Boosting a Network of Semantic Peers (Extended Abstract) ⋆

Stefano Lodi[1], Federica Mandreoli[2], Riccardo Martoglia[2], Wilma Penzo[1], and Simona Sassatelli[2]

[1] DEIS - University of Bologna, Italy
{stefano.lodi, wilma.penzo}@unibo.it
[2] DII - University of Modena e Reggio Emilia, Italy
{federica.mandreoli, riccardo.martoglia, simona.sassatelli}@unimo.it

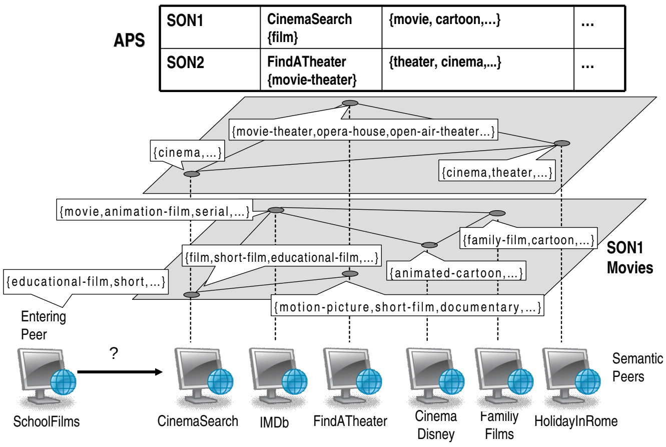
Abstract. In a Peer Data Management System (PDMS), semantic peers connect with each other through semantic mappings between their own schemas. Because of schema heterogeneity, due to peers' autonomy as for data representation, querying a PDMS implies query reformulations across semantic mappings, possibly incurring in a semantic degradation due to the reiterated approximations given by the traversal of long paths. The linkage closeness of semantically similar peers is thus a crucial issue. In this paper we present a strategy for the incremental maintenance of a flexible network organization for PDMSs that clusters together semantically related peers.

1 Motivation and Related Work

In recent years, information sharing has gained much benefit by the large diffusion of Peer-to-Peer systems. On the other hand, in line with the Semantic Web vision, the stronger and stronger need of adding semantic value to the data has emerged. In this view, Peer Data Management Systems (PDMSs) have been introduced as a solution to the problem of large-scale sharing of semantically rich data [6]. In a PDMS, peers are autonomous and heterogeneous data sources, having their own content modeled upon schemas. Because of the absence of common understanding of the vocabulary used at each peer's schema, semantic relationships are established locally between peers, thus implementing a decentralized schema mediation.

One of the main challenges in such a semantically heterogeneous environment is concerned with query processing. A query is routed through the network by means of a sequence of reformulations, according to the semantic mappings encountered in the routing path. Reformulations may lead to semantic approximations, possibly inducing information loss due to the need of traversing incomplete or missing mappings, thus, for a given peer, the linkage closeness to

⋆ The material presented in this paper has been published as [7] in the Proceedings of the 11th International Conference on Extending Database Technology (EDBT 2008). This work is partially supported by the Italian Council co-funded project NeP4B.



APS			
SON1	CinemaSearch {film}	{movie, cartoon,...}	...
SON2	FindATheater {movie-theater}	{theater, cinema,...}	...

APS
{movie-theater,opera-house,open-air-theater...}
{cinema,...}
{cinema,theater,...}
{movie,animation-film,serial,...}
{family-film,cartoon,...}
SON1
Movies
{film,short-film,educational-film,...}
{animated-cartoon,...}
{educational-film,short,...}
{motion-picture,short-film,documentary,...}
Entering
Peer
?
Semantic
Peers
SchoolFilms
CinemaSearch
IMDb
FindATheater
Cinema
Disney
Family
Films
HolidayInRome

Fig. 1. Sample of network organization

semantically similar peers is a crucial issue, leading to the problem of how to boost a network of mappings in a PDMS [6]. This matter has also been evidenced recently by works on Semantic Overlay Networks (SONs) [2–4, 9] for P2P systems, where peers with semantically similar content are clustered together in logical subnetworks. The main goal of a SON in a P2P system is to improve the efficiency of query processing by limiting the number of contacts only to relevant peers. In a PDMS, instead, SON principles substantially aim at reducing semantic degradation during query answering by arranging semantically related peers close to each other. Moreover, semantic heterogeneity makes most of the solutions proposed in the literature inapplicable in a PDMS setting.

The work presented in this paper aims to support the creation and maintenance of a flexible network organization for PDMSs that clusters together heterogeneous peers which are semantically related. In a PDMS which adopts the network organization we propose instead of a random one, it is more likely that closely associated peers are relevant to the same queries thus reducing semantic degradation.

The network we refer to is made up of a set of semantic peers, each represented by a set of concepts describing its main topics of interest. The process leading to the representation of each peer is out of the main scope of this paper. To this end, solutions like the one recently proposed in [10] could be adopted. The network is organized in a set of SONs. Fig. 1 shows a sample of network made up of two SONs concerning cinema-related data. Some peers of the network, such as the Internet Movie Database (IMDb) and the web site HolidayInRome are "monothematic", i.e. they only deal with movies and movie theaters, respectively. Other peers, instead, are concerned with both themes, e.g. FindATheater. Peers are assigned to one or more SONs on the basis of their own concepts. In a PDMS, this operation is a really challenging one because of the peers' autonomy as for data representation. This means that similar or even the same contents in different peers are not guaranteed to be described by the same concepts. Our proposal is to cluster together in the same SON peers with *semantically similar* concepts. Our approach is general, in that semantic similarity can be measured by means of any knowledge-based distance function between concepts, under the only requirement of being a metric, i.e. satisfying the non-negativity, the symmetry, and the triangle inequality properties. In particular, in [7] we experienced several distances which take advantage of the WordNet external knowledge source. As it is described in the following sections, the network evolves incrementally to assimilate entering peers in such a way to assist the peers in the selection of their neighbors in a two-fold fashion: First, in the choice of the semantically closest overlay networks; Then, within each overlay network, in the selection of their own neighbors among their most semantically related peers.

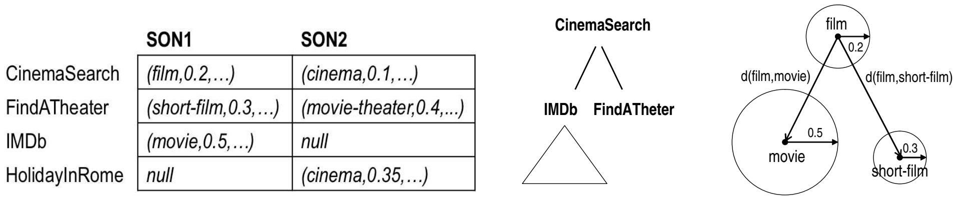
The paper is organized as follows: Section 2 describes the actions a peer performs for selecting the best SONs to join to; Section 3 completes the positioning of a peer in the network by discussing how neighbors are chosen; Section 4 discusses the experimental evaluation and, finally, Section 5 concludes the paper.

2 Selecting the Best SONs

When a new peer joins the system (e.g. SchoolFilms in Fig. 1), it first performs a coarse-grained neighbor selection by accessing the *Access Point Structure (APS)*. This is a structure which maintains cumulative information about the SONs available in the network. It is conceptually centralized, but can be stored in a distributed manner, for instance, by means of a Distributed Hash Table (DHT). The APS ignores the linkage among peers and provides an abstraction of the SONs as *clusters of concepts* (e.g. $film, movie, cartoon...$ etc. for SON1 in Fig. 1). In order for the APS to be a “light” structure which scales to the large, we do not keep all concepts at the APS level and follow an approach similar to the one adopted in [5] for clustering large datasets. For each SON, the APS treats its concepts collectively through a summarized representation called *Semantic Feature (SF)* which expresses the main characteristics of the SON, such as its clustroid (i.e. the centrally located concept according to the adopted distance function between the concepts) with the identifier of the peer it belongs to, some sample concepts, and other properties. Fig. 1 shows a portion of the APS for the reference scenario. The APS contains the SFs of two SONs, SON1 and SON2, whose clustroids are the concepts *film* and *movie-theater* owned by peers CinemaSearch and FindATheater, respectively. In the third column some sample representative concepts for the two SONs are listed.

Each new entering peer exploits the information provided by the APS in order to decide which SONs to join to or whether to form new SONs. As a first step, the peer computes the semantic distances between its own concepts and the clustroids of the SFs in the APS. Then, each peer's concept is associated to the SON whose clustroid is the closest one.[3] After, the peer enters each SON

[3] This complies with classical proposals in the field of incremental clustering [5].



	SON1	SON2
CinemaSearch	*(film,0.2,…)*	*(cinema,0.1,…)*
FindATheater	*(short-film,0.3,…)*	*(movie-theater,0.4,...)*
IMDb	*(movie,0.5,…)*	*null*
HolidayInRome	*null*	*(cinema,0.35,…)*

(a) CinemaSearch's SCI

(b) Tree abstraction of SON1

Fig. 2. CinemaSearch's SCI and tree abstraction of SON1

which is associated to with at least one concept. In our reference scenario in Fig. 1, the concepts of the entering peer SchoolFilms are more concerned with movies than with movie theaters, thus the peer enters SON1. In order to avoid clusters distortions, a threshold T is used: concepts having a distance greater than T are clustered together and the peer originates a new SON for each cluster.

Finally, the evolution of the APS is managed in an incremental fashion to reflect the network changes due to the joining/leaving of peers. The detailed algorithms which implement the above described SON selection mechanism and the APS update process can be found in [7] where the theoretical framework of our work is presented.

3 Locating the Most Desirable Neighbors

Once the entering peer has chosen the SONs to join to, it navigates the link structure within each selected SON with the aim of finding its semantically closest peers. Adopting a broadcast-based approach to search neighbors could imply wasting precious resources. Instead, we propose to exploit a distributed indexing mechanism which maintains at each peer specifically devised indices named *Semantic Clustering Indices (SCIs)*.

Each SCI maintains summarized information about the SONs'concepts available in each direction. In particular, the SCI SCI_P of a peer P is a matrix where each cell $SCI_P[i,j]$ refers to the set of concepts in the j-th SON (SON_j) which are reachable in the sub-SON rooted at the i-th neighbor. Each column j contains non-null values in correspondence of each peer belonging to SON_j. Each cell stores a summarized description similar to a SF (i.e. the clustroid of the sub-SON $SON_{i,j}$, the radius, i.e. the maximum distance between the clustroid and $SON_{i,j}$'s concepts, and other information).

Fig. 2-a shows peer CinemaSearch's SCI. The concept in each cell is the clustroid of the corresponding sub-SON, while the score is the radius. Notice that the first row refers to peer CinemaSearch itself and thus the two cells refer to the sets of concepts through which CinemaSearch joined SON1 and SON2, respectively.

A SCI SCI_P provides an abstraction of the SONs P belongs to as trees. More precisely, if P belongs to SON_j then P is the root node and it has as many children as the number of its neighbors in SON_j, i.e. the number of non-null cells in $SCI_P[*, j]$. Fig. 2-b depicts this tree-based abstraction of SON1 from peer CinemaSearch's point of view: All concepts in the sub-SON rooted at IMDb (resp., FindATheater) are within a semantic distance of 0.5 (resp., 0.3) from the clustroid concept *movie* (resp., *short-film*).

SCIs are used to lighten the neighbor selection process. The objective is to reduce the network load, i.e. the number of accessed peers and the computational effort which is required to each accessed peer. To select neighbors in a SON, the entering peer starts from the clustroid peer and it navigates the link structure by visiting (some of) the peer's immediate neighbors, then their immediate neighbors, and so on. Two neighbor selection policies are supported: 1) a *range-based selection*, where the selected peers are those within a semantic distance bounded by a given threshold t, and 2) a *k-NN selection*, which finds out the k semantically nearest peers.[4]

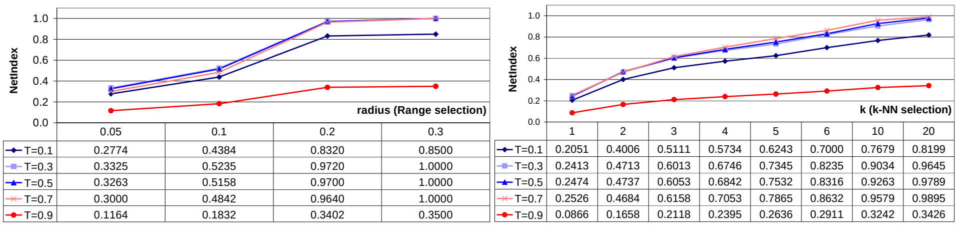
More precisely, an entering peer P_{new} starts the exploration and follows each path which can not be excluded from leading to peers satisfying the (range or k-NN) selection condition. In the range-based selection, for each contacted peer P, the distance between P_{new}'s concepts and P's concepts is computed and, if the threshold condition is satisfied, P is chosen as P_{new}'s neighbor. For k-NN selection, a branch-and-bound technique is applied, based on a priority queue of pointers to active sub-SONs, i.e. subnetworks where the k nearest neighbors of P_{new} can possibly be found, and a k-elements array which at the end of the process contains the k selected neighbors.

For both policies, the information stored at SCI_P is used to *prune out* non-relevant subnetworks and to avoid useless distance computations by exploiting the triangle inequality property of the distance function used. In particular, let $r_{i,j}$ be the radius of sub-SON $SON_{i,j}$ and consider the case of range-based selection. All the sub-SONs in SCI_P whose clustroids are at a distance d from P_{new}'s concepts such that $d > r_{i,j} + t$ can be safely pruned. In fact, this condition guarantees that all concepts in the sub-SON $SON_{i,j}$ have a distance greater than the threshold t. A similar condition can be exploited for k-NN-based selection, where the distance d_k from the current k-th nearest neighbor is used as a dynamic threshold (i.e. the test condition is $d > r_{i,j} + d_k$).

Going back to our reference example, according to the APS in Fig. 1 the entering peer SchoolFilms starts the exploration of SON1 from the clustroid peer CinemaSearch. Then, consider Fig. 2 and let us suppose the peer wants to find its neighbors in a range of 0.2 and that the semantic distances from SchoolFilms' concepts to IMDb's and FindATheater's concepts are 0.4 and 0.8, respectively. Since $0.4 < 0.5 + 0.2$ the sub-SON rooted at IMDb is explored, whereas the one rooted at FindATheater is pruned since $0.8 > 0.3 + 0.2$.

As for the APS, SCIs need to be updated whenever the SONs change because of the joining/leaving of peers. SCIs creation and evolution is managed in an incremental fashion according to a message exchange protocol detailed in [7].

[4] It is worth noting that the topology of the network is heavily influenced by the kind of neighbor selection policy each peer chooses when it joins the network.



	0.05	0.1	0.2	0.3
T=0.1	0.2774	0.4384	0.8320	0.8500
T=0.3	0.3325	0.5235	0.9720	1.0000
T=0.5	0.3263	0.5158	0.9700	1.0000
T=0.7	0.3000	0.4842	0.9640	1.0000
T=0.9	0.1164	0.1832	0.3402	0.3500

(a) Range selection

	1	2	3	4	5	6	10	20
T=0.1	0.2051	0.4006	0.5111	0.5734	0.6243	0.7000	0.7679	0.8199
T=0.3	0.2413	0.4713	0.6013	0.6746	0.7345	0.8235	0.9034	0.9645
T=0.5	0.2474	0.4737	0.6053	0.6842	0.7532	0.8316	0.9263	0.9789
T=0.7	0.2526	0.4684	0.6158	0.7053	0.7865	0.8632	0.9579	0.9895
T=0.9	0.0866	0.1658	0.2118	0.2395	0.2636	0.2911	0.3242	0.3426

(b) k-NN selection

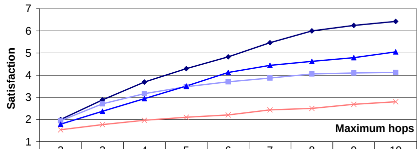
Fig. 3. Effectiveness tests: NetIndex network quality

4 Experimental Evaluation

For our experiments we used the SUNRISE simulation framework [1] through which we generated scenarios corresponding to networks of semantic peers, each with its own schema describing a particular reality. The schemas are derived from real-world data sets, such as the DBLP Computer Society Bibliography and the ACM SIGMOD Record, and enlarged with new schemas created by introducing variations on the original ones. The representation of each peer is derived from its schema by following an approach similar to [10]. Each representation contains about a dozen of concepts, while the size of the reproduced scenarios is in the order of some hundreds of schemas.

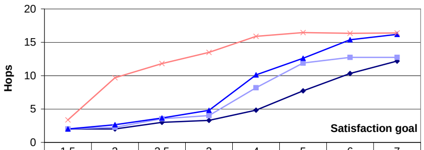
In order to perform a preliminary theoretical effectiveness evaluation of our proposal, we used a specifically devised index we called NetIndex [7] which quantifies the goodness of the networks resulting from our mechanism measuring their similarity w.r.t. manually designed ideal situations with a scores between 0 (bad network) and 1 (semantically ideal configuration). Fig. 3 shows the obtained results for both Range (Fig. 3-a) and k-NN (Fig. 3-b) selections and for different values of the threshold T which is used in the SON selection phase. As we expected, the NetIndex trends are growing for increasing values of radius (range) and k (k-NN), since the semantically ideal networks are very complex and thus the high number of connections is better approximated by larger radiuses and ks. However, as we can see, a radius of 0.2 or a k of 6, while avoiding to produce over-connected and, thus, possibly inefficient networks, already achieve very good semantic optimality grades. Finally, different thresholds such as $T = 0.3$ and $T = 0.7$ produce equally good results, while too high ($T = 0.9$) or too low ones ($T = 0.1$) produce semantically inferior network configurations.

Then, we deepen our effectiveness analysis in order to evaluate the impact of the network organization we propose on query answering by simulating a querying process on the networks produced by our algorithms. The results we present are collected for different values of radius and k. As to the querying process, we simulated it by instantiating different queries on randomly selected



	2	3	4	5	6	7	8	9	10
◆ Cls/SRI rout.	2.0000	2.8878	3.6927	4.2997	4.8267	5.4670	5.9989	6.2454	6.4273
■ Rnd/SRI rout.	1.9429	2.7100	3.1637	3.4787	3.7008	3.8711	4.0556	4.1001	4.1291
▲ Cls/rnd rout.	1.7889	2.3731	2.9431	3.5099	4.1256	4.4443	4.6244	4.7914	5.0540
× Rnd/rnd rout.	1.5365	1.7714	1.9673	2.1023	2.2063	2.4370	2.5011	2.6861	2.8036

(a) Satisfaction/Hops



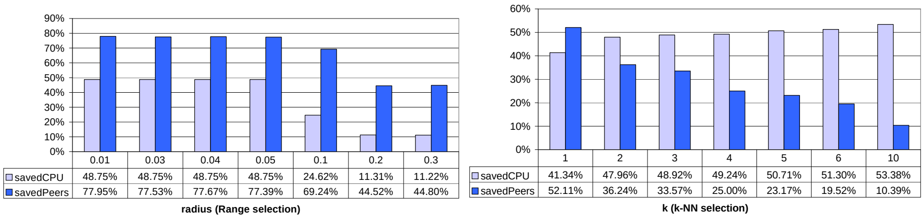
	1.5	2	2.5	3	4	5	6	7
◆ Cls/SRI rout.	2.0000	2.0000	3.0000	3.2929	4.82828	7.7253	10.3299	12.1818
■ Rnd/SRI rout.	2.0202	2.2525	3.5455	4.0202	8.18182	11.8889	12.7374	12.7374
▲ Cls/rnd rout.	2.0000	2.6465	3.6566	4.8085	10.1111	12.5979	15.3737	16.1919
× Rnd/rnd rout.	3.3721	9.6837	11.8081	13.4848	15.8889	16.4545	16.3434	16.4040

(b) Hops/Satisfaction

Fig. 4. Effectiveness tests: Network quality for query processing

peers where each query is a combination, through logical connectives, of a small number of predicates specifying conditions on concepts. We propagate queries until a stopping condition is reached [8]: we measure the quality of the results (*satisfaction*) when a given number of hops (*hop*) has been performed or, in a dual way, we measure the number of hops required to reach a given satisfaction goal. Satisfaction is a specifically introduced quantity that grows proportionally to the goodness of the results returned by each queried peer. Each contribution is computed by composing the semantic mappings scores of the traversed peers. The search strategy employed is the depth-first search (DFS). For the selection of the most promising neighbor, besides the simplest random strategy (*Rnd*), a Semantic Routing Indices (*SRI*) based strategy is experienced too. SRIs [8] implement a fully distributed indexing mechanism which summarizes the semantics underlying whole subnetworks. They are exploited to select the best direction to forward a query to, by relying on the estimated semantic degradation of information they maintain locally for each subnetwork. For both these routing strategies, we compared the results measured in our clustered networks (*Cls*) with random ones (*Rnd*). Notice that all the results we present are computed as a mean on several hundreds query executions. Fig. 4-a shows the trend of the obtained satisfaction when we gradually vary the stopping condition on hops, while Fig. 4-b represents the dual situation. As we expected, both for the random and the SRI routing strategies, the curves associated to the clustered networks outperform the corresponding ones for the un-clustered situations.

Finally, in order to evaluate the efficiency of our approach we firstly considered the CPU load generated on each peer by the execution of our algorithms of neighbors selection. In particular, we wanted to quantify the percentage (*savedCPU*) of useless distance computations that we are able to avoid thanks to the exploitation of the triangle inequality (see Section 3). Then, we focused on the number of peers that is necessary to contact in order to execute our neighbors selection processes. In particular, we were interested in quantifying the percentage (*savedPeers*) of useless peers that we can prune out. Fig. 5-a and 5-b show the savedCPU and savedPeers values obtained executing range and k-NN selection



	0.01	0.03	0.04	0.05	0.1	0.2	0.3
□ savedCPU	48.75%	48.75%	48.75%	48.75%	24.62%	11.31%	11.22%
■ savedPeers	77.95%	77.53%	77.67%	77.39%	69.24%	44.52%	44.80%

	1	2	3	4	5	6	10
□ savedCPU	41.34%	47.96%	48.92%	49.24%	50.71%	51.30%	53.38%
■ savedPeers	52.11%	36.24%	33.57%	25.00%	23.17%	19.52%	10.39%

(a) Range selection

(b) k-NN selection

Fig. 5. Efficiency tests: % of saved computations and contacted peers

processes, respectively. The results are collected for different values of radius and k. As we can see, for both graphs the obtained results are high, signifying a great saving for the CPU load and the number of contacted peers.

5 Conclusions

In this paper we presented a strategy for the incremental maintenance of a flexible semantic network organization for PDMSs. In the future, we plan to extend our work by investigating strategies for query processing in this context.

References

1. The SUNRISE Project. `http://www.isgroup.unimo.it/sunriseProject`.
2. K. Aberer, P. Cudré-Mauroux, M. Hauswirth, and T. V. Pelt. GridVine: Building Internet-Scale Semantic Overlay Networks. In *Proc. of ISWC*, pages 107–121, 2004.
3. A. Crespo and H. Garcia-Molina. Semantic Overlay Networks for P2P Systems. In *Proc. of the 3rd AP2PC Workshop*, pages 1–13, 2004.
4. C. Doulkeridis, K. Nørvåg, and M. Vazirgiannis. DESENT: Decentralized and Distributed Semantic Overlay Generation in P2P Networks. *IEEE J. on Selected Areas in Comm.*, 25(1):25–34, 2007.
5. V. Ganti, R. Ramakrishnan, J. Gehrke, A. Powell, and J. French. Clustering Large Datasets in Arbitrary Metric Spaces. In *Proc. of the 15th ICDE Conf.*, pages 502–511, 1999.
6. A. Halevy, Z. Ives, J. Madhavan, P. Mork, D. Suciu, and I. Tatarinov. The Piazza Peer Data Management System. *IEEE TKDE*, 16(7):787–798, 2004.
7. S. Lodi, F. Mandreoli, R. Martoglia, W. Penzo, and S. Sassatelli. Semantic Peer, Here are the Neighbors You Want! In *Proc. of the 11th EDBT Conf.*, 2008.
8. F. Mandreoli, R. Martoglia, W. Penzo, and S. Sassatelli. SRI: Exploiting Semantic Information for Effective Query Routing in a PDMS. In *Proc. of the WIDM (in conj. with CIKM)*, pages 19–26, 2006.
9. J. Parreira, S. Michel, and G. Weikum. P2PDating: Real Life Inspired Semantic Overlay Networks for Web Search. *Inf. Proc. & Manag.*, 43(3):643–664, 2007.
10. C. Yu and H. Jagadish. Schema Summarization. In *Proc. of the 32nd VLDB Conf.*, pages 319–330, 2006.