HAL
open science

# Semantic Divergence based Evaluation of Web Service Communities

Hafida Naim, Mustapha Aznag, Mohamed Quafafou, Nicolas Durand

## To cite this version:

Hafida Naim, Mustapha Aznag, Mohamed Quafafou, Nicolas Durand. Semantic Divergence based Evaluation of Web Service Communities. IEEE 13th IEEE International Conference on Services Computing (SCC 2016), 2016, San Francisco, CA, United States. pp.736-743. hal-01465111

**HAL Id: hal-01465111**

**https://hal.science/hal-01465111**

Submitted on 12 Dec 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.

# Semantic Divergence based Evaluation of Web Service Communities

Hafida NAIM, Mustapha AZNAG, Mohamed QUAFAFOU and Nicolas DURAND
Aix-Marseille University, LSIS CNRS UMR 7296, 13397 Marseille, France.
hafida.naim@etu.univ-amu.fr, {mustapha.aznag,mohamed.quafafou,nicolas.durand}@univ-amu.fr

***Abstract*—The number of community detection algorithms is growing continuously adopting a topological based approach to discover optimal subgraphs or communities. In this paper, we propose a new method combining both topology and semantic to evaluate and rank community detection algorithms. To achieve this goal we consider a probabilistic topic based approach to define a new measure called semantic divergence of communities. Combining this measure with others related to prior knowledge, we compute a score for each algorithm to evaluate the effectiveness of its communities and propose a ranking method. We have evaluated our approach considering communities of real web services.**

***Index Terms*—Community detection algorithms, Web services, Service communities, Topic models, Semantic divergence, Dependency network.**

## I. Introduction

Web services are increasingly being adopted to access data and applications through the Web. In order to ease the process of web services discovery, it is suggested to gather similar web services into groups known as communities or optimal subgraphs [1]. In this regard, the number of community detection algorithms is growing continuously adopting a *topological* based approach, as reported in [2]. Unfortunately there is no generally accepted definition of community; each algorithm makes different assumptions that are consistent with the intuitive concept. Most assume that a graph or network contains a flat set of disjoint communities.

In the literature, several works have been proposed for comparing the resulted partitions in a supervised way [3]. In a social network, for instance, the communities could be groups of people with common interests. Many social networks emerge going far beyond the topological aspect and focus on the semantic dimension. In fact, nodes are not only black boxes but have a specific semantic. It became easy to construct a network on a specific topic where each node represents a user and contains all his tweets related to the topic at hand. In the context of web services, the notion of community is also very important. Classically, a community is a set of web services with similar functionalities grouped together independently of their location, of their maintenance and of their provider [4]. A node of a network represents for example a web service which generally have a textual description. Recently, authors of [5] show that the discovery of web services communities facilitates the service discovery process.

An information network is usually represented by a graph with data entities corresponding to the nodes in the graph and edges indicating the relations among entities [6]. In addition to the topological structure, nodes are usually represented with various types of information that describe their semantics. The task of identifying communities involves discovering groups with common properties, such as similarity among group members or densely connected structure. The main problem with the majority of community detection algorithms is that they completely ignore the *semantics* of nodes and consider just the *topological* nature of the communities [13]. Community detection algorithms are usually informed only by the network structure. So, the produced communities are not sometimes homogeneous but nodes are divided into classes with a higher probability of connections between nodes of the same class than between nodes of different classes.

The work proposed in this paper aims to evaluate the quality and the semantic consistency of detected communities. More precisely, we propose a new method combining both topology and semantic to evaluate and rank community detection algorithms. Different phases of our method are defined as follows (see Section III for more details):

1) Community quality evaluation.
2) Community semantic divergence evaluation.
3) Community detection algorithms ranking.

To evaluate the quality of detected communities, we use the classical measures, i.e. purity and entropy [26]. These measures are widely used to evaluate the performance of supervised/unsupervised learning algorithms, particularly the clustering and community detection algorithms. To evaluate the semantic consistency of detected communities, we define a new measure called community semantic divergence based on the Kullback Leibler (KL) Divergence [8]. We use the probabilistic topic models, more precisely Correlated Topic Models [15] to extract topics from web service descriptions. These extracted topics are then used to compute the semantic divergence for each detected community (see Section III-B). In our context, topic models are used as efficient dimension reduction techniques, which are able to capture semantic relationships between word-topic and topic-service interpreted in terms of probability distributions.

In our experiments, we consider a set of community detection algorithms, described in Section II-D, to evaluate empirically our approach.

The remainder of this paper is organized as follows: Section

II introduces the basic concepts and problem formulation. It lists also a set of community detection algorithms selected to evaluate our approach. Section III presents the proposed method to evaluate and rank community detection algorithms. Section IV is devoted to the experimental evaluation. Finally, the conclusion and future work can be found in Section V.

## II. Preliminaries and problem formulation

### A. Web services representation

Generally, every web service has a WSDL (Web Service Description Language) document that contains the description of the service. The WSDL is an XML-based language, designed according to standards specified by the W3C, that provides a model for describing web services. It describes one or more services as collections of network endpoints, or ports. It provides the specifications necessary to use the web service by describing the communication protocol, the message format required to communicate with the service, the operations that the client can invoke and the service location. To manage efficiently web service descriptions, we extract all features that describe a web service from the WSDL document.

Before representing web services as a *TF-IDF* (Text Frequency and Inverse Frequency) [9] vectors, we need some preprocessing. The objective of this preprocessing is to identify the textual concepts of services, which describe the semantics of their functionalities. There are commonly several steps: *Features extraction*, *Tokenization*, *Tag and stop words removal*, *Word stemming* and *Service Transaction Matrix construction* (see [10] for more details). After identifying all the functional terms, we calculate the frequency of these terms for all web services. We use the Vector Space Model (VSM) technique to represent each web service as a vector of these terms. In fact, it converts service description to vector form in order to facilitate the computational analysis of data. In information retrieval, VSM is identified as the most widely used representation for documents and is a very useful method for analyzing service descriptions. The TF-IDF algorithm [9] is used to represent a dataset of WSDL documents and convert it to VSM form. We use this technique, to represent a services descriptions in the form of *Service Matrix*. In the service matrix, each row represents a WSDL service description, each column represents a word from the whole text corpus (vocabulary) and each entry represents the TF-IDF weight of a word appearing in a WSDL document. TF-IDF gives a weight $w_{ij}$ to every term $j$ in a service description $i$ using the equation: $w_{ij} = tf_{ij}.\log(\frac{n}{n_j})$. Where $tf_{ij}$ is the frequency of term $j$ in WSDL document $i$, $n$ is the total number of WSDL documents in the dataset, and $n_j$ is the number of services that contain term $j$.

### B. Web services graph

Before computing web service communities, we explain the structure of the web service space by modeling relationships between services. Let $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ be the web services space, we construct a semantic similarity graph, $G = (V, E)$, which captures the semantic similarity between different web services. We compute the similarity between the two services using the proximity measure called *Multidimentional Angle* (also known as *Cosine Similarity*); a measure which uses the cosine of the angle between two vectors [11]. The semantic of the service $s_i$ is a vector $d$, which consists of the *TF-IDF* scores extracted from the *Service Transaction Matrix*. The multidimensional angle between a vector $p$ containing the TF-IDF scores of the service $s_p$ and a vector $q$ containing the TF-IDF scores of a service $s_q$ can be calculated using Equation 1:

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

where $t$ is the number of terms representing each web service. Consequently each vertex $v_i$ of our graph $G$ represents the service $s_i$ and two vertices are $s_p$, $s_q$ will be connected if and only if their similarity, i.e., $cos(p, q)$, is larger than a given threshold. After the construction of the web services graph we apply several algorithms to detect communities of web services. Our goal is not to use the semantic of web services and to combine it with the structure of the graph to find homogeneous web services communities like [12]. In fact, we apply several graph structure based detection community algorithms and we use the semantic of web services and their categories to evaluate the resulted communities and to rank the used algorithms.

### C. Global semantic and topics extraction

In Information Retrieval, Probabilistic Topic Models were originally developed and utilized for topic extraction and document modeling. Topics (or latent factors) are a concept introduced by Probabilistic Topic Models [8]. These are a family of generative probabilistic models based on the assumption that documents are generated by a mixture of topics where topics are probability distributions on words. In [10], [14], we investigated the use of three probabilistic topic models PLSA, LDA and CTM [15] to extract latent factors from semantically enriched service descriptions. These latent factors provide a model which represents any web service's description by a vector of terms. The results obtained from comparing the three methods based on PLSA, LDA and CTM showed that the CTM model provides a scalable and interoperable solution for automated service discovery and ranking in large service repositories. The CTM model assumes that the concepts of each service arise from a mixture of topics, each of which is a distribution over the vocabulary. In our work, a topic $t$ is associated with a group of textual concepts (i.e. words) and/or semantic concepts that can appear in service descriptions and can be expressed as a probability distribution over words [16]. In this paper, we use the Correlated Topic Model (CTM) [15] to extract latent factors from web service descriptions. These extracted latent factors are then used to compute the similarity between the services. In our context, topic models are used as efficient dimension reduction techniques, which are able to capture semantic relationships between word-topic and topic-service interpreted in terms of probability distributions.

CTM is a probabilistic topic model that enhances base LDA (Latent Dirichlet Allocation) [7] with modeling of correlations between topics.

### D. Community detection Algorithms

In the literature, there are numerous algorithms that have been proposed for detecting communities by analyzing the structure of graphs focusing on specific graph properties and particular computing method [2]. Such algorithms are generally tested on well known benchmark graphs [17] and their results are summarized in different reviews [3]. However, there are crucial problems to tackle, related to communities, like the stability of communities' structures [18], their dynamic and evolution, etc. Moreover, the analysis of most of community algorithms shows that the objective assessment of the algorithms quality is a complex task and until now, there is no satisfactory answer to the question of choosing the most appropriate algorithm in the context of a given network.

In this work, we therefore consider a set of algorithms, among those who have received the most attention from the scientific community, in order to evaluate empirically our approach.

- Louvain [19] adopts an agglomerative hierarchical method, but it relies on a slightly different greedy optimization process, and includes an additional aggregation step to improve processing on large networks.
- Multistep Greedy (MSG) [20] is a multistep extension of the classical greedy algorithm. It allows the simultaneous merging of several pairs of communities at each iteration. The essential idea is to prevent the premature condensation into few large communities.
- greedy clique expansion (GCE) [21] works by first detecting a set of seeds in a graph G, then expanding these seeds in series by greedily maximizing a local community fitness function, and then finally accepting only those communities that are not nearduplicates of communities that have already been accepted.
- WalkTrap [22] uses a distance measure based on random walks and applies a hierarchical agglomerative clustering approach. It is based on the idea that a small random walk will stay inside the community from where it is originating because there are many links inside and few bridges leading outside.
- Community Overlap Propagation Algorithm (COPRA) [23] is proposed for the first time by Raghavan, Albert and Kumara in [24]. COPRA relies on a label propagation strategy. The information takes the form of a label, and the propagation mechanism relies on a vote between neighbours. Initially, each node is labelled with a unique value. Then, an iterative process takes place, where each node takes the label which is the most spread in its neighbourhood (ties are broken randomly). This process goes on until convergence, i.e. each node has the majority label of its neighbours. Communities are then obtained by considering groups of nodes with the same label. By construction, one node has more neighbours in its community than in the others. This algorithm is faster than most other algorithms.
- CMN [25]: this algorithm is for inferring community structure from network topology which works by greedily optimizing the modularity.

### E. Problem formulation

Let us consider a web service network represented as a graph where each node has not only a black-box but has a specific information represented as textual data.

***Definition** II-E.1:* A **network** is a graph $G = (V, E)$, where $V$ is a set of web services and E is a set of edges. The neighbor of a node $v$, denoted $N_v$, is the set of nodes linked to it.

***Definition** II-E.2:* A **community** $C$ on a network $G$ is a subgraph $G_C = (V_C, E_C)$ such that $V_C \subset V$ and $E_C \subset \{link(x, y) : x \in V_C \wedge y \in N_x\}$ such that a given predefined property $P(C, G)$ is true.

The goal of community detection consists of the formalization of one or several properties $P(C, G)$ and the development of a method to compute such constrained subgraph with a minimum complexity. Several algorithms have been proposed considering different $P(C, G)$ which are generally related to both the structure of graph $G$ and its properties like connex components, Cores, etc. It is also well known in information retrieval that the weight of a given document depends on the other documents which constitute the corpus. By analogy, we associate to a given node in the graph a semantic which is computed considering all other nodes in the graph. This latter semantic is called global semantic and consequently, a node will be generally represented by both a local and global semantics.

***Definition** II-E.3:* The **semantic** of a vertex $v$ is a pair $\langle l(v), g(v)\rangle$ where $l$ and $g$ are two functions returning respectively local and global semantic of $v$.

We will consider in the remainder part of this paper that the local semantic of a node is a vector of words and its global semantic is a mixture of topics. More precisely, we will consider that our nodes belong to predefined categories. they represent the real truth. Consequently, the main tasks are : (1) how to evaluate the compatibility of the computed communities, (2) how to evaluate semantic coherence of a community using topic modeling, and (3) how to evaluate community detection algorithms from the semantic viewpoint.

## III. Proposed method for community detection algorithms ranking

This section proposes a method for communities' evaluation by combining classical measures with a topic modeling on the network: topics shared with different nodes of a community. More specifically, we use the *Entropy* and *Purity* measures [26] to evaluate the quality of the detected communities and the *Kullback Leibler (KL)* Divergence [8] to measure the semantic divergence of these communities.

In the following, we will explain how we calculate the entropy, purity and semantic divergence measures in our

context. Let us assume that we have $q$ classes representing the partitioned web services (service domains), $f$ communities which are produced by each algorithm approach and $n$ is the total number of services.

### A. Entropy and purity measures

The **entropy** measures how the various semantic classes are distributed within each community. Given a particular community $C_j$ of size $n_j$, the *entropy* of this community is defined to be:

$$E(C_j) = -\frac{1}{log(q)} \sum_{i=1}^{q} \frac{n_j^i}{n_j} log(\frac{n_j^i}{n_j}) \quad (2)$$

Where $q$ is the number of domains in the dataset, and $n_j^i$ is the number of services of the *i*th domain that were assigned to the *j*th community. The averaged entropy of the algorithm solution is defined to be the weighted sum of the individual community entropies (Equation 3). In general, smaller entropy values indicate better algorithm solutions.

$$Entropy = \sum_{j=1}^{f} \frac{n_j}{n} E(C_j) \quad (3)$$

The **purity** measure evaluates the coherence of a community. It is the degree to which a community contains services from a single domain. The purity of $C_j$ is formally defined as:

$$P(C_j) = \frac{1}{n_j} max_i(n_j^i) \quad (4)$$

Where $max_i(n_j^i)$ is the number of services that are from the dominant domain in community $C_j$ and $n_j^i$ represents the number of services from community $C_j$ assigned to domain $i$.

The purity gives the fraction of the overall community size that the largest domain of services assigned to that community. Thus, the overall purity is the weighted sum of the individual community purities (Equation 5). In general, larger purity values indicate better algorithm solutions.

$$Purity = \sum_{i=1}^{f} \frac{n_i}{n} P(C_i) \quad (5)$$

The entropy measure is more comprehensive than purity because rather than just considering the number of services in and not in the dominant domain, it considers the overall distribution of all the domains in a given community. Contrary to the purity measure, for an ideal algorithm with services from only a single domain, the entropy of the community will be 0.

### B. Semantic divergence measure

Using the correlated topic model, we have associated to each node $x$ a model $\theta_x$ which is a probability distribution on the topic space. There are several metrics to compute the distance between the semantics of $x$ and $y$. In this paper, we consider the divergence of Kullback Leibler (KL) [8] defined as follows:

$$KLD(\theta_x, \theta_y) = \sum_{i=1}^{T} \theta_x^i \log_2 \frac{\theta_x^i}{\theta_y^i} \quad (6)$$

Where $\theta_x^i = P(i \mid x)$ represents the proprability of the topic $i$ given the node $x$. $KLD$ is not symetric and $KLD = 0$ when $\theta_x^i = \theta_y^i$ for all i.

In our work, the semantic divergence for each edge in the network is measured by symmetric-KL divergence. Indeed, KL-divergence is a non symmetric measure between two probability distributions, while symmetric-KL divergence obtains the symmetry by adding two KL-divergences. For purpose of expediency, we will use the following symetric distance based on KLD:

$$SymmetricKL(\theta_x, \theta_y) = \frac{1}{2}[KLD(\theta_x, \theta_y) + KLD(\theta_y, \theta_x)] \quad (7)$$

### C. Community detection algorithms ranking

Based on the tree measures defined above, we inroduce a new method to evaluate and rank the community detection algorithms. In our context, each algorithm produces several communities which are ranked according to some criterion, i.e., purity/entropy and divergence semantic of a community. Once the web service communities are detected using the selected algorithms described in the Section II-D, we can rank these communities as follows:

1) **Community quality evaluation**: we rank the communities detected by each algorithm considering both the purity and the entropy measures. Each community is then characterized by the product $Purity * (1 - Entropy)$ and the higher the product value, the better the quality of the community is.
2) **Community semantic divergence evaluation**: we compute the Community Semantic Divergence, i.e. denoted CSD, of each community, as the mean of the semantic divergence of their pairs (i.e. linked nodes) as defined in the equation 8. Then, we rank communities of each algorithm according to their semantic divergence. The lower Semantic Divergence Community value, the better the community is: it is semantically coherent.

$$CSD(C_i) = \frac{1}{|E|} \sum_{(x,y) \in E} SymmetricKL(\theta_x, \theta_y) \quad (8)$$

Where $E$ is the set of edge (linked nodes) of the community $C_i$.
3) **Community detection algorithms ranking**: each criteria, i.e. purity/entropy and semantic divergence, allow us to rank communities of each algorithm. Consequently, we consider that these two rankings are coherent: communities which are pure and with low entropy value are also semantically coherent. Hence, we define the score of a community detection algorithm as a distance between its two ranked lists. The distance between two ranked lists is measured using the **Canberra distance** [27]. Smaller canberra distance values indicate better algorithm solution. In fact, the Canberra distance is used to measure the disarray for ranking lists, where rank differences in the top of the lists should be penalized

more than those at the end of the lists. Given two real-valued vectors $l, m \in \mathbb{R}^n$, their Canberra distance is defined as follows:

$$\text{Ca}(l, m) = \sum_{i=1}^{N} \frac{|l_i - m_i|}{|l_i| + |m_i|} \tag{9}$$

## IV. Evaluation

### A. Web services collection

Our experiments are performed out based on real-world web services obtained from the WSDL service retrieval test collection called *SAWSDL-TC3*[1]. The WSDL corpus consists of 1051 semantically annotated WSDL 1.1 based Web services which cover 8 different application domains. Each web service belongs to one out of nine service domains named as: Communication, Education, Economy, Food, Geography, Medical, Travel and Military. Table I lists the number of services from each domain.

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

TABLE I
DOMAINS OF WEB SERVICES

In order to construct the web services graph and to apply the community detection algorithms, we start a pre-processing step of the WSDL corpus. The objective of this first step is to identify the functional terms of services, which describe the semantics of their functionalities. WSDL corpus processing consists of several steps: *Features extraction*, *Tokenization:*, *Tag and stop words removal*, *Word stemming* and *Service Transaction Matrix construction*.

### B. Topic Extraction

To extract topics from web service descriptions, we use the constructed service transaction matrix STM as training data for the CTM model (based on the Blei's implementation[2], which is a C implementation of CTM using Variational EM for Parameter Estimation and Inference). The choice of the number of topics corresponding to the original dataset has an impact on the interpretability of the results. In CTM model, the number of topics must be decided before the training phase. There are several methods to choose the number of topics that lead to best general performance [8]. In computational linguistics, the measure of perplexity has been proposed to assess generalizability of text models. We computed the *perplexity* of a held-out test set to choose the optimal number of topics. A lower perplexity score indicates better generalization performance. Assume we have $M$ web services as a held out dataset $D_{test}$ and each web service $s$ contains $N_d$ word tokens. More formally, the perplexity for a dataset $D_{test}$ is defined by:

$$Perplexity = \exp\left(-\sum_{d=1}^{M}\sum_{n=1}^{N_d} \frac{\log p(w_n|s_d)}{\sum_{d=1}^{M} N_d}\right) \tag{10}$$

Where $p(w_n|s_d)$ is the probability of having word $w_n$ given the $d$-th. service.

Figure 1 shows the perplexity of the held-out data for learned model by varying the number of topics (lower numbers are better). As observed from this figure, the better performance is obtained for $K = 90$ (where $K$ is the number of topics). Thus, in our experiment, the CTM model was trained to generate 90 topics.

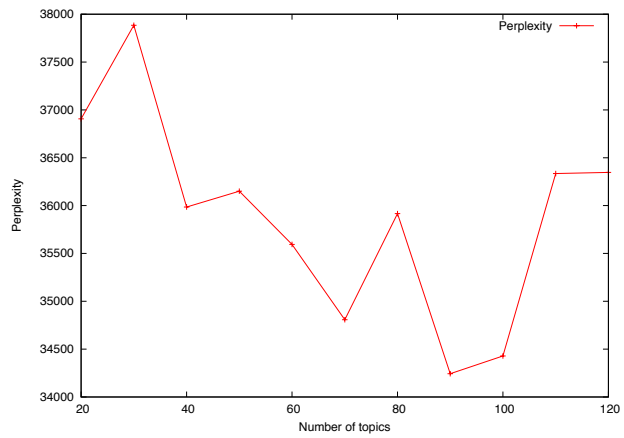


Fig. 1. Perplexity values obtained for learned CTM model.

| # | Algorithm | #Communities | #Communities (Nodes>=2) |
|---|---|---|---|
| 1 | CMN | 3 | 3 |
| 2 | LOUVAIN | 9 | 8 |
| 3 | MSG | 6 | 3 |
| 4 | GCE | 7 | 7 |
| 5 | COPRA | 2 | 2 |
| 6 | WALTRAP | 10 | 6 |

TABLE III
NUMBER OF COMMUNITIES GENERATED BY EACH ALGORITHM

### C. Web service communities detection

After the construction of the web services graph, we have applied the community detection algorithms listed in section II-D. In order to avoid the resulted communities, we have filtered them to avoid very small communities; i.e. communities that contain less than 2 nodes, see Table III.

Please remark that the algorithms CMN and COPRA detect few communities, respectively 2 and 3, and 50% of MSG detected communities are small, this number became 40% for WALKTRAP, and finally, only 1 community is small for the

[1]http://www.semwebcentral.org/projects/sawsdl-tc
[2]http://www.cs.princeton.edu/~blei/ctm-c/index.html

| # | Algorithm | Nbr Communities | Metrics | Generated communities | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
| 1 | CMN | 3 | Purity | 0.68 | 0.36 | 0.85 | - | - | - | - | - |
| | | | Entropy | 0.48 | 0.74 | 0.24 | - | - | - | - | - |
| | | | Purity * (1 - Entropy) | 0.35 | 0.09 | 0.65 | - | - | - | - | - |
| | | | Semantic Divergence | 4.97 | 2.06 | 0.83 | - | - | - | - | - |
| 2 | Louvain | 8 | Purity | 0.52 | 0.49 | 0.93 | 0.6 | 0.71 | 0.88 | 0.94 | 0.41 |
| | | | Entropy | 0.52 | 0.49 | 0.15 | 0.43 | 0.46 | 0.23 | 0.15 | 0.59 |
| | | | Purity * (1 - Entropy) | 0.25 | 0.25 | 0.79 | 0.34 | 0.38 | 0.68 | 0.8 | 0.17 |
| | | | Semantic Divergence | 4.35 | 3.08 | 5.81 | 3.35 | 0.25 | 0.00 | 0.14 | 2.10 |
| 3 | MSG | 3 | Purity | 0.64 | 0.39 | 0.38 | - | - | - | - | - |
| | | | Entropy | 0.55 | 0.71 | 0.6 | - | - | - | - | - |
| | | | Purity * (1 - Entropy) | 0.29 | 0.11 | 0.15 | - | - | - | - | - |
| | | | Semantic Divergence | 5.27 | 3.59 | 2.14 | - | - | - | - | - |
| 4 | GCE | 7 | Purity | 0.56 | 0.32 | 0.79 | 0.68 | 0.63 | 0.93 | 0.98 | - |
| | | | Entropy | 0.55 | 0.7 | 0.35 | 0.36 | 0.36 | 0.14 | 0.04 | - |
| | | | Purity * (1 - Entropy) | 0.25 | 0.1 | 0.51 | 0.44 | 0.4 | 0.8 | 0.94 | - |
| | | | Semantic Divergence | 3.48 | 1.94 | 4.64 | 3.54 | 4.31 | 3.56 | 0.00 | - |
| 5 | COPRA | 2 | Purity | 0.66 | 0.36 | - | - | - | - | - | - |
| | | | Entropy | 0.51 | 0.72 | - | - | - | - | - | - |
| | | | Purity * (1 - Entropy) | 0.32 | 0.10 | - | - | - | - | - | - |
| | | | Semantic Divergence | 4.95 | 2.01 | - | - | - | - | - | - |
| 6 | Walktrap | 6 | Purity | 0.87 | 0.59 | 0.67 | 0.8 | 0.61 | 0.39 | - | - |
| | | | Entropy | 0.19 | 0.49 | 0.4 | 0.27 | 0.38 | 0.67 | - | - |
| | | | Purity * (1 - Entropy) | 0.7 | 0.3 | 0.4 | 0.58 | 0.38 | 0.13 | - | - |
| | | | Semantic Divergence | 1.30 | 3.79 | 4.10 | 5.98 | 3.17 | 1.58 | - | - |

TABLE II
COMMUNITY QUALITY EVALUATION FOR SELECTED ALGORITHMS

Louvain algorithm. According to the number of filtered communities, we divide the used algorithms into two categories: those finding only few filtered communities in contrast to all the other algorithms. Finally, we will consider only the filtered communities during the evaluation process.

### D. Prior knowledge based community quality evaluation

Let us compute the purity and the entropy of all detected communities by each used algorithm. The results are given in Table II which may be summarized follows: (1) the communities detected by CMN, MSG and COPRA have similar entropy (very large communities), but CMN is better than MSG and COPRA if we consider the purity viewpoint, (2) The algorithm Louvain detects communities which are better than those computed by CGE and Walktrap. Consequently, we rank the communities detected by each algorithm considering both the purity and the entropy. As described in the section III-C, each community is then characterized by the product $Purity * (1 - Entropy)$ and the higher the product value, the better the quality of the community is. For example, communities rank of those detected by CMN is C3, C1, C2, whereas those detected by Louvain algorithm are ranked as follows : C7, C3, C6, C5, C4, C2, C1, C8. The overall entropy and purity are calculated and drawn in the Figure 3.

### E. Community semantic divergence evaluation

Now, let us consider the evaluation of detecting communities according to the semantic divergence of their linked nodes. In fact, we compute the semantic divergence of a community as the mean of those of their pairs, i.e., linked nodes. Figure 2 shows the results for the two kinds of

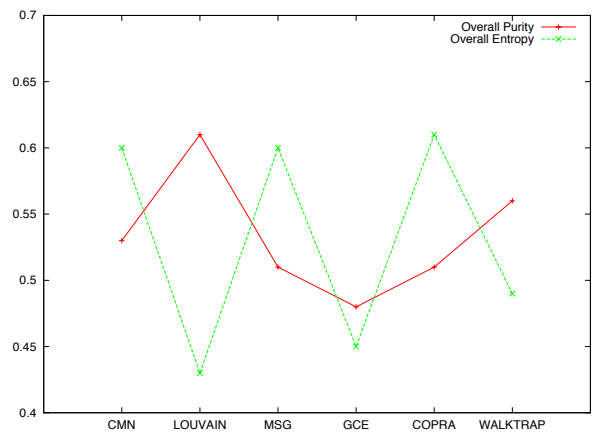


Fig. 3. Overall Purity and Entropy for all algorithms.

considered algorithms giving more statistical details on the semantic divergence measures within a given community. In Figure 2 (Left), detected communities are enough large and are not semantically homogeneous except the very small one, for example the community 3 detected by the CMN algorithm that contains only 33 nodes.

This situation is not the same for an algorithm which detects several communities. In fact, communities must be both enough large and semantically homogeneous. So, an algorithm may detect small and homogeneous communities or large and semantically divergent communities. The obtained values of Community Semantic Divergence measure of each detected

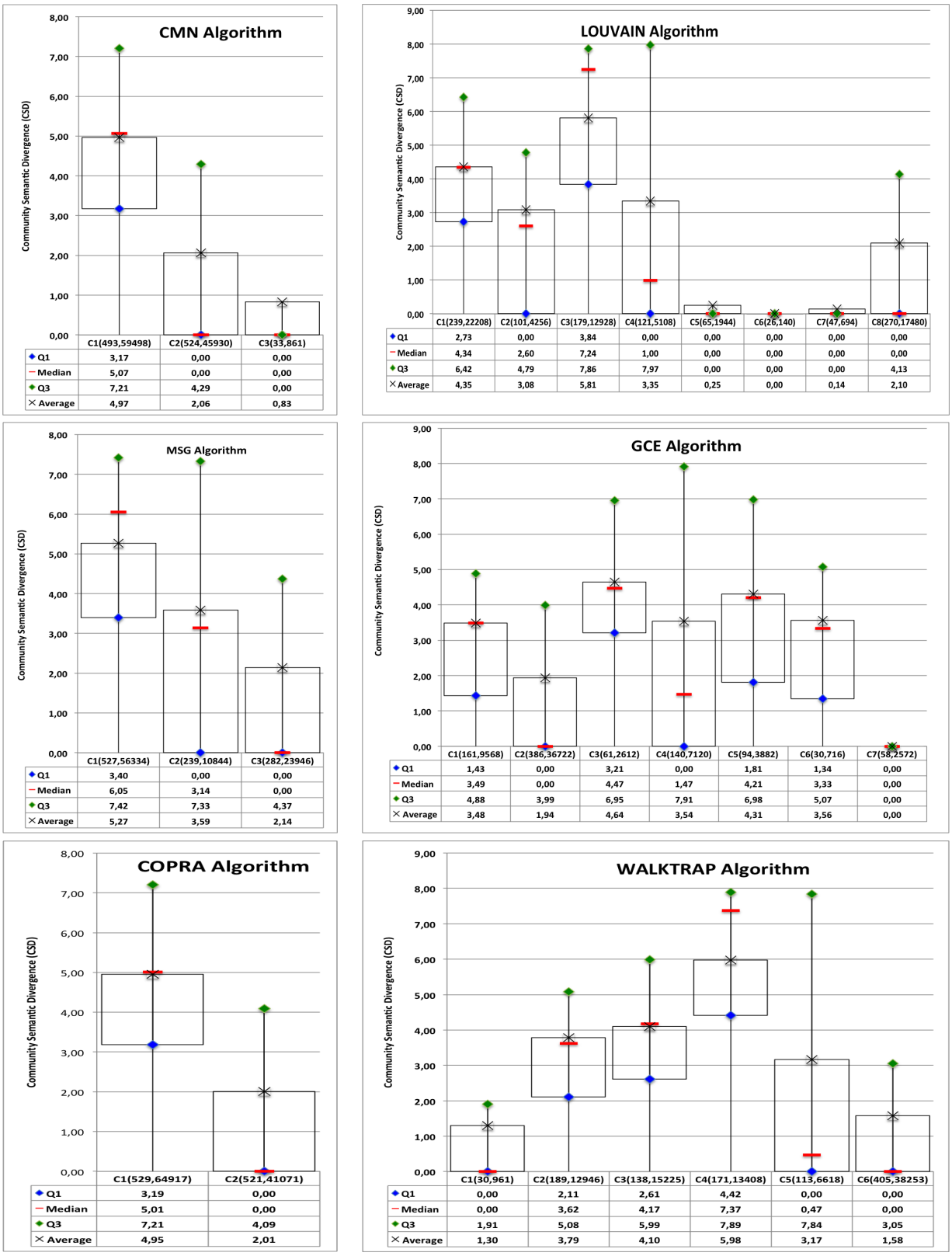


Fig. 2. Semantic divergence - (Left) algorithms detecting few communities (Right) algorithms detecting several communities

community are drawn also in the Table II. The Louvain algorithm detects 3 small and 3 enough large communities, but others are larger and divergent. Note that the community C1 is larger, more dense and less divergent than C3. The other algorithms are balancing between enough large and small communities with different semantic divergence. Similarly to the criteria $Purity * (1 - Entropy)$, we can also use the semantic divergence to rank communities detected by each system. How to score algorithms using these two ranking lists?

### F. Score of algorithms

We have evaluated the community detection algorithms using two criteria. The first, is based on categories or domains of web services. The second one, focuses on the semantic coherence of a community considering probabilistic topic modelling. Each criterion allows us to rank the detected communities for each algorithm. Consequently, we consider that an algorithm is better if the two ranking of its detected communities are compatible: if a community is well classified by the criterion $Purity*(1-Entropy)$ so, it will be also well ranked by the semantic divergence and vice versa. To achieve this goal we use the Canberra distance which is a measure of disarray for ranked lists [27]. Thus, the score of each algorithm is equal to the Canberra distance applied to its two resulted ranking. Of course, this method is useful when the number of detected communities is large because rank differences in top positions need to pay higher penalties than movements in the bottom part of the lists. The score for algorithms that detect more than 3 communities are 4.68 for Louvain, 3.43 for GCE and finally 1.81 for Walktrap. This latter is the best one because even if it does not detect the most homogeneous, but its communities are enough homogeneous and well classified according to the purity and the entropy.

## V. Conclusion

In this paper, we have proposed a new method for community detection algorithms ranking by combining classical measures with a topic modeling. First, we have ranked the detected communities for each algorithm using purity, entropy and semantic divergence measures. A community is considered well ranked if it has a low entropy value and a high purity. In addition, according to the semantic divergence, the lower Semantic Divergence Community value, the better the community is: it is semantically coherent. Our approach allows us to evaluate the semantic homogeneity of the computed communities that are based only on topology. We can also evaluate the semantic coherence of a community using the semantic divergence measure proposed in this paper. We have tested our approach using several community detection algorithms and considering real world web services.

In future work, we will propose a semantic homogeniety-based approach to detect semantic web service communities.

## References


[1] Maamar, Z., Subramanian, S., Thiran, P., Benslimane, D., & Bentahar, J. (2009). Communities of Web Services -Concepts, Architecture, Operation, and Deployment. International Journal of EBusiness Research, 5(4), pp. 1-12. Hershey, PA: , IGI Global Publishing

[2] Fortunato, S. 2010. Community detection in graphs. Phys. Rep. 486, 75-174.

[3] Leskovec, J., Lang, K. J., and Mahoney, M. Empirical comparison of algorithms for network community detection. In Proc. WWW'2010. pp. 631-640.

[4] Cherifi, Chantal and Santucci, Jean François. "Community structure in interaction web service networks.." IJWBC 9 , no. 3 (2013): 392-410.

[5] Yu Q., Manjeer Rege, On Service Community Learning: A Co-clustering Approach. In ICWS'2010.

[6] Liu, L.; Xu, L.; Wang, Z. & Chen, E. (2015), Community Detection Based on Structure and Content: A Content Propagation Perspective., in IEEE ICDM 2015. pp. 271-280.

[7] Blei D., Andrew Y. Ng and Jordan M. I.: Latent dirichlet allocation, J. Mach. Learn. Res. pp. 993-1022 (2003).

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

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

[10] Aznag, M., Quafafou, M. and Jarir, Z.: Correlated Topic Model for Web Services Ranking. In IJACSA, vol. 4, no. 6, pp. 283–291, July 2013.

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

[12] Liu, X., Huang, G., Discovering Homogeneous Web Service Community in the User-Centric Web Environment. IEEE TSC'2009, Vol. 2, No. 2, 2009

[13] Reihanian A, Minaei-Bidgoli B, Alizadeh H (2015) Topic-oriented community detection of rating-based social networks. In ournal of King Saud University - Computer and Information Sciences.

[14] Aznag, M., Quafafou, M., Rochd, El M., and Jarir, Z.: Probabilistic Topic Models for Web Services Clustering and Discovery. In ESOCC, LNCS 8135, pages 19–33, 11 September 2013.

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

[16] Aznag, M., Quafafou, M. and Jarir, Z.: Leveraging Formal Concept Analysis with Topic Correlation for Service Clustering and Discovery. In 21th IEEE ICWS 2014. Alaska, USA.

[17] Lancichinetti, A., Fortunato, S., and Radicchi, F. 2008. Benchmark graphs for testing community detection algorithms. Phys. Rev. E 78, 046110.

[18] Gfeller, D., Chappelier, J.-C., and De Los Rios, P. 2005. Finding instabilities in the com- munity structure of complex networks. Phys. Rev. E 72, 056135.

[19] Blondel, VD., Guillaume, JL. Lambiotte, R., Lefebvre, E., Fast unfolding of communities in large networks. In JSTMTE'2008.

[20] Schuetz, P. and Caflisch, A.: Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. Phys. Rev. E, 77-4, 2008.

[21] Lee, C., Reid, F., McDaid, A., and Hurley, N. 2010. Detecting highly overlapping community structure by greedy clique expansion. In Proc. SNAKDD Workshop. 33-42.

[22] Pons, P. and Latapy, M.: Computing communities in large networks using random walks. J. of Graph Alg. and App. bf, 10, 2004, 284-293.

[23] McDaid, A. and Hurley, N. Detecting highly overlapping communities with model-based overlapping seed expansion. In Proc. ASONAM Conf 2010. pp. 112-119.

[24] U. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. Physical Review E, 76(3), 2007.

[25] Clauset A., Newman, M. E. J., Moore, C.: Finding community structure in very large networks. Physical Review E, vol. 70, Issue 6, 12/2004.

[26] Zhao, Y. and Karypis, G.: Empirical and theoretical comparisons of selected criterion functions for document clustering. In: Machine Learning, 55, pp. 311-331. (2004)

[27] Jurman, G., Riccadonna, S., Visintainer, R., Furlanello, C.: Canberra distance on ranked lists. In: Proceedings of Advances in Ranking NIPS 2009 Workshop, pp. 22-27