Dynamic Semantic Data Replication for K-Random Search in Peer-to-Peer Networks

Xiaoqi Cao
Computer Science Department
Saarland University
Campus E1 3, 66123 Saarbrücken, Germany
Email: Xiaoqi.Cao@dfki.de

Matthias Klusch
German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, Campus D3 2
66123 Saarbrücken, Germany
Email:Klusch@dfki.de

***Abstract*—We present a dynamic semantic data replication scheme called DSDR for classic k-random search in unstructured peer-to-peer (P2P) networks. During its k-random search each peer periodically updates its local view on the semantic overlay of the network based on observed queries (demand) and received information about provided items (supply), in particular their semantics. Peers dynamically form potentially overlapping groups for semantically equivalent or similar items they are actually demanding. Besides, each peer predicts the number of needed item replicas in the future based on its local observations in the past. The decision of which item to best replicate to which member is made within each demander group based on the maximal expected utility, traffic costs, and plausibility of such replication. Our experimental evaluation evidences that k-random search with DSDR-based replication can significantly outperform its combination with a near-optimal but non-semantic replication strategy, as well as a peer expertise-based semantic P2P search without replication.**

***Keywords*-semantic overlay, data replication, unstructured P2P networks.**

I. Introduction

Unstructured P2P networks like Gnutella, eMule, Morpheus and FreeNet are widely used for sharing user-generated multimedia content in a decentralized way. A classical way of peers to search for relevant data items in such networks is to perform k-random search [9] in which case the relevance is determined by the exact matching of query and item topics. It is well known that the performance of k-random search in unstructured P2P networks can be significantly improved by additional use of data replication strategies with reasonable traffic overhead [18], [8], [2], [15].

On the other hand, semantic search in unstructured P2P systems like in Bibster [6] and RS2D [1] determines the relevance of items based on the result of semantic reasoning on formal ontology-based annotations of items. This enables peers to make more informed decisions for query routing and item selection, which may result in higher precision and recall with reasonable traffic and computational overhead [16]. Semantic replication schemes utilize semantic relevance computation for replication decisions, which is to decide how many copies of which semantically relevant items to best replicate to which peers.

However, it is unknown under which conditions k-random search when combined with what kind of semantic replication can perform better than (a) its combination with near-optimal but non-semantic data replication, and (b) semantic search without any replication in such networks. Besides, to the best of our knowledge, there does not exist a semantic replication scheme for unstructured P2P networks yet. In this paper, we provide answers to these major research questions.

For this purpose, we present the first scheme for dynamic semantic data replication, called DSDR, in combination with k-random search in unstructured P2P networks and conduct a comparative evaluation with existing but fundamentally different alternatives. In particular, we show that k-random search when combined with DSDR can outperform its combination with the recently proposed near-optimal, non-semantic replication strategy P2R2 [15].

In addition, we show that depending on the item popularity distribution, k-random search with DSDR can also outperform an approach for semantic peer expertise based search [6] in unstructured P2P networks without replication. Besides, we provide experimental evidence for a significant outperformance of k-random search with P2R2 by its alternative minimal coupling with DSDR in terms of local data lookup tables with semantic synonyms. Moreover, our evaluation revealed that both combinations of DSDR with k-random search are at least as robust against changes of the network than k-random search with P2R2-based replication.

Finally, the DSDR scheme is agnostic to the kind of semantic description of data items and the selected method for semantic data relevance computation to be used by each peer. Though we conducted our experimental evaluation over a RDF test data collection from DBpedia, DSDR can easily be adjusted for replication of semantic services, or data items with other forms of ontology-based semantic descriptions.

In the following Sect.II we provide the underlying assumptions and definitions of our approach which is then detailed in Sect.III. Main results of its comparative experimental evaluation are presented in Sect.IV. We discuss the related works in Sect.V and conclude the paper in Sect.VI.

II. Preliminaries

In this section, we briefly provide the basic terms and assumptions which are required to understand our approach.

We assume that (a) each peer can provide, replicate, and request any data item replica from known peers under the copyright restriction, which in our context limits the number of replicas of an item an individual peer can propagate; (b) all peers share a minimal vocabulary of primitive concepts and roles out of which their users can canonically build local ontologies O_p in the same knowledge formalism like OWL2.

Definition 1: *Item, item concept*.
An *item* i provided and maintained by peer p consists of both data and metadata as defined by the item tuple $i = \langle l,\ \tau(C, O_p),\ URI,\ pid,\ sz,\ n_s,\ da\rangle$ where l is the label (topic term) of i; C the name of the semantic annotation concept and $\tau(C, O_p)$ its self-contained logical definition which describes the semantics of i in the local ontology O_p of peer p; URI the item identifier; pid the id of the peer p providing i; da the item data (e.g. movie file); sz the size of $i.da$; and n_s the number of available copies of i at p.

$i.C$ is called *item concept* of i. The item tuple without the item data $i.da$ is called metadata or *item description* ($i.desc$) of i. $i.l$ may correspond with the item concept $i.C$. ■

Definition 2: *Query, query satisfaction, query concept*.
A *query* q of a peer req is defined by the query tuple $q = \langle l,\ C,\ \tau(C,\ O_{req}),\ req,\ \{(res, its)\},\ \{Pa\},\ t,\ st,\ pbd,\ TTL,\ n_d\rangle$ where l denotes the query keyword (or topic of the query item); C the name of the *query concept* used to describe the semantics of the requested item and $\tau(C, O_p)$ its self-contained logical definition in O_{req}; req the identifier of the requesting peer; $\{(res, its)\}$ the actual answer set for the query which consists of pairs of identifiers res of peers who respond to the query with an array its of item descriptions; $\{Pa\}$ the set of query q paths; t the query issuing time; $st \in$ {*Issued, Success, Fail*} the query status where *Success* (*Fail*) means that the query q is satisfied (unsatisfied) and *Issued* indicates that the satisfaction of q has not been determined by the original requestor peer req yet (or else that q is not issued by the current peer); pbd the piggybacked data of a query containing information on provided items of peers along the query path; TTL the query time-to-live value; n_d the requested number of copies of the query item. ■

The semantic relevance of an item i to a query item i' is determined by means of semantic matching $dc(C, C')$ of their semantic annotation concepts C, C'. The semantic relevance computation means $dc : O_p \times O_p \to [0, 1]$ can vary for different kinds of semantic annotations such as the example of the dc-function given in Sect.IV for logic subsumption-based comparison of item concepts in OWL2.

The set of (relevant) items in the answer set $\{(res, its)\}$ is denoted as (I_{sq}) I_q; The topic terms of items in I_{sq} syntactically match the topic term of the query. A query q is *satisfied* or successful if $I_{sq} \neq \emptyset$ and $q.n_d \leq \sum_{i \in I_{sq}} i.n_s$.

III. Semantic Replication with DSDR

In this section, we first provide a brief overview of DSDR for k-random search in unstructured P2P networks. This is followed by the detailed presentation and examples.

A. Overview

During random search, in regular observation intervals, each peer p observes all its received queries along with their piggybacked item descriptions. While the queries indicate the actual demand of items by requesting peers including p itself, the piggybacked data on items including their semantic description provide p with knowledge about the actual supply in and the semantic overlay of the network from its local perspective. At the end of each observation period, p predicts for each semantic concept C the number $n_r(C, p, ot_{next})$ of item copies p will probably be requested for in its next observation interval ot_{next}. It computes the plausibility $pl(C, p)$ of its predicted demand of items based on all demands it has observed in the past. It then forms for each concept C a demander group $dg_p(C)$ with other known peers that have requested items which topics are semantically similar with the ones requested by p itself.

Once a group $dg_p(C)$ has formed by p, it will make a joint replication decision to determine how many copies of semantically relevant items i supplied by known peers outside the group should be replicated to which peer in $dg_p(C)$. For this purpose, p requests any other member peer p' in $dg_p(C)$ to provide its predicted demand $n_r(C, p', ot_{next})$ of items on C, the expected utility $EU(i, p')$ of replicating these item i to p', and its plausibility $pl(C, p')$. The replication decision to which member peer p^* how many item replicas shall best be replicated is then made w.r.t. the maximum expected utility and plausibility, while the number of replicas is the minimum of $i.n_s$ and $n_r(C, p^*, ot_{next})$.

B. Dynamic local observation, prediction and plausibility

Dynamic local observation. The dynamic local observation is the up-to-date local view of each peer p on the semantic overlay. Based on this, the demander group formation and semantic replication decision can be performed. Assume that the current interval ot_m is the m-th ($m \geq 0, m \in \mathbb{N}$) interval of p since it joins into the network. When ot_m ends, p computes (updates) its dynamic local observation.

Definition 3: *Local observation of* p *over all past intervals*.
The observation record of p is a series of values:

- $Q(p)$ ($Q(p, ot_j)$): the set of all queries observed by p (in the j-th ($0 \leq j \leq m,\ j \in \mathbb{N}$) interval);
- $Q_C(p)$ ($Q_C(p, ot_j)$): the set of queries observed in all past intervals (ot_j) about concept C' sufficiently semantically similar to concept C;
- $UQ(p)$ ($UQ(p, ot_j)$): the set of all non-successful queries q ($q.st \in \{Fail,\ Issued\}$) (observed in ot_j);
- $UQ_C(p)$ ($UQ_C(p, ot_j)$): the set of all non-successful queries (observed in ot_j) about concept C' that are sufficiently semantically similar to concept C;
- $\mathbf{C}_{UQ}(p)$ ($\mathbf{C}_{UQ}(p, ot_j)$): the set of concepts of non-successful queries in $UQ(p)$ ($UQ(p, ot_j)$);

• M: the set of pairs each associating the query concept C of a failed query issued in ot_m by p with a set $Q_{dist,C}(p, ot_m)$ of queries observed by p in ot_m but issued by other peers $p' \neq p$. Any $q \in Q_{dist,C}(p, ot_m)$ is requesting for items on concepts C' that are sufficiently semantically similar with C ($dc(C, C') > \theta, \theta \in [0, 1]$), where θ is set individually at each peer. Please note that $Q_{dist,C}(p, ot_m)$ contains exactly one query q for each originator $p' \neq p$ in the interval ot_m;
• $pop(C, p)$: the popularity of query concept $C \in \mathbf{C}_{UQ}(p)$: $pop(C, p) = \sum_{j=0}^{m} \frac{pop(C,p,ot_j)}{e\cdot(m-j)!} \in [0, 1], where$
$pop(C, p, ot_j) = \frac{n_C(p,ot_j)}{n_{total}(p,ot_j)} \cdot \frac{|UQ_C(p,ot_j)|}{|Q(p,ot_j)|}$;
$n_C(p, ot_j) = \sum_{q\in UQ_C(p,ot_j)} q.n_d$;
$n_{total}(p, ot_j) = \sum_{q\in Q(p,ot_j)} q.n_d$.
• $rc(C, p)$: the recentness of a query concept $C \in \mathbf{C}_{UQ}(p)$ is based on the most recent issuing time of a query for items about C over all periods. $rc(C, p) = \frac{q_{mr}.t - ot_{0,start}}{T\cdot m} \in [0, 1]$ where $ot_{0,start}$ is the starting time of the 0-th (first) interval, $q_{mr}.t$ is the issuing time of the most recent query q_{mr} for items about C, T is the length of each interval.■

Prediction. Based on the local observation, peer p predicts the number $n_r(C, p, ot_{m+1})$ of desired replicas of item about each demand concept $C \in \mathbf{C}_{UQ}(p)$, which is prone to be queried in the next ($(m + 1)$-th) observation interval of p.

Definition 4: *Predicted number of desired replicas on demand concept C of peer p in the ($m + 1$-th) observation interval.*
$N = \{\langle C, n_r(C, p, ot_{m+1})\rangle\}$. Each element pair associates a demand concept $C \in \mathbf{C}_{UQ}(p)$ with a predicted number $n_{r,C}(p, ot_{m+1})$, which can be computed via time series analysis [7] with double exponential smoothing:
$n_r(C, p, ot_{m+1}) = s_m + b_m, where$
$s_j = \phi n_C(p, ot_j) + (1 - \phi)(s_{j-1} + b_{j-1})$;
$b_j = \psi(s_j - s_{j-1}) + (1 - \psi)b_{j-1}$;
$s_0 = n_C(p, ot_0);\ b_0 = \frac{1}{2}(n_C(p, ot_1) - n_C(p, ot_0))$.■
The best values of weights ϕ ($\phi \in [0, 1]$) and ψ ($\psi \in [0, 1]$) are computed via the Levenberg-Marquardt algorithm (LM) [10] that efficiently resolves the following least squares problem: $MSE_m(\phi, \psi) = \frac{1}{m+1} \sum_{j=0}^{m} (s_j - n_C(p, ot_j))^2$;
minimize : $MSE_m(\phi, \psi)$; subject to : $\phi \in [0, 1], \psi \in [0, 1]$.
where MSE refers to the mean squared error between the smoothed and observed values.

Plausibility. Plausibility $pl(C, p)$ refers to the overall strength of the demand on concept $C \in \mathbf{C}_{UQ}(p)$ observed by p in the past intervals. Based on the evidence theory in [11], this value will be used as support for the expected utility computed in the replication decision process.

Definition 5: *Plausibility of the demand on concept C observed by peer p.*
Let $\mathcal{H} = 2^{\mathbf{C}_{UQ}(p)}$ the power set of $\mathbf{C}_{UQ}(p)$; $v : \mathcal{H} \to [0, 1]$ the mass function subject to the properties: $v(\emptyset) = 0; \sum_{H\subseteq\mathcal{H}} v(H) = 1$. Plausibility $pl(C, p)$ is computed by:
$v(H) = \frac{n_H}{n_{\mathcal{H}}};\ n_H = \sum_{C\in H} n_C(p);\ n_{\mathcal{H}} = \sum_{H\subseteq\mathcal{H}} n_H$;
$n_C(p) = \sum_{j=\{1,...,m\}} n_C(p, ot_j);\ Bel(H) = \sum_{h\subseteq H} v(h)$;
$pl(C, p) = 1 - Bel(\mathbf{C}_{UQ}(p)\backslash C))$.■

C. Demander group formation

One key idea of DSDR is that each peer forms a demander group (cf. Alg.1) for each of its own unsatisfied requests that were observed in the most recent observation period with peers that actually share semantically similar demands. Such demander groups are formed in a distributed fashion.

Definition 6: *Demander group.*
A demander group for query concept C at peer p is $dg_p(C) = \{p' : \exists q \in UQ(p) : q.req = p', dc(q.C', C) \geq \theta\}$. This group is represented at each other member peer p as $dg'_p(C')$ and commonly recorded at every member as a tuple $\langle dgid, P_{dg}, C_{dg}\rangle$ where $dgid$ denotes the UUID of the group, P_{dg} the set of member peers p knows, and $C_{dg} = \{C\}$. ■

Algorithm 1 GroupConstruction() of peer p.

1: Let ot_m the current observation interval of p.
2: **for** each pair $\langle C, Q_{dist,C}(p, ot_m)\rangle$ in M **do**
3: Let q the corresponding query of the query concept C;
4: p locally creates a group tuple $dg_p(C) = \langle dgid, \{p\}, C\rangle$;
5: **for** each query q' in $Q_{\sim C}(p)$ **do**
6: p sends a message ($p.id$, $q.\tau(C, O_p)$, $q'.\tau(C', O_p)$, *Dgc*, $dgid$) to $q'.req$ (denoted as p') and receives a *result*. C' is the query concept of q' under O_p;
7: p does $P_{dg} \leftarrow P_{dg} \cup p'.id$, **if** $result =$ *Ack*;
8: p does ReplicationDecision($i.desc$) for any gossiped item i, **if** $result =$ *Gsp*;
9: p does nothing, **if** $result =$ *Refuse*;
10: **end for**
11: **end for**

From the view of peer p' (cf. line 6) whose query $q' \in Q_{dist,C}(p, ot_m)$ with $q.C'$ is observed by p, depending on $q'.st$, p' replies to p with different messages as follows: If $q'.st = Fail$ then p' shares a semantically similar unsatisfied demand with p and therefore acknowledges the group formation invitation from p. In addition, p' represents the initial demander group of p internally as $\langle dgid, \{p', p\}, C'\rangle$. If $q'.st = Success$ or any item i which is relevant for q' has been observed by p' during its current observation interval (which is not necessarily synchronized with the observation interval of p) then p' replies p with a gossiping message (Gsp) containing the metadata of i. Hence p has a chance to decide on the replication of such item i within the group. If $q'.st = Issued$ then p' cannot yet determine the satisfaction of q' and replies with a message $Refuse$. The formation of a demander group is not synchronized, and demander groups can overlap. The introduced distributed process can result in multiple versions of representations of the same group at member peers. However, all these versions have the same group identifier. If multiple peers send invitations for group formation to each other at the same time then the invitation from the peer with maximal lexicographical UUID is valid.

D. Semantic replication decision

The completion of group construction on some concept C at peer p triggers the replication decision for each of its observed item i. Performed by member peers collaboratively, this process (cf. Alg.2) is to decide how many replicas of what item shall be best replicated to which member peer.

Algorithm 2 ReplicationDecision($i.desc$) of peer p

1: Let C' the item concept of data item i (maintained by peer pro) under O_p;
2: **for** any demander group $dg_p(C)$ p belongs to **do**
3: $\quad$ p locally computes the (i) semantic gains $g_r(C, C', p, pro)$, $g_{nr}(C, C', p, pro)$, (ii) expected utility $EU(i, p)$ of replicating i to p, (iii) predicted number $n_r(C, p, ot_{m+1})$ of copies of items that would be requested in the next interval (the $(m+1)$-th interval) and (iv) the plausibility $pl(C, p)$ of the demand on concept C;
4: $\quad$ p sends $i.desc$ to any other peer p' in $dg_p(C)$, requests p' to compute and return the same series of values: $[g_r(C, C', p', pro)$, $g_{nr}(C, C', p', pro)$, $EU(i, p')$, $n_{r,C}(p', ot_{next})$, $pl(C, p')]$;
5: $\quad$ p computes a candidate set $P_T = \{p' | g_r(C, C, p', pro)$ $> g_{nr}(C, C', p', pro)$, p' in $dg_p(C)\}$;
6: $\quad$ p selects the recipient peer p^*=$maxarg_{p' \in P_T}$ $(EU(i, p') \cdot pl(C, p'))$;
7: $\quad$ p sends message to item i's providing peer pro for replicating item i to p^*, if p^* is p; p sends message to p^* telling it to replicate item i otherwise.
8: **end for**

The semantic gains (cf.line 3) $g_r \in [0, 1]$ $(g_{nr} \in [0, 1])$ indicate the estimated benefit of p for (not) obtaining replicas of items on concept C' from the known item provider pro such that its demand for items on C' or semantically similar concepts C could be satisfied. Further, these semantic gains for p are then traded off with their estimated traffic costs, yielding the individual expected utility $EU(X)$ of such replication for p (cf. Def.7). The predicted number of replicas $n_r(C, p, ot_{m+1})$ is used to compute the network traffic penalty depending on the communication overhead cm_{in} of the messages exchanged between peers during the replication decision process, and the amount of traffic produced by the replication. That is the number $min(n_r(C, p, ot_{m+1}), n_{C'}(pro))$ of replicated items of size $i.sz$ with traffic costs κ per unit of the size. The utility $u(\bar{X})$ of not replicating i bases on the inverse costs cm_{in}.

Definition 7: *Semantic gain, expected utility*
Consider dc, pop, rc as defined in Def.3. Let i an item provided by peer pro on a topic which semantics is defined with concept $C' \in O_{pro}$ and C is a query concept of an unsatisfied query of p. The *semantic gains* g_r, g_{nr} and the *expected utility* EU of replication and non-replication of item i from pro to p are defined as follows:
$g_r(C, C', p, pro) = dc(C, C') \cdot rc(C, p) \cdot pop(C, p)$;
$g_{nr}(C, C', p, pro) = (1 - dc(C, C')) \cdot (1 - rc(C, p)) \cdot (1 - pop(C, p))$;
$EU(X) = P(X) * u(X) + P(\bar{X}) * u(\bar{X})$;
$P(X) = \frac{g_r(C, C', p, pro)}{g_r(C, C', p, pro) + g_{nr}(C, C', p, pro)}$;
$P(\bar{X}) = \frac{g_{nr}(C, C', p, pro)}{g_r(C, C', p, pro) + g_{nr}(C, C', p, pro)}$;
$u(X) = (cm_{in} + \kappa \cdot i.sz \cdot min(n_r(C, p, ot_{m+1}), n_{C'}(pro)))^{-1}$; $u(\bar{X}) = -cm_{in}^{-1}$.
where X $(\bar{X})$ denotes the event of (not) replicating item i from pro to p; $P(X)$ $(P(\bar{X}))$ the probability of (not) replicating based on the computed gains; $u(X)$ the utility of replication in terms of a traffic cost penalty.■

Peer p receives the necessary values (cf. lines 3–4) from all other group members p'. That enables p to identify those qualified candidates for replication (cf. P_T in line 5) and then to determine the most beneficial target peer p^* (cf. lines 6–7) within the group. This process is based on the product of the collected expected utility and supporting plausibility from each peer $p' \in P_T$: $p^* = maxarg_{p' \in P_T}(EU(i, p') \cdot pl(C, p'))$ where $EU(i, p')$ represents the expected utility $EU(X)$ of replicating item i to some peer p' of the group including p. Finally, only the target peer requests the provider peer pro for the item data on C. Providers may satisfy such requests for item data downloads to the target peers of demander groups on a first come first served basis. After replication, peer p^* (i) discards the local record of the group $dg_{p^*}(C)$ (ii) decreases $q.n_d$ of any $q \in Q(p)$ with demand concept C by the number of actually replicated copies and (iii) updates the local observation and prediction.
Complexity. The overall traffic $\mathcal{O}(|V|)$ and computation $\mathcal{O}(|V|\eta)$ complexity of both group construction and replication decision in the worst case are linear with the total number of peers $|V|$ in the network. The complexity $\mathcal{O}(\eta)$ of semantic relevance computations by the dc-function depends on the chosen ontology language for semantic descriptions (e.g. the OWL2 concept subsumption-based dc-function used for our experiments in Sect. IV is in NEXP). The overhead of updating local prediction is trivial by the iterative computation in Def.4[1]. For computing the plausibility $pl(C, p)$, peer p has to execute $\mathcal{O}(2^{|\mathbf{C}_{UQ}(p)|-1})$ extra operations for each observed query in the current interval. However, this can be performed off-line, which is totally in parallel with p's routing process.

[1]LM algorithm iteratively finds the proper weights ϕ, ψ near to the optimal. It terminates if the difference of target function values $(\nabla MSE_m(\phi, \psi))$ is less than a given threshold ϵ, which can be achieved in $\mathcal{O}(\epsilon^{-2})$ iterations [13]. For efficiency, it is proper to set $\epsilon = 1$ since its volume is at the level of $\mathcal{O}(mn^2) \gg \mathcal{O}(1^{-2})$ where n denotes the number of requested replicas. This guarantees that p can compute good values of weights in one iteration.

Example. We illustrate the principled working of DSDR-based replication combined with k-random search by a simple example of an unstructured P2P network $N = (V, E)$ which just consists of three sequentially connected peers with $V = \{p, p_1, p_2\}$, $E = \{(p, p_2), (p_2, p_1)\}$. Each peer has its local ontology defined in OWL2 and only one item is available which is provided by peer p_1. This item $i_1 = \langle taxi, Taxi, \tau(Taxi, O_{p_1}), uri(i_1), uuid(p_1), 1.5MB, 50,$ $yellowcabs.mpg\rangle$ is labeled with the topic term $i_1.l = cab$ which formal semantics is defined by the concept $i.C = Taxi$ in the local ontology O_{p_1} of p_1.
k-random search. Suppose that peer p is searching for one item i ($i.l = taxis$) which formal semantics is defined by the concept $i.C = CAB$ in O_p. Since p does not have any items at all, it randomly forwards the issued query $q = \langle cab, CAB, \tau(CAB, O_p), uuid(p, N), \emptyset, \emptyset,$ 070312:1:15pm, $Issued, [-], 2, 1\rangle$ to a number k of its neighbor peers, in this case only peer p_2 with $q.TTL = 2$. Since peer p_2 has no items neither it forwards the query to one of its neighbor peers, that is p_1 which determines that its only item i_1 is not relevant for q since the item and query topic terms do not syntactically match. Peer p_2 then triggers ($q.TTL = 0$) the backward propagation of the random walker q to its originator p along the same path in reverse direction who eventually determines that the query q is unsatisfied ($q.st$ = Fail). Meanwhile, peer p_2 issued a query q_1 on the topic "yellowcars" with semantic annotation concept $TaxiVehicles$ defined in the local ontology O_{p_2}.
Observed semantic overlay. Before returning the query q along its path, p_1 adds the semantic descriptions of all its items to the piggybacked dataset of q, in this case only the one for item i_1 ($(Taxi, \tau(Taxi, O_{p_1}))$). As a result, upon receipt of the returning random walker q its originator peer p knows the actual semantic domain of items provided by p_1 which becomes part of its observed semantic overlay of the network. Peer p_1 decides when to further exploit some random walker it receives from p to communicate an update of its semantic item domain to p. Besides, peer p observed that its neighbor p_2 is demanding items with a certain semantic description.
Prediction and plausibility. On p's observation of q, p employs an independent thread to update the local plausibility $pl(CAB, p)$ of demand concept CAB in parallel with its main thread for routing. This yields available result for the replication decision for items on CAB, which is possible to happen in the future. Suppose that the current (m-th) observation interval ot_m of p now ends. p update its predicted number $n_r(CAB, p, ot_{m+1})$ of the requested replicas about concept CAB for ot_{m+1}. That is to compute (cf.Def.4) s_m, b_m and $n_r(CAB, p, ot_{m+1})$ based on the volumes of s_{m-1} and b_{m-1} in ot_{m-1}).
Demander group formation. Since q was unsatisfied in the past period, peer p uses its actual local knowledge about the semantic overlay to form a demander group with those peers from which it received queries for items with semantically similar descriptions. In fact, p invites p_2 to form a demander group $dg_p(CAB) = \{p, p_2\}$ on CAB since both peers demanded semantically equivalent items on this subject in the past observation period: $dc(CAB, TaxiVehicles) = (\tau(TaxiVehicles, O_p2) \equiv \tau(CAB, O_p)) = 1$. Since p_2's query is unsatisfied as well the demander group on concepts $CAB, TaxiVehicles$ is formed and denoted as $dg_{p_2}(TaxiVehicles)$ at p_2 and as $dg_p(CAB)$ at p.
Replication decision within demander groups. In this example, p knows that p_1 supplies the item i on concept $Taxi$. For deciding the replication of i, both member peers then compute their own semantic gains, expected utility, predicted number of future requested replicas and plausibility of the demand on concept semantically similar to $Taxi$. These values are sent to p on p's request. Assume that peer p has the maximal product of expected utility and plausibility within the group, it requests the provider p_1 to download the desired number of replicas of the item i_1 data.

IV. Evaluation

We present and discuss the results of our comparative experimental evaluation of the performance of k-random search with DSDR for different configurations of P2P networks.
Setup. For our experiments, we created unstructured P2P networks with one million peers and topologies based on random graphs (RG) and random power law graphs (RLPG). The latter is known to be a realistic model in particular for social networks. Further, we employed two models of item popularity distribution in these networks which are used for many real-world item popularity rankings: Uniform at random (R) and Zipf's law (Z) based distribution. We restrict the k-random search by all peers to $k = 3$. The initial value of TTL (TTL_{init}) of each walker is 20.

As a test collection we use a random subset of 50k RDF linked data items (in files: instance_types_en.nt.bz2 and mappingbased_properties_en.nt.bz2) taken from DBpedia (http://downloads.dbpedia.org/3.7/en/) with its ontology (dbpedia_3.7.owl.bz2) O of 319 defined concepts and 1635 roles. We built peer ontologies through random sampling of 250 concepts and 1450 roles taken from O on average.

For non-semantic random search the relevance of items for queries is based on the Levenstein edit distance between their topic terms. The semantic relevance of an item with annotated concept C with a query on concept C' is computed by semantic similarity $dc(C, C')$: $dc(C, C') = [1.0 \quad \text{if } C \equiv C';\ 0.9 \quad \text{if } C \sqsubseteq_1 C'$ **or** $C \sqsupseteq_1 C';\ 0.1 \quad \text{if } C \sqsubseteq_k C'$ **or** $C \sqsupseteq_k C', \quad k > 1, k \in \mathbb{N};\ 0$ otherwise.] Since DBpedia does not provide the relevance sets for item queries, we use the following heuristics for relevance judgments: Item i about concept C is relevant (a true positive) for query item i' about concept C', if any of the logic-based concept relations in $\{C \equiv C', C \sqsubseteq_1 C', C \sqsupseteq_1 C'\}$ holds. The semantic relevance threshold θ

for the demander group construction is 0.5. All experiments are conducted via our semantic P2P simulation framework (http://sourceforge.net/projects/dsdr/).

Evaluation measures. Let Q the set of queries in the network; I_q ($I_{q,j}$) the set of items collected by k walkers of a query $q \in Q$ (at its j-th hop, $1 \leq j \leq TTL_{init}; j \in \mathbb{N}$); I^*_q ($I^*_{q,j}$) the set of relevant items in I_q ($I_{q,j}$); $I^*_{tq,j}$ (I^*_{tq}) the set of relevant items for q at the j-th peer (all peers) on the query path;

• Average cumulative recall (CRE_m) over all queries in Q: $CRE_m = \frac{1}{|Q|}\sum_{q\in Q}\frac{\sum_{j=1}^{m}|I^*_{q,j}|}{|I^*_{tq}|}$.

• Macro-averaged precision (MAP_λ) at 11 recall levels (RE_λ) with equidistant steps of 0.1: $MAP_\lambda = \frac{1}{|Q|}\sum_{q\in Q} max\{pre_{q,m}|re_{q,m} \geq RE_\lambda,$ for $\forall\langle pre_{q,m}, re_{q,m}\rangle \in PR_q\}$. A set PR_q of precision-recall $\langle pre_{q,m}, re_{q,m}\rangle$ pairs is computed for each query q at different number of hops m. Nearest-neighbor interpolation is used for estimation of missed precision values for some queries at some recall levels: $PR_q = \{\langle pre_{q,m}, re_{q,m}\rangle\} = \{\langle\frac{\sum_{j=1}^{m}|I^*_{q,j}|}{\sum_{j=1}^{m}|I_{q,j}|}, \frac{\sum_{j=1}^{m}|I^*_{q,j}|}{\sum_{j=1}^{m}|I^*_{tq,j}|}\rangle\}$.

• Averaged precision $ap = \frac{1}{|Q|}\sum_{q\in Q}\frac{|I^*_q|}{|I_q|}$.

• Traffic utility $tu = \frac{\text{total \# of successful queries}}{\text{total \# of hops}}$. Instead of the avg. messages per query, this metric exams the usage of overall traffic cost on satisfying user queries with the underlying data replication.

• Replication utility $ru = \frac{\text{total \# of successful queries}}{\text{total \# of replicas}}$.

Experiment 1: Semantic-based vs. non-semantic-based replication. We compare the retrieval performance of the same k-random search with different replication schemes: DSDR, DSDR plus lookup table (DSDR(L)), and non-semantic P2R2 in a RPLG-based network with one million peers and 50k initial items. The item popularity distributions considered are uniform at random (R) distribution over all items, and the Zipf (Z) distribution ($\beta = 1.05$) over pre-clustered 127 topics. The latter is a well known model for the common search behavior of human users. Our experiments revealed that the k-random search when combined with DSDR-based replication can significantly outperform its combination with the non-semantic replication P2R2 in terms of precision (Fig.1, top-left) and cumulative recall (Fig.1, top-mid). Particularly, it achieved 27.3% more precision with similar volumes of traffic utility (Fig.1, top-right).

Synonym lookup table L is an optional structure for each peer to store the learned term relevancy during semantic replication decision. Each record (e.g. ($taxis$, cab, $yellowcars$) in the example in Sect.III) in L contains the unsatisfied query terms (cab and $yellowcars$) and replica terms ($taxis$). Based on this, syntactic irrelevance on semantically relevant terms is less prone to happen in future queries. When DSDR working without lookup table, syntactic based local item selection at peer p to a query for item i could yield syntactic false negative given a true positive item i_1 at p. DSDR indirectly bridges this gap by enabling p to replicate another observed item i_2 semantically similar to i, on the observed demand of i. To some extent, the transitive syntactic relevancy between i and i_2 can exist. It follows that the replica i_2 can syntactically match the future queries for i. Thus, syntactic true positive can be achieved in future rounds. Our experiment result evidences this. Both precision and recall are indirectly improved over time as they increase monotonically w.r.t. the probability of p receiving a replica that is a true positive for future queries on the same topic.

DSDR needs extra network traffic for the group construction and replication decision. For uniform item popularity distribution, k-random search with DSDR yielded lower traffic utility (Fig.1, top-right) than the search with P2R2 while the precision of both are similar in this case. Besides, the replication utility of the former is slightly higher than the latter (Fig. 1, bottom-left). Finally, the k-random search with DSDR(L) performs even better than its combination with DSDR or P2R2 regardless of the item popularity distribution, as the relevancy computations of search and replication are minimally coupled with lookup tables. Overall, these results clearly evidence the benefit of employing the semantic-based replication scheme DSDR rather than a non-semantic replication scheme like P2R2 with k-random search.

Experiment 2: Semantic search without replication vs. Non-semantic search with semantic replication. We test in this experiment whether simple but fast random search with semantic replication can outperform semantic search without replication. We choose the representative semantic search Bibster [6] as the competitor of the k-random search with DSDR. Each peer in Bibster routes query to at most s ($s \geq 0, s \in \mathbb{N}$) peers whose peer expertise are semantically similar to query. The peer expertise knowledge is advertised by each peer via TTL bounded (denote b ($b > 0, b \in \mathbb{N}$) the bound) flooding after its joining into the network. We independently run these two competing systems in random graph networks with one million peers under different initial number of items and peer connectivity degrees. For Bibster, $b = 5$; $TTL_{init} = 5$; $s = 2$. For k-random search with DSDR, $TTL_{init} = 20$, $k = 3$. This setup ensures that each query can traverse at most similar number of peers, though systems run with different routing strategies.

Shown in Fig.1 (bottom-mid), Bibster achieves better ap k-random search with DSDR in case the number (50k) of initial items and peer connectivity degree (5.75) are relatively large. However, the latter combination performs better if the initial number of items (5k) or the peer connectivity degree (2.17) is smaller. The reason is that the effect of peer expertise advertising is sensitive to the decrease of both factors. When items are rare in the network, it is possible that a query can never be satisfied if there is no available knowledge about the desired items on remote peers that is reachable from the query originator within TTL number of hops. Data replication overcomes this issue by enabling

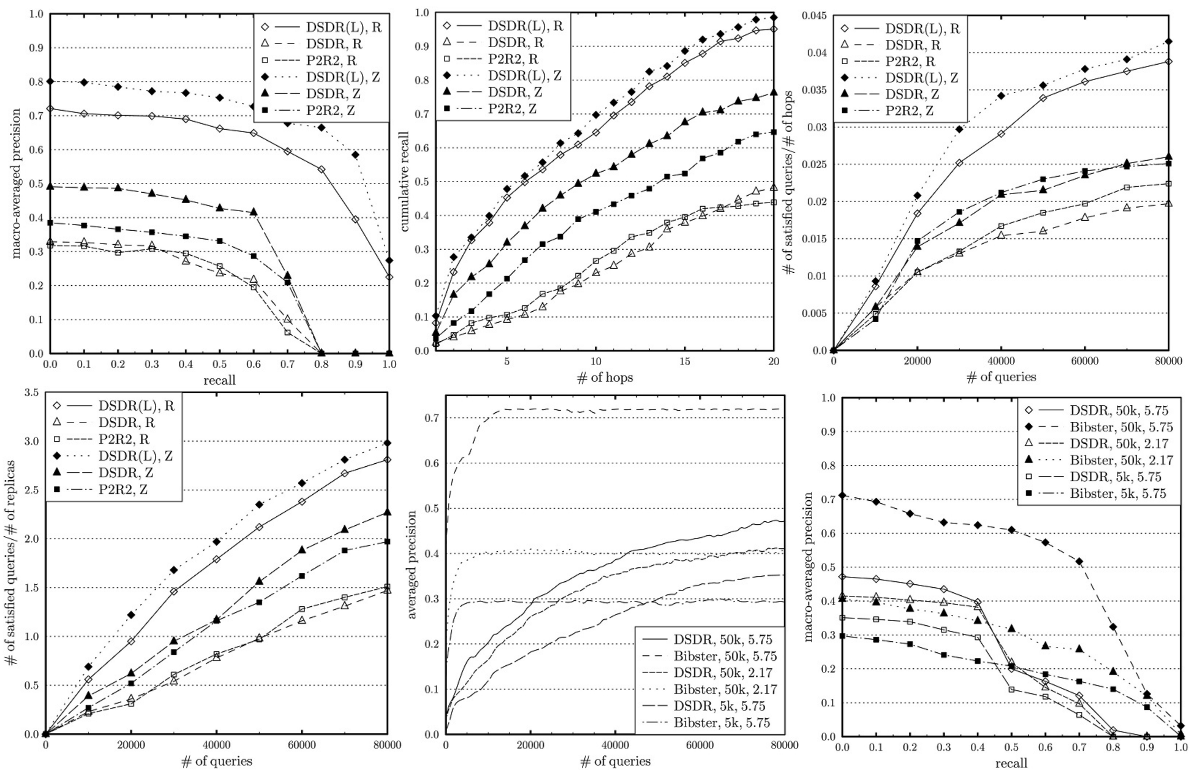

Figure 1: (top-left, top-mid, top-right, bottom-left) Search performance comparison between k-random search with DSDR and the same search with P2R2; (bottom-mid, bottom-right) Search performance comparison between k-random search with DSDR and Bibster without replication.

peer (transitively) to propagate (replicate) items to remote peers according to the observed demands. It indirectly breaks through the TTL limitation and therefore enlarges the chance of peer's finding the desired item. When the peer connectivity degree is smaller, the expertise information will be known by less peers, which is crucial for routing.

Bibster performs better than k-random search with DSDR in terms of MAP@recall (Fig.1, bottom-right) if the initial number of items and peer connectivity are large. K-random search with DSDR achieves better precision at small recall levels (0.0–0.5), while the latter results in better recall at large recall levels (0.6–1.0). The reason is that DSDR replicates items based on the semantic relevance of the demand and supply. These replicas then have risk to be ignored by the syntactic based item selection of k-random search. This situation is much less serious in the Bibster case, in which the semantic item selection is used.

Experiment 3: Robustness. Random search with non-semantic replication by P2R2 has been shown to be highly robust against dynamic changes of the network topology [15]. Our third experiment analyses the robustness of k-random search with DSDR for networks with RPLG-based topology containing one million peers which randomly issue 200k queries. After the processing of 80k and 160k queries, we randomly deleted 200k peers from the network while adding them randomly to the network after 120k queries were processed. As expected, both types of topology changes resulted in a decrease of precision for some time, since either the replicas were removed or the semantic overlay structure was partially destroyed through peers leaving the network, or the replicas were diluted in case of peers entering the network. The averaged precision of k-random search with both replication methods dropped at each change event (Fig.2) but both systems were able to recover within almost the same time period. In this context, not surprisingly, the leaving of peers had a greater negative impact than the

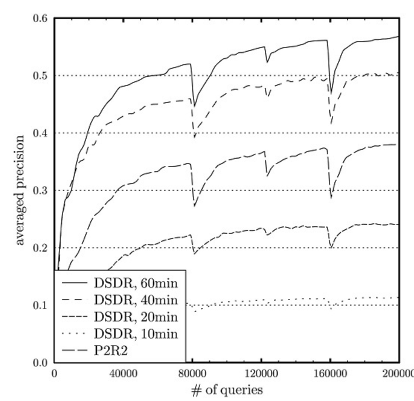

Figure 2: Robustness under churn: averaged precision.

arrival of new ones. For DSDR, the individual time interval T of each peer is crucial. In this experiment, for each run of DSDR, we manually control the expected value $E(T)$ of interval lengths, like 40min, by configuring the interval lengths of peers to uniformly distribute between $0.5E(T)$ and $1.5E(T)$, like 20min and 60min. The evaluation revealed that shorter intervals are less prone to such topology changes than larger intervals while the recovery behavior is comparable to P2R2 but with significantly higher precision.

V. Related works

To the best of our knowledge, DSDR is the first dynamic semantic replication scheme for k-random search in unstructured P2P networks. It differs from previous efforts, like [2], [15], [5], [8], [9], [12], [17] and the ones in the survey [18], in that it incorporates the statistics of observed demand or supply and the semantic relevance between them as well.

In [2], the square-root rule is derived, which identifies the optimal number of replicas to minimize the expected random search size rather than to improve the search performance in terms of cumulative recall and precision@recall like DSDR. The works [5], [12], [8] and [17] propose the proactive item replication strategies by which each item providing peer issues probing or limited flooding messages that either detect the item rareness or advertises its items. In addition to the risk of large network traffic, the actual demands of users in the network are not considered. In comparison with such proactive replication strategies, the DSDR scheme appears to be more demand-oriented based on actual local observations. In P2R2 [15], the replica distribution problem is reduced to the known multi-knapsack problem by regarding replicas as elements that are supposed to be put into bins which are representing target peers. Its replication scheme has been proven to converge to a 2-approximation solution of the problem under the following assumptions: the scheme is executed only in small networks in a sufficiently long steady state since the knapsack algorithm needs to know all the bins in advance. Unlike DSDR, this assumption hinders the applicability of P2R2 to large-scale scenarios. In another work[14], each peer is prone to replicate its provided items to the remote peers that have larger query routing traffic, in order to guarantee higher replication utility. In contrast to DSDR, this replication strategy is less robust, highly sensitive to network dynamics. Other approaches, such as [4], [3], [19], are based on peer grouping or partitioning widely used in P2P systems so far. While the first two systems make the assumption of global knowledge of the network which renders them unsuitable for unstructured P2P scenarios, the latter is restricted to the flooding radius thus, unlike DSDR, being very sensitive to the network topology.

VI. Conclusion

We presented and comparatively evaluated the first dynamic semantic data replication scheme DSDR for k-random search in unstructured P2P networks. Main contribution is that the search performance of k-random search with DSDR can outperform the same search combined with a near-optimal data replication and a semantic P2P search with elaborated routing but without replication as well.

References

[1] Basters, U.; Klusch, M. (2006): Risk-Driven Semantic P2P Service Retrieval. Proc. 6th IEEE Intl. Conference on P2P Computing.

[2] Cohen, E.; Shenker, S. (2002): Replication strategies in unstructured peer-to-peer networks. ACM SIGCOMM Comp. Comm. Review, 32(4).

[3] Doulamis, N.D.; Karamolegkos, P.N.; Doulamis, A.D.; Nikolakopoulos, I.G. (2007): Optimal decomposition of P2P networks based on file exchange patterns for multimedia content search and replication. Proc. ACM Intl. Workshop on multimedia information retrieval.

[4] Doulkeridis, C.; Norvag, K.; Vazirgiannis, M. (2007) DESENT: Decentralized and distributed semantic overlay generation in P2P networks. *Selected Areas in Communications*, 25(1).

[5] Gao, G.; Li, R.; Wen, K.; Gu, X. (2012): Proactive replication for rare objects in unstructured peer-to-peer networks. *Network and Computer Applications*, 35.

[6] Haase, P., et al. (2006): Bibster - A Semantics-Based Bibliographic Peer-to-Peer System. Chapter 18 in [16].

[7] Hamilton, J.D. (1994): Time series analysis, Cambridge Univ Press.

[8] Leng, C.; Terpstra, W.W.; Kemme, B.; Stannat, W.; Buchmann, A.P. (2008): Maintaining replicas in unstructured P2P systems. In: A. Azcorra, G. de Veciana, K. W. Ross, and L. Tassiulas, editors, CoNEXT.

[9] Lv, Q.; Cao, P.; Cohen, E.; Li, K.; Shenker, S.: Search and Replication in Unstructured Peer-to-Peer Networks. Proc. 16th Intl. Conf. on Supercomputing.

[10] Moré, J. J. (1977): The levenberg-marquardt algorithm: Implementation and theory. Lecture Notes in Mathematics, vol 630. Springer.

[11] Parsons, S. (2001): Qualitative methods for reasoning under uncertainty, MIT Press.

[12] Puttaswamy, K.; Sala, A.; Zhao, B. (2008): Searching for rare objects using index replication. Proc. 27th IEEE Intl. Conf. on Comp. Communications.

[13] Ueda, K. and Yamashita, N. (2012): Global complexity bound analysis of the levenberg-marquardt method for nonsmooth equations and its application to the nonlinear complementarity problem. In Journal of Optimization Theory and Applications, 152(1).

[14] Shen, H. (2008): An Efficient and Adaptive Decentralized File Replication Algorithm in P2P File Sharing Systems. *Parallel and Distributed Systems*, 21(6).

[15] Sozio, M.; Neumann, T.; Weikum, G. (2008): Near-Optimal Dynamic Replication in Unstructured Peer-to-Peer Networks. Proc. 27th ACM SIGMOD Symposium on Principles of Database Systems.

[16] Staab, S.; Stuckenschmidt, H. eds. (2006): Semantic Web and Peer-to-Peer, Springer.

[17] Terpstra, W.W.; Kangasharju, J.; Leng, C.; Buchmann, A.P. (2007): Bubblestorm: Resilient, probabilistic, and exhaustive peer-to-peer search. *SIGCOMM Computer Communication Review*, 37(4).

[18] Thampi, S.M. (2010): Survey of search and replication schemes in unstructured P2P networks. *Network Protocols and Algorithms*, 2(1).

[19] Zhou, J.; Zhang, X.; Bhuyan, L.; Liu, B. (2007): Clustered K-Center: Effective Replica Placement in Peer-to-Peer Systems. Proc. Global Telecommunications Conference, IEEE.