A Framework for Resource Annotation and Classification in Bioinformatics

Nadia Yacoubi Ayadi[†], Malika Charrad[†], Soumaya Amdouni[‡] and Mohamed Ben ahmed[†]

[†] National School of Computer Science,
University of Manouba, 2010 Tunisia
nadia.yacoubi@asu.edu, malika.charrad@riadi.rnu.tn,
mohamed.benahmed@riadi.rnu.tn
[‡] High Institute of Management, Bardo City, Tunisia

Abstract. Semantic annotation is commonly recognized as one of the cornerstones of the semantic Web. In the context of Web services, semantic annotations can support effective and efficient discovery of services, and guide their composition into workflows. Because semantic annotation is a time consuming and expensive task, (semi-)automatic approaches for semantic annotation extraction are required. In this paper, we propose a semi-automatic extraction approach of lightweight semantic annotations from textual description of Web services. In contrast with most of the existing semi-automatic approaches for semantic annotations of Web services which rely on a predefined domain ontology, we investigate the use of NLP techniques to derive service properties given a corpus of textual description of bioinformatics services. We evaluate the performance of the annotation extraction method and the importance of lightweight annotations to classify bioinformatics Web services in order to bootstrap the service discovery process. Our framework relies an unsupervised clustering approach based on a simultaneous clustering algorithm that enables to determine biclusters of Web services and semantic annotations highly correlated.

Keywords: Semantic Annotation, Semantic Web Service, Block Clustering, Bioinformatics

1 Introduction

During the last decade, semantic Web services (SWS) [20] technology have been proposed and investigated to support effective and efficient service discovery, composition and invocation by machines. Despite the appealing characteristics of semantic Web services principles, their uptake on a Web-scale has been significantly less prominent than initially anticipated [21]. In fact, research on semantic Web services has mostly focused on devising domain-independent Web service description ontologies such as OWL-S [19] and WSMO [22]. Semantic Annotations for WSDL (SAWSDL) [15] adopts a bottom-up approach by adding semantics to existing Web service standards through mapping syntactic definitions to

a set of ontological concepts. All of these approaches rely on a pre-determined domain ontology to explicit service semantics. Reasoning tasks performed with semantic Web service descriptions is mainly conditioned by the quality of this domain ontology [4]. The existence of a domain ontology to capture domain knowledge in an explicit and formal way is crucial. In several fields, many domain ontologies have been developed for several purposes. The complexity of reasoning tasks increases when semantic service descriptions are generated by means of several domain ontologies. In the bioinformatics field, the OBO foundary[1] lists around 60 ontologies for life sciences including molecular biology, anatomy, biochemistry, environment, neuroscience, etc. (for a survey, see [24]). None of these ontologies is suitable to annotate bioinformatics Web services; although, they are *rich in semantics* but not *enough generic* to capture high-level concepts and their semantic relationships.

In this paper, we propose a *bottom-up* approach to extract domain-dependant lightweight semantic annotation from textual description of Web services. Such annotations of Web services aims to capture static (i.e., domain concepts) and procedural knowledge (i.e., tasks) of a domain. Despite their importance, few domain ontologies exist for the purpose of Web services annotation, and thus, building such ontologies is a challenging task. Natural language documentations of Web services are short textual descriptions intended to close the "*semantic gap*" between low-level technical features of Web services (e.g., data types, port types, or data formats) and the high-level, meaning-bearing features a user is interested in and refers to when discovering a Web service. Hence, our semi-automatic approach combines different extraction patterns to generate lighweight annotations describing service properties such as inputs, outputs, or functionnalities. We notice that our extraction method provides a good starting point for ontology building.

Therefore, we rely on a simultaneous clustering algorithm, namely CROKI2 [13], to identify clusters (groups) of services that are described by a specific subset of highly correlated annotations. Simultaneous clustering step has two benefits. Firsly, clustering Web services based on semantic annotations would greatly boost the ability of Web services search engines to select suitable services given a discovery query. Secondly, it enables to detect implicit associations (relationships) between highly correlated annotations which is crucial in an ontology building process. In fact, the co-occurrence of a subset of annotations within a subset of Web services reflects implicit relationships that could be taxonomic or non taxonomic between these annotations. To the best of our knowledge, no approach was developed using block-clustering, however, most of the approaches enables either annotations clustering [16, 1] or services clustering [17, 12].

The paper is organized as follows. The section 2 reviews related work conducted in the fields of automatic annotation of Web services and block clustering. Section 3 presents our framework for semantic annotation and clustering of Web services. In the section 4, we present and discuss the results of our experimentations. Section 5 concludes the paper and outlines our future work.

[1] http://www.obofoundry.org/

2 Related Work

2.1 Semantic annotation learning for Semantic Web services

Converting an existing Web service into a semantic Web service requires significant effort and must be repeated for each new Web service. We review in this section research work that focus on learning semantic annotations by exploiting textual descriptions, WSDL files or even Web forms. Hess and al. proposes ASSAM (Automated Semantic Annotation with Machine Learning), a semi-automatic WSDL annotator application. ASSAM [14] relies on a pre-determined domain ontology and uses a machine learning algorithm to provide users with suggestions on how to describe the elements in the WSDL file. However, because of the intensive expert user intervention, applicability of such solution for large-scale annotation of web services could be impractical despite of the fact that these solutions tend to provide high-quality annotations. Sabou et al. [23] proposes an automatic extraction method based on Natural Language Processing (NLP). Experimentations was conducted in the bioinformatics field by learning an ontology from the documentation of Web services in the context of the ^{my}Grid project. The evaluation of the extracted ontology shows that the approach is a helpful tool to support process of building domain ontologies for Web services. Our approach relies on [23]'s approach by using also NLP processing techniques to generate semantic annotations of Web services.

Also, within the bioinformatics space, Afzal et al. [2] developed a text mining approach based on literature to learn *semantic profile* of bioinformatics resources. The approach identifies a set of semantic classes of descriptors that could be attached to a bioinformatics resource: *data*, *data resource*, *task*, and *algorithm*. The instances of these classes were collected by harvesting a corpus of scientific papers along with related sentences containing the resource name. However, the case study conducted in [2] shows that the coverage broad of the ^{my}Grid ontology used as annotation support is partially limited especially to capture functional service descriptions. The quality of extracted descriptors was only measured from the curator's perspective view which is not accurate in the semantic Web context where Web services are supposed to be discovered and composed by agents.

Ambite and al. [3] present an approach to automatically discover and create semantic Web services. The idea behind their approach is to start with a set of known sources and the corresponding semantic descriptions and then discover similar sources, extract the source data, build semantic descriptions of the sources, and then turn them into semantic Web services. Authors implemented the Deimos system and evaluated it across five domains. In contrast to our work, the goal of Deimos is to build a semantic description that is sufficiently detailed to support automatic retrieval and composition. Our work aims to generate lightweight annotations useful to classify Web services and bootstrap the service discovery process in the bioinformatics field.

2.2 Web service Clustering

With the expectable growth of the number of available Web services and service repositories, the need for mechanisms that enable the automatic organization and discovery of services becomes increasingly important. In this context, most of the existing research rely on a one-way clustering, either annotations clustering [16, 1] or services clustering [12, 17]. When clustering algorithms are used, each service in a given services cluster is described using all annotations. Similarly, each annotation in an annotation cluster characterizes all services. For instance, Based on their approach presented in [2], Afzal and al. propose in [1] to use lexical kernel metrics to identify semantically related networks of resources by computing similarity between annotations. However, the goal of our work is to identify groups of services that are more described by a specific subset of annotations which refers to find biclusters of services and annotations highly correlated in order to bootstrap the service discovery process. We rely on simultaneous clustering which is an approach enabling to find local pattern where a subset of subjects might be similar to each other based on only a subset of attributes. Simultaneous clustering, usually designated by biclustering, co-clustering or block clustering aims to find sub-matrices, which are subgroups of rows and subgroups of columns that exhibit a high correlation. A number of algorithms that perform simultaneous clustering on rows and columns of a matrix have been proposed to date. This type of algorithms has been proposed and used in many fields, such as bioinfomatics [18], Web mining [8] and text mining [6]. Table 1 outlines a comparison between one-way clustering and simultaneous clustering.

Table 1. Comparison between Clustering and Simultaneous clustering

Clustering	**Simultaneous Clustering**
- applied to either the rows or the columns of the data matrix **separately** ⇒ **global model**.	- performs clustering in the two dimensions **simultaneously** ⇒ **local model**.
- produce **clusters** of rows **or** clusters of columns.	seeks **blocks** of rows **and** columns that are interrelated.
- Each subject in a given subject cluster is defined using **all** the variables. Each variable in a variable cluster characterizes **all** subjects.	- Each subject in a bicluster is selected using **only** a subset of the variables and each variable in a bicluster is selected using **only** a subset of the subjects.
- Clusters are **exhaustive**	- The clusters on rows and columns **should not be exclusive** and/or **exhaustive**

3 General Framework

The proposed framework is comprised of two main steps. The first one aims to perform a semi-automatic semantic annotation extraction from Web services textual documentations. Semantic annotations enables to describe service properties

such as functionalities, inputs, outputs, and other domain-dependant features. One particluarity of textual Web service description is that they employ natural language in a specific way. In fact, such texts belong to what was defined as sublanguages [23]. A sublanguage is a specialized form of natural language which is used within a particular domain and characterized by a specialized vocabulary, semantic relations, and syntax (e.g., medical test report). The semantic annotation extraction step exploits the linguistic regularities of a sublanguage to identify semantic service properties. The second step of our approach consists on Web service clustering in terms of semantic annotations. This step allows to discover subgroups (biclusters) of Web services and subgroups of semantic annotations that exhibit a high correlation by applying the CROKI2 algorithm [13]. In following, we present in further details the two steps.

3.1 Semantic Annotation Extraction of Web services

The semantic annotation extraction phase allows to identify two types of knowledge: domain concepts and procedural knowledge describing services tasks. First, a morphosyntactic analysis of textual description of Web services is performed. In this step, a sentence splitter and a tokeniser components are used to extract sentences and basic linguistic entities. Then, a POS (Part-Of-Speech) Tagger is performed to associate to each word (token) a grammatical category and thus distinguish the morphology of various entities. For example, the sentence below, the tagger identify a verb (i.e., *compute*), three nouns (i.e., *structure*, *RNA*, *sequence*), an adjective (i.e., *secondary*), and a preposition (i.e., *for*).

```
compute (VB) Secondary (JJ) Structure (NN) for (Prep) RNA (NN) sequence (NN).
```

We distinguish different types of syntactic patterns depending on the semantic annotation type. Syntactic patterns describe selectional constraints that exploit sublanguages particularities. We distinguish syntactic patterns that allow to extract inputs and outputs of services, services tasks, and domain-dependant features which are strongly related to the bioinformatics domain:

1. **Identifying service tasks is crucial for the service discovery and composition issue.** We observed that, in majority of textual descriptions of Web services, verbs identify the functionnality performed by a Web service. In our work, we consider different classes of verbs which inform on the service task. For example, *VBRetrieval* is the class of verbs that indicates a retrieval process (e.g., *get, retrieve, fetch, search, find, return, query*). A frequently occuring pattern which involves this verbs class and the preposition *from* can be used to easily determine the output and the retrieved resource as described by the following selectional pattern:

$$VBRetrieval <Output> \ from <Source>.$$

Other verb classes were recognized, such as *VBExtraction* which is a class of verbs denoting an extraction process, *VBExtraction*={*extract, scan, identify, locate, analyse*}.

2. **Identifying inputs and outputs of Web services.** Inputs and outputs of Web services denote domain concepts which are generally depicted by nouns in the corpus. However, to get high-quality annotations, we create a list of biological terms comprised by a set of single word terms. When two or more biological concepts are used together, we interpret them as a single biological concept and update the list by adding it, i.e., *gene expression*, *transcription factors*, *protein structure*, *tertiary protein structure*, *amino acid sequence*, *chromosome segment*, etc. We define different heuristics that identify the roles of concepts (input or output) depending on the structure of the sentence. Some extraction patterns are presented in Table 2. Therefore, our extraction patterns identifies cases when several concepts are related via logical operators such as "*and*", "*or*". In this case, the same role is assigned to each concept.

Table 2. Examples of Extraction Patterns identifying inputs and outputs of Web services

Extraction Pattern
accepts|**consumes**|**takes input**|**requires**|**Operates On** % <InputService>
VBRetreival|**build** % <OutputService> **given**|**for** % <InputService>
% **Given** <InputService> %
% **returns** <OutputService> %
% <OutputService> **is returned** %
% **compares** <InputService> to <InputService> %
% **compares** <InputService> **against** %

3. **Identifying domain-dependant features.** We define a set of extraction patterns that focus on bioinformatics-dependant features. For example, we propose patterns to identify data formats (e.g., FASTA, GFF, GIF, etc.) related to inputs/outputs formats. An example of such patterns is described as follows: % **computes** <OutputService> **for** % <InputService> **described with** <dataFormat> %.

3.2 Web services Clustering

We propose to use a simultaneous clustering approach to classify Web services in terms of semantic annotations. Our approach aims to find biclusters of Web services and annotations by applying CROKI2 algorithm [13]. We propose an accelerated version of this algorithm in [7]. The general purpose of a block clustering algorithm is described as follows. Given the data matrix A, with set of rows $X = (X_1, ..., X_n)$ and set of columns $Y = (Y_1, ..., Y_n)$, a_{ij}, $1 \leq i \leq n$ and

$1 \leq j \leq n$ is the value in the data matrix A corresponding to row i and column j. Simultaneous clustering algorithms aim to identify a set of biclusters $B_k(I_k, J_k)$, where I_k is a subset of the rows X and J_k is a subset of the columns Y. I_k rows exhibit similar behavior across J_k columns, or vice versa and every bicluster B_k satisfies some criteria of homogeneity.

Croki2 algorithm. The Croki2 algorithm is applied to the contingency table composed of services and annotations to identify a row partition $P = (P_1, ..., P_K)$ composed of K clusters and a column partition $Q = (Q_1, ..., Q_L)$ composed of L clusters that maximizes $\mathcal{X}^2$ value of the new contingency table (P,Q) obtained by regrouping rows and columns in respectively K and L clusters. Croki2 consists in applying K-means algorithm on rows and on columns alternatively to construct a series of couples of partitions (P^n, Q^n) that optimizes Chi2 value of the new contingency table $T_1(P, Q)$ defined by this expression:

$$T_1(k, l) = \sum_{i \in P_k} \sum_{j \in Q_l} a_{ij}$$

$k \in [1, ..., K]$ and $l \in [1, ..., L]$.
Marginal frequencies in table T1 are :

$$f_{kl} = \sum_{i \in P_k} \sum_{j \in Q_l} f_{ij}$$

$$f_{k.} = \sum_{i \in P_k} f_{i.}$$

$$f_{.l} = \sum_{j \in Q_l} f_{.j}$$

Biclusters validity. The application of Croki2 algorithm leads to an exhaustive enumeration of biclusters. It is possible to select only biclusters satisfying certain criteria such as a user-specified bicluster size, bicluster homogeneity and bicluster relevancy [13].

– Homogeneity H is the inertia conserved by the bicluster divided by the initial inertia.

$$H = B_{kl}/T_{kl}$$

$$T_{kl} = \sum_{i \in P_k} \sum_{j \in Q_l} f_{i.} f_{.j} (f_{ij}/f_{i.} f_{.j} - 1)^2$$

and

$$B_{kl} = g_{k.} g_{.l} (g_{kl}/g_{k.} g_{.l} - 1)^2$$

The value of this ratio is between 0 and 1. A high value of this ratio indicates that the bicluster is homogenous.

– Relevancy R is the inertia conserved by the bicluster divided by the global inertia.

$$R = B_{kl}/B$$

$$B_{kl} = g_{k.}g_{.l}(g_{kl}/g_{k.}g_{.l} - 1)^2$$

$$B = \sum_{k,l} B_{kl}$$

This ratio indicates whether the bicluster is relevant.

4 Experimentations

4.1 Experimental Dataset

Our experimental corpus consists of 100 bioinformatics services descriptions from the biocatalogue[2], a new curated life science Web services repository. The development of Biocatalogue shows the dramatic increase of bioinformatics Web services and tools with 2053 services and 148 providers[3]. Biocatalogue allows users to discover Web services through keyword-based retrieval or category browsing. Annotations manually attached to Web services are either textual descriptions or lists of tags. Tagging Web services with a set of lexical tokens defined by users is not a perfect way to enable an efficient service discovery. Manual resource tagging is an error prone and time consuming task. Figure 1 shows the top-20 tags used on biocatalogue. In total, 951 tags were created by users to describe services. The use of tags to describe Web services raises several issues such as the ambiguity of their significance (e.g., `BioMoby` or `soaplab` in Figure 1), the variability of the spelling for several tags that may refer to the same concept. Finally, the lack of explicit knowledge representations in folksonomies (a set of tags) to express whenever the tag describes for example a service task, service input or output which prevents their use towards a significant resource discovery. In our work, Web services are semantically annotated based on their textual descriptions. Extracted semantic annotations enable to automatically construct a semantic service profile. In following, we evaluate respectively the annotation extraction module and the block clustering algorithm.

4.2 Annotation Extraction Performance

We designed an annotation extraction module using the GATE [10] framework. We used the ANNIE plugin (A Nearly-New IE system) which contains a tokeniser, a gazetteer (system of lexicons), a POS Tagger, a sentence Splitter, and a Named Entity (NE) transducer. The various extraction patterns described in section 3.1. were implemented using JAPE [11], a rich and flexible rule mechanism which is part of the GATE framework. The NE transducer applies JAPE

[2] http://www.biocatalogue.org
[3] Last Access on 22^{th} april 2011

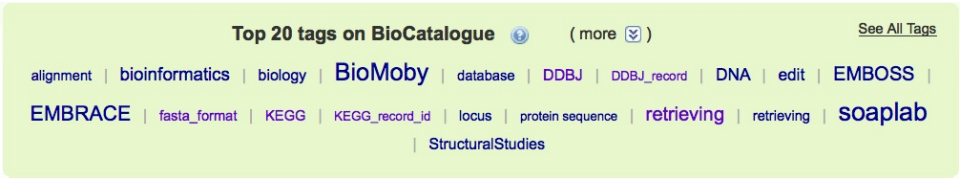


Fig. 1. Top-20 tags in Biocatalogue

rules to input service descriptions in order to generate semantic annotations. Indeed, JAPE (Java Annotation Patterns) engine provides finite state transduction over annotations based on regular expressions. A JAPE grammar consists of a set pattern/action rules. A JAPE rule has a Left-Hand-Side (LHS) and a Right-hand-Side (RHS). The LHS specifies the annotation pattern that may contain regular expression operators (e.g., *, ?, +). The RHS consists of annotation manipulation statements. Annotations matched on the LHS of a rule are referred to on RHS by means of labels that are attached to patten elements. The gazetteer lookup modules, part of the JAPE engine, enable to identify domain concepts in the textual description based on a set of lists of tokens. We have created different lexicons lists containing *bioconcepts*, *service tasks*, *dataformats* and *identifiers* (e.g., EntrezGene_ID, KEGG_ID). Figure 2 illustrates an example of JAPE rule for input service annotation.

We evaluate the results of our experimentations in terms of three metrics: *precision*, *recall* and *F-measure* as depicted in Table 3. The three metrics are calculated as follows.

$$Precision = \frac{Correct + 1/2Partial}{Correct + Spurious + 1/2Partial}$$

$$Recall = \frac{Correct + 1/2Partial}{Correct + Missing + 1/2Partial}$$

$$F - measure = \frac{(\beta^2 + 1)P * R}{\beta^2 R + P}$$

GATE provides an automatic tool for automatic evaluation, named *AnnotationDiff* to compare a set of annotations generated manually and the set of the annotations generated by our extraction method. To measure the performance of the extraction method, we manually identified semantic annotations from the service descriptions corpus. Then, using the *AnnotationDiff* Tool, we compared this set of annotations with the ones that were extracted through extraction patterns.

```
Phase: input
Input: Lookup Token
Options: control = appelt
Rule: input_service

(
({ Token.string == "accept" }
{ Lookup.majorType == bioconcept })
|
({ Token.string == "create" }
{ Lookup.majorType == bioconcept }
{ Token.string == "from" })
|
({ Token.string == "bind" }
{ Lookup.majorType == bioconcept }
{ Token.string == "with" })
|
({ Token.string == "compute" }
{ Lookup.majorType == bioconcept }
{ Token.string == "from" }|{ Token.string == "for" })
|
({ Token.string == "Alignment of" }
{ Lookup.majorType == bioconcept })
|
({ Token.string == "convert" }|{ Token.string == "translate" }
{ Lookup.majorType == dataformat })

)

:label
-->
:label.Input = {rule = "input_service"}
```

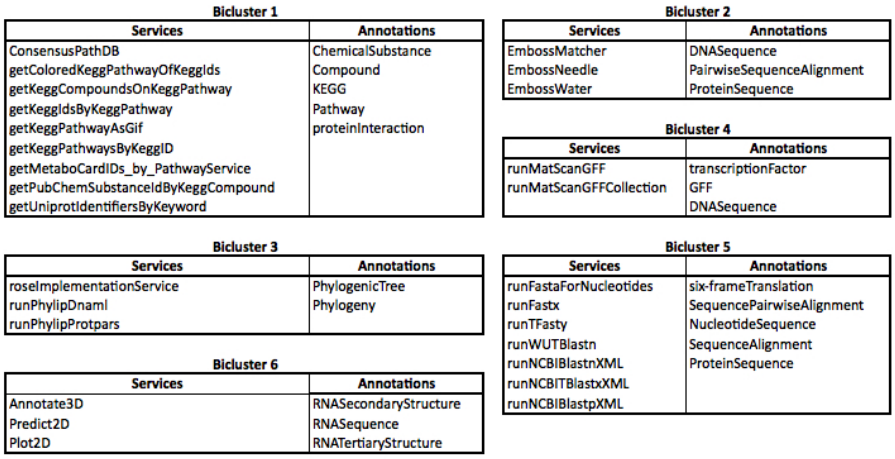
Fig. 2. An example of a JAPE rule

Table 3. Precision, Recall and F-measure

Annotation Type	Precision	Recall	F-measure
Service Name	1	0.83	0.90
Service Input	0.9	0.87	0.88
Service Output	0.9	0.87	0.88
Service Task	0.95	0.97	0.95

4.3 Block Clustering Evaluation

The application of Croki2 algorithm leads to an exhaustive enumeration of biclusters. The data used to evaluate the Croki2 algorithm consists on 98 services and 78 annotations only. The choice of meaningful ones is based on homogeneity and Relevancy as described in the previous section. Given that CROKI2 algorithm uses k-means to cluster rows and columns, the number of clusters needs to be specified by user. Therefore, we extend the use of some validity indices, namely BH [5], proposed initially for one-way clustering to CROKI2 biclustering algorithm [9, 7]. Accelerated CROKI2 algorithm have been implemented in R environment.



Bicluster 1

Services	Annotations
ConsensusPathDB	ChemicalSubstance
getColoredKeggPathwayOfKeggIds	Compound
getKeggCompoundsOnKeggPathway	KEGG
getKeggIdsByKeggPathway	Pathway
getKeggPathwayAsGif	proteinInteraction
getKeggPathwaysByKeggID	
getMetaboCardIDs_by_PathwayService	
getPubChemSubstanceIdByKeggCompound	
getUniprotIdentifiersByKeyword	

Bicluster 2

Services	Annotations
EmbossMatcher	DNASequence
EmbossNeedle	PairwiseSequenceAlignment
EmbossWater	ProteinSequence

Bicluster 3

Services	Annotations
roseImplementationService	PhylogenicTree
runPhylipDnaml	Phylogeny
runPhylipProtpars	

Bicluster 4

Services	Annotations
runMatScanGFF	transcriptionFactor
runMatScanGFFCollection	GFF
	DNASequence

Bicluster 5

Services	Annotations
runFastaForNucleotides	six-frameTranslation
runFastx	SequencePairwiseAlignment
runTFasty	NucleotideSequence
runWUTBlastn	SequenceAlignment
runNCBIBlastnXML	ProteinSequence
runNCBITBlastxXML	
runNCBIBlastpXML	

Bicluster 6

Services	Annotations
Annotate3D	RNASecondaryStructure
Predict2D	RNASequence
Plot2D	RNATertiaryStructure

Fig. 3. Example of biclusters

Table 4. Biclusters and their corresponding Relevance and Homogeneity

Bicluster	Relevancy	Homogeneity
1	6%	37%
2	9%	100%
3	7%	100%
4	8%	100%
5	10%	54%
6	9%	100%

Best biclusters have high values of homogeneity and relevancy (fig.3 and Table 4). For example, biclusters 2, 3, 4 and 6 are the most homogeneous (H=100%)

and bicluster 5 is the most relevant (R=10%). Services and annotations that compose each selected bicluster are highly correlated. Each service in a bicluster is described by a subset of annotations and each annotation in a bicluster describe only services belonging to the same bicluster. All biclusters are significant from the bioinformatics view. For example, bicluster 1 is comprised by services related to pathway and protein interactions, bicluster 2 is composed of services related only to pairwise sequence alignment, in contrast with bicluster 5 which is comprised by services related to pairwise and multiple sequence alignment.

5 Conclusion

This work is part of our ongoing research work. We propose a semi-automatic approach to learn lightweight semantic annotations given a corpus of textual descriptions of Web services. The conducted experimentations show that the approach allows to generate high-quality annotations, mostly because of the fine-grained extraction rules of the approach and the regularity of the sublanguage used to describe Web services in the bioinformatics domain. Our approach consists on a good starting point towards building domain ontologies. As future work, we aim to develop a methodology of domain ontologies building devoted to semantic annotations of Web services by harvesting textual descriptions, WSDL files, and even existing domain ontologies. The main goal of the methodology would be the automatic construction of semantic Web services. Therefore, one motivation of this work is to facilitate the resource discovery within the bioinformatics domain. Thus, we rely on a block clustering algorithm to determine a set of biclusters of services coupled with a set of semantic annotations highly correlated. The results demonstrate the potential of block clustering to model the relatedness between both resources and annotations which is very prominent in the context of service discovery.

References

1. Hammad Afzal, James Eales, Robert Stevens, and Goran Nenadic. Mining semantic networks of bioinformatics e-resources from the literature. In *Semantic Web Applications and Tools for Life Sciences (SWAT4LS) Workshop*, 2009.
2. Hammad Afzal, Robert Stevens, and Goran Nenadic. Mining Semantic Descriptions of Bioinformatics Web Resources from the Literature. In *Proceedings of European Semantic Web Conference*, pages 535–549, 2009.
3. José Luis Ambite, Sirish Darbha, Aman Goel, Craig A. Knoblock, Kristina Lerman, Rahul Parundekar, and Thomas A. Russ. Automatically constructing semantic web services from online sources. In *International Semantic Web Conference*, volume 5823 of *Lecture Notes of Computer Science*, pages 17–32. Springer, 2009.
4. Nadia Yacoubi Ayadi, Zoé Lacroix, and Maria-Esther Vidal. Bionmap: a deductive approach for resource discovery. In *Proceedings of International Conference on Information Integration and Web-based Applications Services (iiWAS'08)*, pages 477–482. ACM, 2008.

5. Frank B. Baker and Lawrence J. Hubert. Measuring the power of hierachical cluster analysis. *Journal of the American Statistical Association*, pages 31–38, 1975.
6. Charles-Edmond Bichot. Co-clustering Documents and Words by Minimizing the Normalized Cut Objective Function. *Journal of Mathematical Modelling and Algorithms (JMMA)*, 9(2):131–147, June 2010.
7. Malika Charrad. *Analyse croisée des sites Web par des méthodes de bipartitionnement*. Editions Universitaires Européenne, 2011.
8. Malika Charrad, Yves Lechevallier, Mohamed Ben Ahmed, and Gilbert Saporta. Block clustering for web pages categorization. In *Proceedings of Intelligent Data Engineering and Automated Learning (IDEAL'2009)*, number 5788 in Lecture Notes in Computer Science, pages 260–267. Springer, 2009.
9. Malika Charrad, Yves Lechevallier, Mohamed Ben Ahmed, and Gilbert Saporta. On the number of clusters in block clustering algorithms. In *Proceedings of FLAIRS Conference*. AAAI Eds, 2010.
10. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.
11. H. Cunningham, D. Maynard, and V. Tablan. JAPE : a java annotation patterns engine (second edition). department of computer science, university of sheffield, 2000.
12. Khalid Elgazzar, Ahmed E. Hassan, and Patrick Martin. Clustering wsdl documents to bootstrap the discovery of web services. In *Proceedings of IEEE International Conference on Web Services (ICWS'10)*, pages 147–154, 2010.
13. G. Govaert. *Classification croisée*. PhD thesis, Paris 6, 1983.
14. Andreas He, Eddie Johnston, and Nicholas Kushmerick. ASSAM: A tool for semi-automatically annotating semantic web services. In *Proceedings of International Semantic Web Conference (ISWC'04)*, volume 3298 of *LNCS*, pages 320–334, 2004.
15. Jacek Kopecky, Tomas Vitvar, Carine Bournez, and Joel Farrell. SAWSDL: Semantic annotations for WSDL and XML schemas. *IEEE Internet Computing*, 11(6):60–67, 2007.
16. Victor Kunin and Christos A. Ouzounis. Clustering the annotation space of proteins. *BMC Bioinformatics*, 6:24, 2005.
17. Jiangang Ma, Yanchun Zhang, and Jing He. Efficiently finding web services using a clustering semantic approach. In *Proceedings of Context enabled source and service selection, integration and adaptation Workshop*, pages 51–58. ACM, 2008.
18. SC. Madeira and AL. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics*, pages 24–45, 2004.
19. David Martin, Mark Burstein, Drew Mcdermott, Sheila Mcilraith, Massimo Paolucci, Katia Sycara, Deborah L. Mcguinness, Evren Sirin, and Naveen Srinivasan. Bringing semantics to web services with OWL-S. *World Wide Web*, 10(3):243–277, 2007.
20. Sheila A. McIlraith, Tran Cao Son, and Honglei Zeng. Semantic web services. *IEEE Intelligent Systems*, 16:46–53, 2001.
21. C. Pedrinaci and J. Domingue. Toward the next wave of services: Linked services for the web of data. *Journal of Universal Computer Science*, 16(13):1694–1719, 2010.
22. Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Rubén Lara, Michael Stollberg, Polleres, Cristina Feier, Cristoph Bussler, and Dieter Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1(1):77–106, 2005.

23. Marta Sabou, Chris Wroe, Carole Goble, and Gilad Mishne. Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In *Proceedings of the 14th international conference on World Wide Web*, pages 190–198. ACM, 2005.
24. Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J. Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J. Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H. Scheuermann, Nigam Shah, Patricia L. Whetzel, and Suzanna Lewis. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11):1251–1255, November 2007.