesResponse"* produit *'Get'*, *'All'*, *'Country'*, *'Currencies'* et *'Response'*. (3) **Suppression des mots vides :** les mots vides de sens (articles, pronoms, etc.) sont supprimés. (4) **Lemmatisation :** l'algorithme de Porter (Porter, 1980) est utilisé pour calculer le *lemme* de chaque mot. Etant donné que les mots ayant le même lemme ont généralement le même sens, seuls les lemmes sont conservés. Par exemple, les mots '*computing*' et '*compute*' produisent le même lemme '*comput*'.

3 Différentes représentations de services web

Dans cette section, nous présentons une méthode classique de construction de représentations de services web et en proposons deux nouvelles. Notons que les représentations générées sont des représentations vectorielles.

3.1 Utilisation des Documentations et des Types WSDL (DT) :

La première représentation est centrée sur les descriptions textuelles. Pour un service web, toutes les documentations présentes dans le document WSDL (documentation générale, noms et documentations des opérations) sont concaténées et nous appliquons des traitements textuels classiques utilisés par la communauté *recherche d'information* et résumés dans la section 2 pour ensuite construire une description générale d'un service web donné, que nous appelons *descriptionGenerale*.

Notons que le principal inconvénient est que la plupart des services web ont des documentations de mauvaise qualité ne respecte pas les règles de syntaxe ou de grammaire ou n'en ont pas du tout. Pour faire face à ces limites nous rajoutons les types WSDL pour ensuite construire la représentation vectorielle pour un service web donné. Les types sont utilisés par les messages pour transmettre des informations entre services. Par conséquent, les types sont de pertinentes informations pour décrire la fonctionnalité d'un service web. Pour un service donné, tous les noms de types WSDL (types simples/complexes, etc.) sont alors ajoutés à son ensemble de descriptions *descriptionGenerale* formé précédemment. La représentation est construite donc après application des traitements 2, 3 et 4 de la section 2 sur les nouvelles données utilisées.

Cette approche est classique et est basée sur des traitements syntaxiques pour construire la représentation d'un service. Elle est bien connue par la communauté des services web. Nous introduisons à présent deux nouvelles méthodes de calcul de représentations notées *REI* et *RS*.

3.2 Reconnaissance et extraction d'information (REI) :

De nombreux services web ont des documentations trop détaillées. En particulier quand ils offrent beaucoup d'opérations et chacune avec sa propre description. Dans ce contexte, il est donc important de savoir reconnaître les parties ou les entités importantes dans ces textes afin de construire de meilleures représentations.

Notre approche consiste, grâce à des règles, à identifier, annoter et extraire les termes (simples ou multiples) pertinents à partir des descriptions textuelles disponibles pour un service web. Notons que nous avons utilisé auparavant l'approche dans le domaine de la génétique. Le lecteur intéressé peut se reporter à l'article (Charnois et al., 2006) pour plus de détails. Dans le cadre des descriptions de services web, nous avons écrit un ensemble de règles en Prolog traduisant la grammaire à clauses définies (DCG) mise en place. Ces règles sont déclaratives et rendent le traitement léger et rapide. Elles sont spécifiques au domaine mais elles ne sont pas spécifiques au corpus de services web utilisé. L'approche est endogène c'est-à-dire qu'aucune ressource telle qu'un dictionnaire n'est nécessaire. L'implantation a été réalisée avec la plateforme LinguaStream [4] (Bilhaut et Widlöcher, 2006). Cette plateforme est un environnement d'expérimentation intégré destiné aux chercheurs en traitement du langage naturel.

Nous considérons la description générale *descriptionGenerale* formée précédemment dans la section précédente 3.1 et nous appliquons sur cette description les étapes de traitement du processus de la méthode *REI*. Les étapes de traitement que nous appliquons ("tokenization", "Part-Of-Speech tagging" i.e. étiquetage des parties du discours, extraction avec les règles définies, etc.) ont été mises en oeuvre sous forme de modules. Notons que nous avons utilisé TreeTagger [5] pour l'étape "Part-Of-Speech tagging". Les règles fonctionnent de la façon suivante. A partir d'un *contexte*, une expression (en général un terme de plusieurs mots, une phrase nominale, etc.) est reconnue jusqu'à ce qu'une "phrase d'arrêt" soit rencontrée. Le contexte est un ensemble de mots déclencheurs. Les phrases d'arrêt peuvent être des mots, des symboles, des verbes, la ponctuation, etc.

Enfin, Nous rajoutons tous les noms de types WSDL à l'ensemble des termes construits par le processus précédent. La représentation vectorielle d'un service web donné est construite donc après application des traitements 2, 3 et 4 de la section 2 sur les nouvelles données considérées.

3.3 Réputation symbolique (RS) :

Toutes les représentations précédentes sont produites directement à partir du contenu de documents WSDL qui représentent le point de vue du fournisseur. Notre objectif, à présent, est de construire une représentation à partir de son contexte (i.e. les services voisins). Cela est particulièrement intéressant dans les cas où les documentations d'un service web sont absentes.

Pour cela, nous nous sommes basés sur la notion de réputation d'un service web qui reflète une perception commune des autres services web ou des consommateurs à l'égard de ce service. En d'autres termes, elle agrège les évaluations du service rendu par les consommateurs de service. Typiquement, une réputation sera construite à partir d'un historique de notations par les différents acteurs. Plusieurs systèmes de réputation ont été proposés dans la littérature (Xu

4. http ://www.linguastream.org
5. http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

et al., 2007; Maximilien et Singh, 2002; Majithia et al., 2004; Goldbeck et Hendler, 2004; Nepal et al., 2011). Cependant, un système de réputation qui se base sur le retour des utilisateurs peut utiliser des évaluations malhonnêtes. Des modèles de Markov cachés (Hidden Markov Models) peuvent être utilisés pour prédire la réputation d'un service web (Malik et al., 2009).

Notre travail introduit un nouveau modèle, la *réputation symbolique*, en tenant compte de l'aspect qualitatif de la réputation des services web. Elle est produite à partir de la représentation (qui est une description symbolique) des autres services web. Afin de calculer la réputation symbolique, nous modélisons les relations entre les services par un graphe de dépendance. Ce dernier forme un réseau de services qui est notamment utilisé pour traiter différentes problématiques comme la composition de services web (Hashemian et Mavaddat, 2005; Omer et Schill, 2009). L'approche générale de création de ce graphe consiste à rechercher les dépendances entre les opérations disponibles pour les services en utilisant les paramètres d'entrée et de sortie. Ensuite, le graphe de dépendance des services web, présenté dans la définition 1, est induit. Nous considérons ici qu'une dépendance se produit entre deux services lorsqu'ils offrent deux opérations dépendantes : un service nécessite (resp. offre) des données à partir (resp. à) d'un autre service. La définition 2 présente plus formellement cette notion. Bien entendu, d'autres définitions de dépendances peuvent être utilisées sans affecter notre approche générale de calcul de la réputation symbolique.

Définition 1 *Graphe de dépendance des services : Soit* $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ *l'espace de services contenant tous les services web. On définit le graphe de dépendance des services comme un graphe orienté* $G = (\mathcal{S}, \mathcal{V})$ *tel que :* $\mathcal{V} = \{(s_i, s_j) \in S \times S, \exists f_i \in s_i, \exists f_j \in s_j : f_i \rightarrow f_j\}$. *Où* $f \in s$ *signifie qu'un service* s *offre l'opération* f.

Définition 2 *Dépendances des opérations : Soient* $\mathcal{F} = \{f_1, f_2, ..., f_n\}$ *l'ensemble des opérations offertes par tous les services web,* $In(f_i), Out(f_i)$ *représentent respectivement les paramètres d'entrée et de sortie de l'opération* f_i, $\alpha \in [0, 1]$ *est un seuil donné et* sim *la fonction de similarité. L'opération* f_j *dépend de l'opération* f_i, *noté* $f_i \rightarrow f_j$, *si et seulement si* $\forall p \in In(f_j), \exists q \in Out(f_i) : sim(p, q) \geq \alpha$.
Où $sim(p, q) = 1 - NGD(p, q)$ *est le facteur de similarité entre deux termes* p *et* q *calculé en utilisant NDG (Normalized Google Distance) (Cilibrasi et Vitányi, 2007) comme mesure de distance entre deux termes.*

Notre modèle de réputation symbolique est basé sur les marches aléatoires (Lovász, 1993) dans le graphe de dépendances entre les services web. Une marche aléatoire est une séquence de noeuds dans le graphe construit par le procédé suivant : sélectionner un noeud de départ de manière aléatoire dans le graphe, puis passer à une sélection au hasard d'un voisin de ce noeud et ainsi de suite. Les analyses de la marche aléatoire ont été appliquées à différents domaines (Lovász, 1993). Plus particulièrement, un modèle formel a été proposé dans (Rafiei et Mendelzon, 2000) pour permettre le calcul de la réputation d'une page web en utilisant la structure de lien hypertexte. Nous avons adapté ce modèle pour calculer la réputation symbolique de services web. Nous considérons le graphe de dépendances construit précédemment. La probabilité de visiter un service s pour le terme t à l'étape n, notée $P^n(s, t)$ est définie dans l'équation 1.

$$P^n(s,t) = (1-d) \sum_{q \rightarrow s} \frac{P^{n-1}(q,t)}{O(q)} + \begin{cases} \frac{d}{N_t} & \text{Si } t \in RV(q) \\ 0 & \text{Sinon} \end{cases} \quad (1)$$

La probabilité précédente est la base de l'algorithme qui calcule la réputation symbolique.

Algorithme 1 Calcul de la réputation symbolique

ENTRÉES : **S** (l'ensemble des services web), **k** (nombre maximum d'itérations), **d** (saut du marcheur aléatoire), **s** (un service).
SORTIES : $RS(s)$ (Réputation symbolique du service s).

$RS(s) \leftarrow \emptyset$
$RV(s) \leftarrow getVectorialRepresentation(s)$
pour tout $t \in RV(s)$ **faire** P(s,t) = $\frac{d}{N_t}$ **fin pour**
pour $l = 1, .., k$ **faire**
  si $l < k$ **alors** $d^{'} = d$ **sinon** $d^{'} = 1$ **fin si**
  pour chaque chemin $q_l \rightarrow ... \rightarrow q_1 \rightarrow s$ de longueur l et chaque terme t in $RV(q_l)$ **faire**
    $P(s,t) = 0$ si le terme t n'a pas été vu avant
    $P(s,t) = P(s,t) + ((1-d)^l / \prod_{i=1}^{l} O(q_i)(d^{'}/N_t)$
  fin pour
fin pour
pour tout terme t avec $P(s,t) > \frac{1}{N_t}$ **faire**
  $RS(s) = RS(s) \cup t$
fin pour
retourner $RS(s)$

En effet, supposons qu'avec la probabilité d le marcheur aléatoire passe uniformément à un service choisi au hasard parmi l'ensemble des services qui contiennent le terme t. Dans ce contexte, la probabilité qu'un marcheur aléatoire visite un service avec un saut aléatoire est $\frac{d}{N_t}$ (N_t désigne le nombre total de services dont la représentation vectorielle $RV(q)$ contient le terme t), 0 sinon. En outre, la probabilité que le marcheur visite un service s à l'étape n, après en avoir visité un parmi ces parents q est $\frac{1-d}{O(q)} P^{n-1}(q,t)$ où $P^{n-1}(q,t)$ désigne la probabilité que le marcheur aléatoire visite un service q pour un terme t à l'étape $n-1$ et $O(q)$ désigne le nombre de services parents de q dans le graphe de dépendance. L'algorithme 1 donne des détails de calcul de la réputation symbolique pour un service donné. Dans cet algorithme, $RV(s)$ correspond à la représentation vectorielle du service web s qui a pu être calculée avec l'une des méthodes introduites précédemment.

4 Système de découverte et de recommandation de services web

L'algorithme de découverte que nous proposons (cf. algorithme 2) se base sur les trois fonctions suivantes : appariement (*matching*), classement (*ranking*) et sélection. Il utilise les exigences du client : les besoins fonctionnels (la description textuelle du service web souhaité), et le score de la réputation numérique. Comme nous l'avons évoqué dans la section 3.3, cette réputation quantitative se base sur la rétroaction des consommateurs de services. Le processus d'appariement se base sur la représentation vectorielle des services web qui a pu être calculée avec l'une des méthodes introduites précédemment *REI* et *DT*. Le système de découverte se résume comme suit. Etant donné une requête client, il recherche les services qui répondent aux besoins fonctionnels du client. Si plusieurs services répondant aux besoins fonctionnels

sont retournés et si le score de la réputation numérique est spécifié, le système sélectionne les meilleurs services web et les retourne au client.

Nous proposons également un algorithme pour la recommandation de services basée sur la réputation symbolique (cf. algorithme 3). L'idée est d'enrichir les résultats retournés lors de la découverte des services. Pour chaque service découvert, l'algorithme recherche l'ensemble des services web à recommander. Les résultats ne se réduisent alors pas aux services qui correspondent à la requête, mais contiennent également les services recommandés pour chaque service découvert. Les étapes de l'algorithme sont les suivantes. La réputation symbolique $RS(s)$ pour chaque service découvert s est récupérée. Ensuite, l'algorithme trouve tous les services q qui ont une représentation vectorielle $RV(q)$ similaire à $RS(s)$. Remarquons que α désigne un seuil donné. Enfin, les services trouvés sont classés en fonction de leur score de réputation numérique et la similarité entre $RS(s)$ et $RV(q)$. Seuls les meilleurs services en fonction du résultat de ce classement sont recommandés.

Algorithme 2 Découverte et séléction de services web

ENTRÉES : **funReq** : Description textuelle, **repReq** : score de la réputation, **nbMax** : Nombre maximum de services à retourner.
SORTIES : select : la liste de services découverts.
1: $funMatch \leftarrow$ Matching(funReq)
2: $sRank \leftarrow$ Ranking(funMatch,repReq)
3: $select \leftarrow$ Select(sRank,nbMax)
4: **retourner** $select$

Algorithme 3 Recommandation de services web basée sur la réputation symbolique

ENTRÉES : **dS** : services découverts.
SORTIES : **L** : liste des services recommandés.
1: $L = \emptyset$
2: $RS(s)$ = getSymbolicReputation(s)
3: **pour** $q \in dS$ **faire**
4: $RV(q)$ = getVectorialRepresentation(q)
5: **si** $(\|RS(s) \cap RV(q)\| \geq \alpha)$ **alors**
6: L = $L \cup q$
7: **fin si**
8: **fin pour**
9: L $\leftarrow$ Ranking(L)
10: L $\leftarrow$ SelectBest(L)
11: **retourner** L

5 Expérimentations

Nous avons collecté 993 services web réels sur *service-finder.eu* [6] qui a l'avantage de classer les services en utilisant une ontologie (Funk et Bontcheva, 2010). Pour chaque service collecté, nous disposons donc de sa catégorie. Les catégories considérées sont : "Weather",

6. http ://www.service-finder.eu/

"SMS", "Currency Exchange", et "Jobs". Remarquons que d'autres sources de services web, à savoir *WebservicesX.net*[7], *xMethods.net*[8] et *seekda.com*[9] ont été envisagées mais elles ne permettent pas la classification en catégories.

Afin d'évaluer l'intérêt des représentations introduites précédemment (cf. section 3), nous avons procédé de la façon suivante :

- Pour chaque catégorie c, générer une requête req_c de découverte,
- Pour chaque requête req_c et pour chaque représentation rep, calculer les services web découverts $Q_{c,rep}$ avec l'algorithme 2,
- Calculer la précision et le rappel pour chaque catégorie et chaque représentation,
- Calculer la précision globale et le rappel global pour chaque représentation.

La *précision* (cf. équation 2a) représente le pourcentage de bonnes réponses parmi les services web trouvés ($Q_{c,rep}$) par notre système de découverte pour une requête. W_c représente l'ensemble des services web de la catégorie c. Le *rappel* (cf. équation 2b) représente, pour une requête, le pourcentage de services web pertinents trouvés par notre système parmi la totalité des services web de cette catégorie.

$$(a)Precision_{c,rep} = \frac{\|Q_{c,rep} \bigcap W_c\|}{\|Q_{c,rep}\|} * 100; (b)Rappel_{c,rep} = \frac{\|Q_{c,rep} \bigcap W_c\|}{\|W_c\|} * 100 \qquad (2)$$

Pour ces expérimentations, le calcul de la réputation symbolique (cf. algorithme 1) d'un service a été réalisé avec l'union des représentations vectorielles *DT* et *REI* obtenues pour ce service.

La table 1 regroupe les résultats, en termes de précision (*P*) et de rappel (*R*), obtenus pour chaque représentation et pour chaque catégorie de services web. D'une façon globale, la représentation *REI* est la meilleure. Ensuite nous avons *DT* et en dernier *RS*. Remarquons que le rappel global est moyen quelque soit la représentation. La représentation *DT* a obtenu des résultats moyens pour la catégorie *"Weather"*. *REI* est la meilleure représentation pour la catégorie *"Weather"*. Pour les autres catégories, nous remarquons que seule *REI* a obtenu des valeurs moyennes. Les autres représentations se sont écroulées. Nous pouvons dire d'une façon générale que REI est plus robuste que les autres représentations, au travers ces différentes catégories. En ce qui concerne la représentation *RS*, elle a obtenu des résultats moyens pour *"Weather"*. Notons que cette catégorie a obtenue de bons résultats avec chaque représentation. Nous observons que *RS* n'est pas adaptée à la découverte de services web car elle a des résultats très faibles pour toutes les catégories (e.g. *"SMS"*, *"Currency"* et *"Jobs"*) sauf pour *"Weather"*. Cela s'explique par le fait que pour un service donné, *RS* ne représente pas la description du service lui-même, mais représente la description de la relation du service avec les autres. Si ces autres services ne reflètent pas suffisamment la fonction et le but du service, alors sa découverte est compromise.

Etant donné nos observations sur *RS*, nous avons procédé à des expérimentations sur la recommandation de services web en utilisant la représentation RS. La table 2 présente un exemple de recommandations pour un service résultat d'une requête effectuée avec la catégorie *"Weather"*. Nous avons évalué les services recommandés manuellement et les premiers résultats sont très encourageants. Les services recommandés appartiennent, en effet, à la même catégorie que le service initial.

7. http ://www.webservicex.net/ws/default.aspx
8. http ://www.xmethods.net/ve2/index.po
9. http ://www.seekda.com

	DT		REI		RS	
Catégories	P%	R%	P%	R%	P%	R%
Weather	68,97	57,14	81,82	51,43	68,97	57,14
SMS	48,48	80,00	55,56	50,00	12,12	80,00
Currency	16,67	16,67	58,33	77,78	8,51	22,22
Jobs	04,35	08,33	57,14	33,33	6,67	41.67
Moyenne	34,61	40,53	63,21	53,13	24,06	50,27

TAB. 1 – *Évaluation des différentes représentations pour la découverte.*

Service découvert : http ://www.deeptraining.com/webservices/weather.asmx ?wsdl
Services recommandés :(1) http ://ws.cdyne.com/WeatherWS/Weather.asmx ?wsdl
(2)http ://rightactionscript.com/webservices/nusoap/server.php ?wsdl
(3)http ://www.deeptraining.com/webservices/weather.asmx ?wsdl
(4)http ://ws.soatrader.com/weather.gov/0.1/ndfdXML ?wsdl
(5)http ://ws.serviceobjects.com/fw/FastWeather.asmx ?wsdl

TAB. 2 – *Exemple de recommandations de services de la catégorie "Weather".*

6 Conclusion

Dans cet article, nous avons proposé deux nouvelles méthodes de calcul de représentations des services web. La première est basée sur la reconnaissance et l'extraction d'information à partir de descriptions de service web pour ne garder que les parties les plus importantes. La seconde représentation, la réputation symbolique, est plus contextuelle car elle ne considère pas le service lui-même, mais les services en liaison avec lui. Des algorithmes de découverte et de recommandations de services web, se basant sur ces représentations, ont aussi été proposés.

Dans les expérimentations sur des services web réels, nous avons confronté nos représentations avec une méthode traditionnelle de représentation. Nous avons alors observé les points suivants. La représentation classique *DT* ne produit pas des résultats corrects. La représentation basée sur l'extraction d'information (*REI*) est, quant à elle, plus robuste au travers toutes les catégories. Enfin, la réputation symbolique (*RS*) n'est pas appropriée pour la découverte et parait plus adaptée à la recommandation de services web.

Dans le futur, nous approfondirons nos travaux sur la réputation symbolique pour la recommandation. Nous augmenterons le nombre de services web réels utilisés pour les expérimentations. Enfin, nous rendrons disponible en ligne le portail correspondant à l'implantation complète de notre système.

Références

Bilhaut, F. et A. Widlöcher (2006). LinguaStream : An Integrated Environment for Computational Linguistics Experimentation. In *11th Conf. ECACL*, pp. 95–98.

Charnois, T., N. Durand, et J. Kléma (2006). Automated Information Extraction from Gene Summaries. In *DTMIB'06*, Berlin, Germany, pp. 4–15.

Cilibrasi, R. L. et P. M. Vitányi (2007). The Google Similarity Distance. *IEEE TKDE 19*(3), 370–383.

Dong, X., A. Halevy, J. Madhavan, E. Nemes, et J. Zhang (2004). Similarity Search for Web Services. In *VLDB'04*, Toronto, Canada, pp. 372–383.

Elgazzar, K., A. E. Hassan, et P. Martin (2010). Clustering WSDL Documents to Bootstrap the Discovery of Web Services. In *ICWS*, Miami, Florida, USA, pp. 147–154.

Funk, A. et K. Bontcheva (2010). Ontology-Based Categorization of Web Services with Machine Learning. In *LREC*, Valletta, Malta.

Goldbeck, J. et J. Hendler (2004). Inferring Reputation on the Semantic Web. In *WWW*.

Hashemian, S. V. et F. Mavaddat (2005). A Graph-Based Approach to Web Services Composition. In *SAINT*, pp. 183–189.

Lovász, L. (1993). Random Walks on Graphs : A Survey. *Bolyai Society, Mathematical Studies 2*, 1–46.

Majithia, S., A. S. Ali, O. F. Rana, et D. W. Walker (2004). Reputation-Based Semantic Service Discovery. In *WETICE)*, Washington, DC, USA, pp. 297–302.

Malik, Z., I. Akbar, et A. Bouguettaya (2009). Web Services Reputation Assessment Using a Hidden Markov Model. In *ICSOC)*, Stockholm, Sweden, pp. 576–591.

Maximilien, E. M. et M. P. Singh (2002). Reputation and Endorsement for Web Services. *ACM SIGecom Exchanges 3*(1), 24–31.

Maximilien, E. M. et M. P. Singh (2004). Toward autonomic web services trust and selection. In *ICSOC*, pp. 212–221.

Nepal, S., Z. Malik, et A. Bouguettaya (2011). Reputation management for composite services in service-oriented systems. *Int. J. Web Service Res. 8*(2), 29–52.

Omer, A. M. et A. Schill (2009). Dependency Based Automatic Service Composition Using Directed Graph. In *NWESP)*, pp. 76–81.

Petrova-Antonova, D. et S. Ilieva (2009). Towards a Unifying View of QoS-Enhanced Web Service Description and Discovery Approaches. In *YR-SOC)*, pp. 99–113.

Porter, M. (1980). An Algorithm for Suffix Stripping. *Program 14*(3), 130–137.

Rafiei, D. et A. O. Mendelzon (2000). What do the Neighbours Think ? Computing Web Page Reputations. *IEEE DEB 23*(3), 9–16.

W3C (2004). Web services architecture. Technical report, W3C Working Group Note 11 February 2004. Retrieved April 30, 2006 from http ://www.w3.org/TR/ws-arch.

Xu, Z., P. Martin, W. Powley, et F. Zulkernine (2007). Reputation-Enhanced QoS-based Web Services Discovery. In *ICWS*, pp. 249–256.

Yan, J. et J. Piao (2008). Towards QoS-Based Web Services Discovery. In *ICSOC*, pp. 200–210.

Summary

This paper shows that the problem of representation of web services is crucial and analyzes the various factors that influences it. It presents the traditional representation of web services and proposes two new representations. The first one comes from the natural language processing domain and is based on rules to tag web service descriptions and to extract significant information. The second method, called symbolic reputation, is computed from the relationships between web services. The impact of the use of these representations on web service discovery and recommendation is studied and discussed in the experiments using real world web services.