Semantic-Driven K-Walkers-Based Search in Unstructured Peer-to-Peer Networks

Xiaoqi Cao
German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, Campus D3 2
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*—In this paper, we present a semantic-driven k-walkers-based search scheme, called S2P2P, for data information dissemination and query routing in unstructured peer-to-peer (P2P) networks. In S2P2P, each peer maintains its observation on the semantics of received queries (demands) and data information (supplies), as well as a local view on network topology. On top of this, each peer, in line with forwarding a query, disseminates its known data information to a selected set of remote peers by taking advantage of query piggybacked data. For routing a query, each peer, instead of merely introducing an immediate neighbor or remote peer, suggests a query routing path containing a sequence of peers with expertise on the similar topic of query. This is achieved by a path suggestion heuristics that iteratively applies Dijkstra's algorithm in a greedy manner. Each iteration manages to detect one more expert peer and augments the current path suggestion with the shortest path from its tail to the detected expert peer. The comparative experimental evaluation shows that S2P2P outperforms a semantic flooding based search strategy in terms of search precision and recall. In addition, our evaluation reveals that S2P2P is as least as robust against the network dynamics than the semantic flooding approach.**

I. Introduction

Unstructured P2P networks like Gnutalla, eMule, FreeNet and Morpheus are widely used for decentralized file sharing. Flooding and k-walkers [17] are two of the classic ways for searching data items. The performance of their variants in terms of search precision/recall and network traffic highly depends on item information dissemination and query routing strategy. Many contributions take advantage of (restricted) flooding [11], [16], [18] or selectivity enforced gossiping mechanism [5], [12], [22], [26] to propagate query or item information to remote peers. These approaches commonly offer fairly high recall but almost suffer from a relatively large network traffic cost. Besides, periodically fashioned gossiping of item information dissemination has the risk of propagating duplicated messages for identical item. Those messages does not pay off for the unpopular data that exists very commonly in social networks. K-walkers based approaches, such as [2], [25], [27], [9], [15], provide intelligent query routing strategies by means of machine learning techniques, network topology and query analysis. Although these variants are capable of alleviating the network traffic problem above, unfortunately, their routing suggestion almost contains mere the next peer (commonly one of the immediate neighbors or a remote peer) that maintains the information helpful to answering the current query. This could yield the insufficient use of item expertise information and further the loss of relevant results. How to efficiently disseminate item information and sufficiently make use of them for query answering is one of the main challenges in the field of information retrieval in unstructured P2P networks.

On the other hand, k-walkers based searching strategies will almost fail if relevant items are not reachable under TTL limitation. Data replication strategies are able to break this bottleneck via demand-driven transitive replication of items. It is known that the classic k-random walkers combined with the non-semantic near-optimal replication strategy P2R2 [24] or the semantic replication scheme DSDR [4] can achieve much better search performance for rare items than its running alone. An open question is whether the search performance of a semantic search can be to what extent improved by its combination with a non-semantic or semantic replication strategy.

To this end, we propose S2P2P, a k-walkers based semantic search scheme for item information dissemination and query routing in unstructured P2P networks. Based on its demand-driven selective item information dissemination strategy, each peer in S2P2P is able to suggest a path for query routing, which contains as many known expert peers as possible, instead of mere a direct neighbor or remote peer. S2P2P scheme is agnostic to the kind of semantic description of data items and the data similarity measurements used by each peer for item selection. Our experimental evaluation evidences the outperformance by S2P2P in terms of search precision@recall and averaged precision regardless of item popularity distribution, in comparison with a selective flooding based search approach [16]. In addition, our experiments also reveal that S2P2P is as least as robust against network topology dynamics than [16].

Moreover, we present our experimental investigation on the performance of semantic search S2P2P and non-semantic k-random walkers when each of the both is combined with different replication strategies DSDR and P2R2. The result shows that S2P2P combined with semantic replication scheme DSDR achieves the best averaged precision in comparison with the run of S2P2P without replication and its combination with the non-semantic replication strategy P2R2 as well as k-random search with P2R2 or DSDR.

In the following Sect.II, we introduce the background and definitions for understanding our approach, which is detailed in Sect.III. The experimental evaluation results are presented

in Sect.IV. We discuss the related work in Sect.V and conclude this paper in Sect.VI.

II. Background and Preliminaries

In this section, we briefly introduce the background knowledge, definitions and assumptions which are necessary to understand our approach.

A. Semantic Similarity

The classic way of describing data (e.g. a 3D model) residing in the Internet is by natural language and meta-properties, such as name, author, etc.. Lots of text similarity measures like strict string/word matching, n-gram, etc., have been derived for the purpose of retrieval. Inspired by the growth of Semantic Web [3], more and more data is described with semantic annotations in different logic-based formalisms, in terms of their formation, content, property and functionalities. A merit of semantically annotated data is that logic-based reasoning can be applied to their machine understandable descriptions, and therefore alleviate the risk of mismatching due to the ambiguity and multiplicity of word meaning. Logical concept subsumption determination [21] can be applied for the similarity measure of data annotated with conceptual description (e.g. by concepts in a standard OWL2[1] ontology). Once data annotation is in the form of RDF[2], graph pattern matching based similarity measures like [7] are preferred. A formal way of describing the functionalities and interacting behaviors of data (e.g. the transportation service of a conveyor belt 3D model) is to specify proper semantic service in standard description languages, like OWL-S[3]. For comparing data in terms of functionalities, semantic service matchmaking techniques, such as [13], [14], are often used.

We assume that item selection process (for query answering) of each peer is capable of determining the relevance between a query and local items described in one or a combination of these formalisms above. In addition, the semantic similarity function $sim(\cdot,\cdot) \in [0,1]$ (cf.Sect.III) of S2P2P is designed to be agnostic to the kind of semantic description, which facilities the adoption of our approach to difference systems with customized concerns. Further, we assume that all peers share a minimal vocabulary of primitive concepts, roles and predicates, out of which each peer p can canonically build its local knowledge base KB_p (e.g. a local ontology O_p of p) for specifying query and item in needed formalism.

B. Preliminaries

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 $i = \langle td, sd, URI, pid, isz, n_s, da\rangle$ where td is the text-based description of i; sd the semantic annotation of i based on the local knowledge base KB_p of peer p; URI the item identifier; pid the id of the owner peer p providing i (including $i.da$); da the item data (e.g. mpeg file of a movie); isz the size of $i.da$; and n_s the number of available copies of i at p. The item tuple without the item data $i.da$ is called metadata or *item description* ($i.desc$) of i. ■

By k-walkers based search, a requester peer delegates a user query to a set of k ($k > 0$, $k \in \mathbb{N}$) walkers. With a time-to-live (TTL) limitation, each walker is forwarded by a peer with S2P2P query routing strategy to one of its direct neighbor peers according to the routing path suggestion. In case that the TTL value of a walker is exhausted, the walker backtracks along the inverted path targeting to the requester peer. The latter subsequently determines the satisfaction of the request based on the selected items by walkers. If the request is determined to be satisfied (unsatisfied), each of the k walkers is set with a $Success$ ($Fail$) status (cf. Def.2).

Definition 2: *Query, query satisfaction.*
A *query* q of a peer req is defined by the query tuple $q = \langle td, sd, req, A, Pa, Pa_{sug}, t, st, pbd, TTL, n_d\rangle$ where td denotes the query keyword (or topic of the query item); sd the semantic annotation used to describe the semantics of the requested item; req the identifier of the requesting peer; $A = \{(res, its)\}$ 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 path of this query; Pa_{sug} the suggested path, which consists of a sequence of peers for routing q afterwards. It is empty, if no suggestion is available or q is backtracking; 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 set of a query; TTL the query time-to-live value; n_d the requested number of copies of the query item. ■

We assume that each peer can provide and request any data item from known peers under the copyright restriction, which in our context limits the number of replicas of an item an individual peer can supply ($i.n_s$ in Def.1) or request ($q.n_d$ in Def.2). The item information dissemination process (cf.Sect.III-D) of each peer, in our context, is propagating item description only (cf. $i.desc$ in Def.1.) rather than the whole item with actual data file. As S2P2P is designed to be not adhering to specific kind of data description, it is convenient to use the terminology *semantic topic* (abbr. *topic*) tp, instead of those formalism-sensitive terms, such as *concept*, *semantic service*, etc., when describing the semantics of an observed query q or item i. Each peer p maintains a set of topics $TP_d(p)$ ($TP_s(p)$) (cf.Sect.III-B) derived based on the semantic descriptions of queries and items. In the following, we define the topic of observed demand (supply) and introduce the construction of $TP_d(p)$ ($TP_s(p)$) in Sect.III-B.

Definition 3: *Topic of demand (supply) observed by peer* p.
A *topic* tp_d (tp_s) of demand (supply) observed by peer p is defined by the topic tuple: tp_d (tp_s) = $\langle tsd, t_{lst}, P\rangle$, where tsd denotes the semantic description of this topic; t_{lst} the receiving time of the latest query (item description) that semantically similar to tp_d (tp_s); P is a list of membership entries. Each entry $(p', Q_{p'}, str)$ stands for the membership of a remote peer p' to a topic tp_d (tp_s). $Q_{p'}$ ($I_{p'}$) is the set of queries $Q_{p'}$ (item descriptions $I_{p'}$) issued by (propagated from) p' and sufficiently semantically similar with topic tp_d (tp_s). str ($str \in \mathbb{R}$) is the overall strength value of the observed demand

[1]http://www.w3.org/TR/owl2-overview/
[2]http://www.w3.org/RDF/
[3]http://www.w3.org/Submission/OWL-S/

(supply) of p' on topic tp_d (tp_s). Each remote peer corresponds to at most one entry $(p', Q_{p'}, str) \in P$. ■

III. Semantic Search Scheme

A. Overview

Besides local item selection, each peer p in unstructured P2P network performs path suggestion-based query routing as well as selectively propagates the descriptions of its known items via query piggybacked data. On receiving a query q, the local item selection process of p adds the item description $i.desc$ of a known item i to the query answer set $q.A$, once i is determined to be similar with q.

On forwarding a query q, each peer p suggests not only an immediate neighbor peer to which q should be forwarded but a path of peers with expertise that is relevant to the demand topic of q. Briefly, p computes the routing path suggestion with maximal total expertise gain with respect to the query demand topic $q.sd$ under TTL restriction. It contains as many relevant expert peers as possible. This is done by our greedy path augmenting algorithm that iteratively applies Dijkstra's algorithm for finding next expert peer in order to augment the current suggest path. If q contains a non-empty path suggestion $q.Pa_{sug}$, p adjusts $q.Pa_{sug}$ by means of comparing the total expertise gains of $q.Pa_{sug}$ with a new path Pa'_{sug} computed by p. The latter considers the relevant expert peers known by p together with the ones suggested by $q.Pa_{sug}$. This results in a new path suggestion for routing q.

In addition, each peer p is able to selectively propagate its known item information along the path of a query q, which is currently being forwarded. For this purpose, a subset $D(p, i)$ of peers on the (suggested) query path are selected as the destinations for the propagation of the description of a known item i. The completion of destination peer selection triggers the copy operation on i to the piggybacked data of q, which transmits the item (supply) information to remote peers. In case that a peer p receives a query q that is backtracking, p maintains a copy of the item description i from the query piggybacked data set $q.pbd$, if p is the dissemination destination of i.

B. Peer Local Observation

The path suggestion based query routing and item description dissemination processes performed by each peer p are depending on p's local observation in terms of queries (demands) and items (supplies). The queries comprise the ones issued or routed by p, while the items mean the item descriptions propagated to p or the items owned by p. On top of this, the local observation of p can be derived as follows:

p's local view to network topology: $G(p) = (V(p), E(p))$ where $V(p)$ denotes the set of known peers by p (including p itself); $E(p)$ the set of known direct connections between peers in $V(p)$.

A set of topics of demands observed by p: $TP_d(p)$. Each peer p maintains a set $TP_d(p)$ of demand topics based on the continuously observed queries. For this, a revised version of k-nearest neighbor clustering is applied when a new query q is observed. In addition, we apply a simple sliding time window strategy that collects the queries received during the last t_0 time units (e.g. in last t_0 mins). Let $Q(t_0)$ the set of queries observed in time window t_0; δ ($\delta \in (0, 1]$, $\delta \in \mathbb{R}$) the similarity threshold; $rt(q)$ the receiving time of q:
(i) If $TP_d(p) = \emptyset$, p creates a new topic $tp_d = \langle tsd, t_{lst}, P\rangle$, where $tsd = q.sd$, $t_{lst} = rt(q)$ and $P = (q.req, \{q\}, str)$. The computation of str will be presented later on.
(ii) If $TP_d(p) \neq \emptyset$, p computes a set $Q_k(q)$ containing at most k queries from $Q(t_0)$, which demands are most semantically similar with $q.sd$ and $sim_{sd}(q.sd, q'.sd) > \delta$ holds for each $q' \in Q_k(q)$. If $Q_k(q) = \emptyset$, create a new topic for q. Let $TP_d(p, q) \in TP_d(p)$ be the subset of involved topics. Each contains at least one query in $Q_k(q)$. The final topic tp^*_d for q is determined by:

$$tp^*_d = maxarg_{tp_d \in TP_d(p,q)}\Big\{ \sum_{q' \text{ in } tp_d} sim(q'sd, q.sd)\Big\}. \quad (1)$$

(iii) If the nearest topic tp^*_d of q has been determined during step (ii), p updates the triple of tp^*_d: $tsd = q.sd$, $t_{lst} = rt(q)$. If $tp^*_d.P$ does not contain a triple corresponding to $q.req$, p adds a new triple $(q.req, \{q\}, str)$ for the requester peer of q; otherwise updates the existing triple $(q.req, Q_{q.req}, str)$ by adding q to $Q_{q.req}$ and recalculating the demand strength str of peer $q.req$:

$$str(q.req, tp^*_d) = \frac{\sum_{q \in Q_{q.req}} q.n_d}{\sum_{\forall (p', Q_{p'}, str) \in tp^*_d.P} \sum_{q' \in Q_{p'}} q'.n_d}. \quad (2)$$

Expertise of peer $\mathbf{p' \in V(p)}$ on topic $\mathbf{tp_s}$: Each peer iteratively constructs the supply topic set $TP_s(p)$ based on its observed items $I(p)$. On receiving the disseminated item description of i, p executes an iteration to classify i to a topic in $TP_s(p)$, which is similar with the formation of $TP_d(p)$ above. k semantically nearest items $I_k(i)$ are selected from $I(p)$ but without considering the time window. Instead, the temporal factor of receiving i is taken into account during the estimation of the trustiness $tr(i, p)$ of i:

$$tr(i, p) = [tcr - tfd(i)]^{-1} \cdot topoDist(i.pid, p)^{-1}. \quad (3)$$

where $tfd(i)$ is the time point when the description of i is disseminated by its owner peer (cf.Def.1); tcr is the current time; while $topoDist(p, i.pid)$ is the topological distance from i's owner peer to p. It equals to the length of the concatenated path from $i.pid$ to p. $tr(i, p) = 1$, if p is the owner of i (cf.Def.5). $tr(i, p)$ measures (from p's local view) the availability of the disseminated information of i by its (dissemination latency) temporal and (concatenated path length) spatial factors. On top of this, the expertise $exp(p', tp_s)$ of a remote peer p' on each topic $tp_s \in TP_s(p)$ can be estimated:

$$exp(p', tp_s) = \sum_{i \in I_{p'}} \big\{tr(i, p) \cdot sim(i, tp_s.tsd)\big\}. \quad (4)$$

C. Semantic Query Routing

On forwarding a query q, instead of introducing mere one immediate neighbor to which q will be forwarded, each peer p computes a path Pa_{sug} as a suggestion for routing q (within TTL limitation). It contains a sequence of peers $p_1, \ldots, p_n$ that have expertise with respect to answering q. By properly selecting and arranging the order of known expert peers, our goal is to obtain a path with as much total expertise gain (cf.Def.4.) as possible. For this purpose, a heuristics is derived

(cf.Alg.1), which computes the path by iteratively applying Dijkstra's algorithm in a greedy manner.

Definition 4: *Expertise gain of peer p w.r.t. answering query q.*
Expertise gain $eg(p,q)$ measures the expertise of a peer p with respect to answering a given query q. It is determined by the peer expertise $exp(p,tp_s)$ on a topic tp_s as well as the semantic similarity between tp_s and $q.sd$.

$$eg(p,q) = max_{tp_s \in TP_s(p)}\{exp(p,tp_s) \cdot sim(q.sd, tp_s.tsd)\}. \quad (5)$$

The total expertise gain $eg_T(Pa,q)$ of peers in a path Pa w.r.t. answering q is the sum of $eg(p,q)$ for all peers p in Pa ■

Algorithm 1 $suggestPath(p, P_{exp}(q), q)$

1: $Pa_{sug} \leftarrow \{\}$;
2: $eg_T(Pa_{sug}, q) \leftarrow 0$;
3: $Cand \leftarrow P_{exp}(q)$;
4: $curMin \leftarrow \infty$;
5: $curBP \leftarrow null$;
6: $csp \leftarrow p$;
7: **for** each $p' \in Cand$ **do**
8: compute the shortest path $sPa(csp, p')$ from csp to p';
9: **if** $len(sPa(csp, p') < curMin)$ **then**
10: $curMin \leftarrow len(sPa(csp, p'))$;
11: $curBP \leftarrow p'$;
12: **end if**
13: **end for**
14: **if** $curBP \neq null$ **and** $len(Pa_{sug}) + len(sPa(csp, curBP)) \leq q.TTL$ **then**
15: concatenate $sPa(csp, curBP)$ to the tail of Pa_{sug};
16: $eg_T(Pa_{sug}, q) \leftarrow eg_T(Pa_{sug}, q) + eg(csp, q)$;
17: $csp \leftarrow curBP$;
18: $Cand \leftarrow Cand \setminus csp$;
19: $curMin \leftarrow \infty$;
20: $curBP \leftarrow null$;
21: **goto** line 7;
22: **else**
23: **return** Pa_{sug} and $eg_T(p,q)$;
24: **end if**

In Alg.1, we denote $sPa(p_i, p_j)$ the shortest path from peer p_i to p_j; Pa_{sug} the current path suggestion; $eg_T(Pa_{sug}, q)$ the total expertise gain of the peers in the current Pa_{sug}; $Cand$ the set of remote candidate peers that have expertise $exp(p', tp_s)$ on a topic tp_s, which is sufficiently semantically relevant to $q.sd$. $curMin$ the length of the current shortest path that will be used to augment Pa_{sug}; $len(\cdot)$ the function that computes the topological length of a path; $curBP$ the nearest expert peer from the current starting peer csp. Alg.1 builds path suggestion iteratively. Taking p itself as the starting peer for building the path suggestion, p computes the nearest expert peer $curBP$ (lines 7 – 13). Subsequently, if the length of path that is being formed is smaller than current $q.TTL$, p concatenates the sub-path from the starting peer to $curBP$ with Pa_{sug} and updates the total expertise gain (lines 14–19). This triggers a new iteration (line 21) which regards $curBP$ in the last iteration as the new starting peer. The algorithm runs until the length of concatenated path is larger than $q.TTL$ or no new expert peer is found.

In order to decide the candidate expert peers $P_{exp}(q)$ as an input for Alg.1, p computes a subset of topics $TP_s(p,q)$ from $TP_s(p)$ by measuring the semantic-based similarity of $q.sd$ with each supply topic $tp_s \in TP_s(p)$. If $sim(tp_s.tsd, q.sd)$ is larger than a threshold θ ($\theta \in (0,1], \theta \in \mathbb{R}$), p adds the peers in $tp_s.P$ (cf.Def.3.) to $P_{exp}(q)$.

In case that q contains a path suggestion Pa'_{sug} associated with a total expertise gain value $eg_T(p^*, q)$ made by some peer p^* before, p updates its local view on network topology based on Pa'_{sug} and recomputes a path suggestion Pa_{sug} with its corresponding total expertise gain $eg_T(p,q)$. If $eg_T(p^*, q) < eg_T(p,q)$, p routes q according to the new path suggestion; otherwise, q will be routed according to the old path suggestion. p randomly forwards q to one of its immediate neighbor, if $q.Pa_{sug} = \{\}$ and no path suggestion can be made by p based on its current knowledge.

D. Item Information Dissemination

According to S2P2P scheme, each peer p is able to perform demand-driven item description dissemination to remote peers. This process is triggered by the completion of the query routing decision (cf.Sect.III-C) and finished before the query is forwarded. Inspired by [26], without issuing any specific message, the transmission is done by wrapping the description of an item i into a data structure called item dissemination package i_{dp} (cf.Def.5) and copying i_{dp} into the piggybacked data of a query being forwarded. When receiving a query q that is backtracking, p checks the piggybacked item descriptions. It keeps a copy of the description of an item $i \in q.pbd$ if p is in the receiver set (cf.Def.5.) of i. Subsequently p updates its local knowledge of remote peer expertises.

Definition 5: *Item dissemination package of item i.*
The item dissemination package i_{dp} of an item i being propagated is defined by a tuple: $i_{dp} = \langle idesc, tfd(i), rcv, Pac\rangle$ where $idesc$ is the item description of i; $tfd(i)$ the time point when i is disseminated by its owner peer (cf.Def.1); rcv the receiver peer set of this package; and Pac the concatenated path from the item owner peer to the current peer that is initializing this package. ■

Peer selection for dissemination: Each peer p decides to propagate an item i (description) to a set $D(p,i)$ of destination peers. The latter is a subset of peers selected out of the current query path $D_1(p,i)$ and the ones $D_2(p,i)$ in the suggested query path:
(i) For each $p' \in D_0(p,i) = D_1(p,i) \cup D_2(p,i)$, p estimates the semantic utility $U(p,p',i)$ value of propagating the description of i from p to p':
$U(p,p',i) = \frac{\sum_{tp_d \in TP_d(p,p')}(str(p', tp_d) \cdot sim(tp_d, i.sd) \cdot tr(i,p))}{\sum_{tp_d \in TP_d(p,p')} str(p', tp_d)}$,
$TP_d(p,p') = \{tp_d | tp_d \in TP_d(p)$ **and** $\exists (p'', Q_{p''}, str) \in tp_d.P$ s.t. $p'' = p'\}$,
where $TP_d(p,p')$ is the set of demand topics of p', which have been observed by p; $sim(tp_d, i.sd)$ is the semantic similarity between each topic $tp_d \in TP_d(p,p')$ and i; $str(p', tp_d)$ refers to the strength of the observed demand of p'; and $tr(i,p)$ is the trustiness of i according to the observation of p;
(ii) Select the top m ($m \in \mathbb{N}$, $0 \leq m < |D_0(p,i)|$) peers with maximal utility values from $D_0(p,i)$ as the set of receiver peers of i. In case that $m > |D_0(p,i)|$, all the peers in $D_0(p,i)$ will be selected ($D(p,i) = D_0(p,i)$).

Subsequently, for each item i which description will be propagated, p instantiates a dissemination package i_{dp}. It sets $rcv = D(p, i)$ and computes concatenated path Pac from p to $i.pid$. If p is the owner of i, $i_{dp}.Pac$ contains p itself only.

Remote peer expertise maintenance: On receiving a query q that is backtracking, p checks the item dissemination package i_{dp} of item i in $q.pbd$. If $p \notin i_{dp}.rcv$, p skips to react on i; otherwise p adds i to local observed item set $I(p)$ and updates its observed expertise of remote peers (cf.Sect.III-B).

E. Robustness

S2P2P requires minimal amount of messages to be exchanged to react on dynamic changes such as peers leaving or joining the network. The arrival of a peer in S2P2P enabled P2P network triggers a simple handshake-advertisement: A arriving peer p broadcasts a one-hop advertisement (TTL = 1) to peers in its neighborhood, and waits for acknowledgement-messages. If at least one peer answers, p considers itself to be online and both peers mutually add each other into their local view of the network topology. No action will be triggered by the departure of a peer p from the network. If a peer stops to answer messages, the other part p' of the communication will detect its absence. Subsequently, the local view on network topology on p' is updated.

Furthermore, the maintenance of the demand/supply topic sets on each peer enabled with S2P2P scheme behaves in a lazy manner under the network dynamics in terms of peer arrival/departure. Each allocated membership entry (cf.Def.3.) is associated with a boolean flag indicating its availability. If the absence of p is detected by p', the latter sets the flags of those entries about p with "0", instead of deleting them. It means that they will no longer be taken into account by query routing and item description dissemination, until the flags are set to "1". This would save the cost for memory/disk (re-)allocation, in case that the absence of p is caused by a temporary network disconnection. Likewise, when p' knows the arrival of p, p' searches for the membership entries of p in its maintained demand/supply topics structure. If there exist some entries of p, p' sets the flags of them to "1". If no entry about p has been found, p' does nothing, as no demand/supply of p is detect at this moment.

F. Complexity

As no extra message is needed for path suggestion and item information dissemination processes, we discuss the computational complexities of the both in this section. Let v (e) be the total number of peers (edges); m the total number of distinct demand topics; l the initial time-to-live value for each walker. For routing a query, each peer computes a path suggestion contains up to l expert peers on a requested topic. Each is computed via Dijkstra's algorithm that in worst case costs $\mathcal{O}(e+vlogv)$ [10]. Therefore, the computational complexity of the path suggestion for routing a query is $\mathcal{O}(l(e+vlogv))$. For disseminating the description of an item, each peer p computes the semantic utility values for at most l peers. For candidate destination peer p', p measures the semantic similarity between i and m demand topics (in worst case). As the computation for demand strength and supply trustiness is done in advance, the complexity of deciding the information dissemination for i is $\mathcal{O}(lm \cdot s)$ where s is the computational complexity of the semantic similarity measure used by the system. This can vary from polynomial to NEXP depending on the types of formalisms.

IV. Experimental Evaluation

We present and discuss the results of our preliminary comparative experimental evaluation of the performance of S2P2P search scheme in unstructured P2P networks with different configurations.

A. Experimental Settings

For our experiments, we created unstructured P2P networks with 10000 peers and topologies based on random graphs (RG) with averaged connectivity 3.2 and random power law graphs (RLPG). The latter is known to be a realistic model in particular for social networks. Each peer in our testing environment is implemented as an independent thread with capacity of interacting with other thread via IP layer. Each experiment is conducted by employing two computers: a standard PC with COREi7 2.8GHz CPU, 8G RAM and a laptop with COREi7 2GHz CPU, 4G RAM, which support 7000 and 3000 threads, respectively. 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. The initial value of TTL of each walker in S2P2P is 10 and the number of walkers is 4. Both similarity thresholds δ and θ are 0.5. The time window size t_0 is set to be 600 seconds.

As a test collection we use a random subset of 20k RDF linked data items (in files: instance_types_en.nt.bz2 and mappingbased_properties_en.nt.bz2) taken from DBpedia[4] 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. Since DBpedia does not provide the relevance sets for item queries, we use the following heuristics for relevance judgments: Item i about concept $i.sd = \tau(C)$ is relevant (a true positive) for query item i' about concept $q.sd = \tau(C')$, if any of the logic-based concept relations in $\{C \equiv C', C \sqsubseteq_1 C', C \sqsupseteq_1 C'\}$ holds. For query-item similarity determination, each peer simply checks the data concept subsumption relations based on mere concept hierarchy but ignores the matching on properties.

B. Evaluation Metrics

Let Q the set of queries in the network; I_q ($I_{q,j}$) the set of items collected by 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} ($I^*_{tq,j}$) the set of relevant items for q at all peers (the j-th peer) on the query path;

- 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

[4] http://downloads.dbpedia.org/3.7/en/

$\langle pre_{q,m}, re_{q,m} \rangle$ 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 $ap = \frac{1}{|Q|} \sum_{q \in Q} \frac{|I^*_q|}{|I_q|}$.

C. Comparative Approaches

We compare S2P2P with INGA [16] in terms of search performance and robustness. The latter introduces a shortcut based restricted semantic flooding search strategy. Particularly, each peer of INGA system creates a semantic network overlay (shortcuts of data) by query analysis, which is the common feature with S2P2P. We implemented two global table data structures for the access of shortcuts in content provider and recommender layers. In addition, both the maximum fanout k and $maxTTL$ of the flooding are set with 3. This setup ensures that a query of INGA can traverse about 40 peers in a random graph based network with averaged connectivity 3.1 (cf.Sect.IV-A). It is fair to S2P2P based query, which issues k=4 walkers with initial TTL=10 each.

For testing the search performance of the combination of S2P2P with data replication strategies, we choose the combinations of the same k-random search with different data replication strategies: a near-optimal non-semantic replication strategy P2R2 [24] and a semantic data replication scheme DSDR [4].

D. Experiment Results

Experiment 1 (Search performance): We compare the search performance of S2P2P and INGA in random graph based network. The item popularity distributions are uniform at random (R) distribution over all items and Zipf (Z) distribution ($\beta = 1.03$) over pre-clustered 79 topics. Our experiments revealed that S2P2P can significantly outperform INGA in terms of macro-averaged precision at recall (cf.Fig.1.) and averaged precision (cf.Fig.2.) regardless of the kind of item popularity distribution. Particularly, it achieved around 24% more precision at intermediate recall levels (cf.Fig.1.) and around 20% more averaged precision (cf.Fig.2.). The reason is that an INGA-enabled peer can not transitively propagate its detected shortcuts information; while a peer in S2P2P system is able to propagate the received item description information to those groups of peers located topologically farther. In the latter system, a request is therefore having a higher chance of meeting more relevant items. This merit of S2P2P effects when the maximal numbers of peers a (S2P2P or INGA) query can access are similar (cf.Sect.IV-C). In addition, both semantic search strategies appear to be relatively not sensitive to the kind of item popularity distribution. This is caused by a feature of search: the item information dissemination is mainly driven by the existence of item than the observed demands. In INGA system, the detecting flooding for building shortcuts is issued independently from the query.

Experiment 2 (Robustness): Our second experiment analyses the robustness of S2P2P and INGA for networks with random graph-based topology. After the processing of 8k and 18k queries, we randomly delete 25% peers from the network while

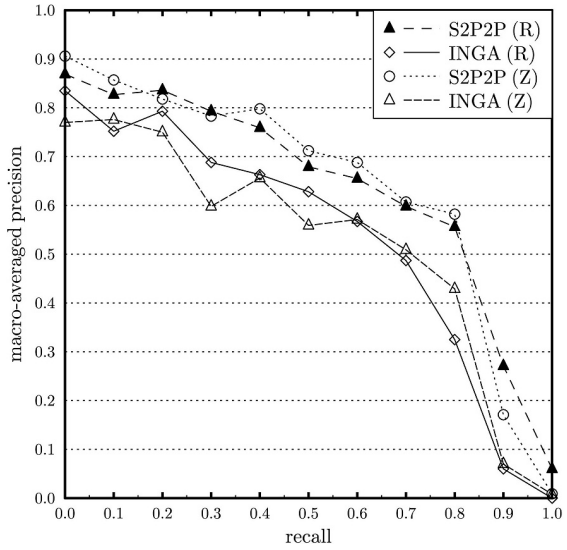

Fig. 1: Macro-averaged precision at recall of S2P2P and INGA.

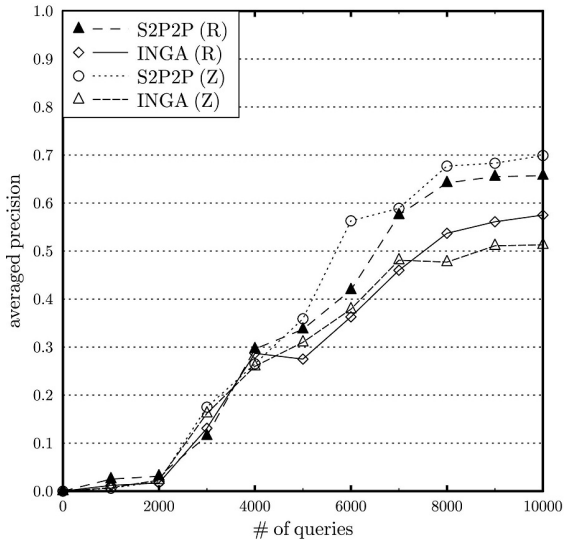

Fig. 2: Averaged precision of S2P2P and INGA.

add them randomly to the network after 15k queries were processed. The results reveals that the departure of peers results in a decrease of precision, since the semantic overlay structure was partially destroyed through peers leaving the network. The precision of both systems is not sensitive to the arrival of peers. Although the shortcuts of INGA or disseminated item information of S2P2P are diluted by the arrival of peers, the knowledge of the shortcuts and the paths targeting to disseminated items are remained. The averaged precision of both semantic search methods drops at each departure event (Fig.3) but both systems were able to recover within almost the same time period.

Experiment 3 (Search combined with replication): In experiment 3, we test the search performance of (non-)semantic search approaches combined with replication strategies in a network with random power law graph-based topology. The configurations includes the run of S2P2P without replication, and the runs of k-random search (abbr. KW. k=4, TTL=10) or S2P2P combined with the non-semantic replication method

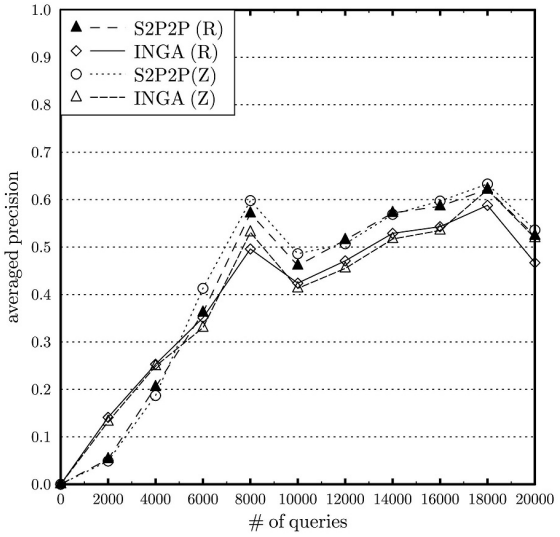

Fig. 3: Robustness: Averaged precision of S2P2P and INGA.

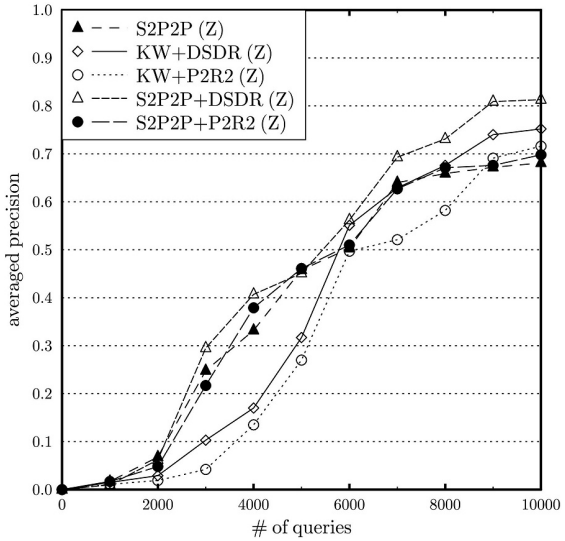

Fig. 4: Averaged precision of S2P2P, KW+DSDR, KW+P2R2, S2P2P+DSDR, S2P2P+P2R2.

P2R2, as well as the same searches with the semantic replication scheme DSDR. The experiment result evidences the effectiveness of data replication, which is able to increase the performance of search in unstructured P2P networks. Particularly, the combination of S2P2P with semantic data replication strategy DSDR yields best precision after its stable network overlay has been established. In comparison with the combination of S2P2P with P2R2, the incentive by DSDR is larger. The reason is that a P2R2 enabled peer needs to know the query satisfaction of its observed query in order to judge the replication. However, this information is not provided by S2P2P peer as the query satisfaction is determined by the requester peer. In contrast, the group formation and subsequent replication decision of DSDR are conducted by a peer which request was judged to be unsatisfied by itself. This leads to adequate information to perform replication decision. Not surprisingly, k-random search can take advantage of P2R2 because of the syntactic matching happened during each peer's item selection process. It directly provides the information of "hit", by which the peer concludes the demand strength for its data replication based on P2R2.

Moreover, the evaluation shows that the search precision of S2P2P enabled configurations increases relatively more faster than the k-random search based systems. The reason is that S2P2P is capable of forming its own semantic overlay for conducting query routing, but k-random search can not. This is evidenced by the independent run of S2P2P without replication. However, this result also reveals that k-random search can be more robust under the network dynamics because of its capacity of working without semantic overlay. Further, after a sufficiently large number of queries (6000 in Fig.4.), the precision of k-random search combined with the both replication methods increases faster than the S2P2P-enabled system without replication. This evidences the merit of replication strategy, which can transitively propagate the item data (not only the item description) to remote peers. This is not achievable by S2P2P since the latter disseminates item description only.

V. Related Work

Majority of search strategies in unstructured P2P networks is mainly the variants of the classic flooding- or walker-based search. In the following, we present representative solutions from each of the both.

Flooding-based search strategies, such as [11], [16], [15], [18], commonly offer high search recall but suffer from relatively large network traffic cost. In the expertise-based semantic search Bibster [11], each peer advertises the topics of its maintained items via TTL-bounded flooding. Semantic links are distributively built among peers. Each peer is therefore enabled to route a query to at most s peers which are expertise on similar topic. Besides the risk of excessive message load in the network, a limitation of this approach comes into that the quality of semantic links is subject to network dynamics, as the links are built in one shot at the stage of peer arrival. In contrast, S2P2P maintains the expertise of peers dynamically in line with the query routing. In P-grid system [1], a virtual distributed search tree is holding by peers. A binary query is routed from a peer to at least one peer whose encoded expertise is determined to be "closer" to the query. Löser et.al. proposes a shortcut-based approach INGA [16], which enables a peer to perform selective flooding relying on the overlay built upon query analysis. A shortcut of the item on a remote peer is created if a query gets answers from it or detected by flooding. This would lead to large amount of comparisons during shortcut selection strategy for forwarding a query. The effort [18] introduces probabilistic flooding strategy aiming at minimizing the cost of excessive message transmissions via the proposed hop distance measure. Whether a peer forwards a query to its neighbors depends on the probability of this query hitting a matched resource. The latter is estimated based on the ratio of the nodes flooded over all in the network. The underlying assumption is that the resource distribution is uniformly random, which is not always true in practice.

K-walkers based search strategies commonly generate much less network traffic than the former flooding based variants. Biased query routing is always a feature of them. For

this, machine learning, query analysis, item information dissemination, etc. techniques are used in order to build/maintain a network overlay, by which a peer obtains more information for routing decision. In RS2D [2] and ACS [23], query routing relies on the learned network overlay, which is achieved by training the system with a set of labeled queries or by collaborative graph investigation. Likewise, peer in [19] is enabled to route a query to one of its semantic neighbor peers according to the local view of semantic network overlay. The latter is established by broadcasting peer profile when each peer joins in the network. The search performance of those systems built by means of similar ways is subject to the network dynamics. Liu et.al. proposes a method [15] to build super-peers that form a semantic overlay by enabling peers to cache the data of in particular the popular items according to storage capacities of peers. Despite the increment of search precision/recall, hot spots are prone to appear in the network. The effort [27] presents a path-traceable query routing strategy based on the propagation of gains of query hits. For this, each peer analyzes the traversing queries and maintains a dynamic traceable gain matrix, which comes into the base of further query routing decision. In addition to the network load balance issue, the approach works relying on an assumption that a "hit" is a "match", which is not always true in decentralized retrieval system for complex objects, as the query satisfaction should be decided by the requester. Filali et.al. proposes walker-based search strategy [9] based on item information advertising and dynamic TTL heuristics. The latter offers incentive to a walker by decreasing TTL with a probability less than 1 when the walker finds relevant resource. Unfortunately, this would not to large extent increase the chance of find matching results, since the radio of a walker hitting a relevant result can be a constant if the overlay is fixed.

VI. Conclusion

In this paper, we presented a semantic based search scheme called S2P2P for item information dissemination and query routing in unstructured P2P networks. The main contribution is that S2P2P offers a novel query path suggestion heuristics, which provides a query with a sequence of expert peers on demand topics. Our experimental evaluation shows that S2P2P outperforms the semantic flooding based approach INGA in terms of search precision@recall and averaged precision. Besides, our experiment reveals that S2P2P search is at least as robust against network dynamics than INGA.

References

[1] Aberer, K.; Cudré-Mauroux, P.; Datta, A.; Despotovic, Z.; Hauswirth, M.; Punceva, M.; Schmidt, R. (2003): P-Grid: a self-organizing structured P2P system. *ACM SIGMOD Record*, 32(3), 29–33. ACM.

[2] Basters, U.; Klusch, M. (2006): RS2D: fast adaptive search for semantic web services in unstructured P2P networks. *International Semantic Web Conference*, 87–100. Springer.

[3] Berners-Lee, T.; Hendler, J.; Lassila, O. (2001): The semantic web. *Scientific american*, 284(5), 28–37. New York, NY, USA.

[4] Cao, X.; Klusch, M. (2012): Dynamic Semantic Data Replication for K-Random Search in Peer-to-Peer Networks. In proc. of the 11th IEEE international symposium on network computing and applications, 20–27. IEEE.

[5] Cheng, A.; Joung, Y. (2006): Probabilistic file indexing and searching in unstructured peer-to-peer networks. *Computer Networks*, 50(1), 106–127. Elsevier.

[6] Datta, A.; Sharma, R. (2011): GoDisco: selective gossip based dissemination of information in social community based overlays. *Distributed Computing and Networking*, 227–238. Springer.

[7] De Virgilio, R.; Maccioni, A.; Torlone, R. (2013): A similarity measure for approximate querying over RDF data. In proc. of the joint EDBT/ICDT 2013 workshops, 205–213. ACM.

[8] Dorigo, M.; Gambardella, L.M. (1997): Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66. IEEE.

[9] Filali, I.; Huet, F. (2010): Dynamic TTL-based search in unstructured peer-to-peer networks. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), 438–447. IEEE.

[10] Fredman, M. L.; Tarjan, R. E. (1987): Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3), 596–615. ACM.

[11] Haase, P.; Broekstra, J.; Ehrig, M.; Menken, M.; Mika, P.; Olko, M.; Plechawski, M.; Pyszlak, P.; Schnizler, B.; Siebes, R. (2004): Bibster–a semantics-based bibliographic peer-to-peer system. In proc. of the 3rd international semantic web conference, 122–136. Springer.

[12] Kermarrec, A.; Van Steen, M. (2007): *ACM SIGOPS Operating Systems Review*, 41(5), 2–7. ACM.

[13] Klusch, M.; Kapahnke, P. (2012): Adaptive Signature-Based Semantic Selection of Services with OWLS-MX3. *Multiagent and Grid Systems*, 7. IOS Press.

[14] Klusch, M.; Kapahnke, P. (2012): The iSeM Matchmaker: A Flexible Approach For Adaptive Hybrid Semantic Service Selection. *Web Semantics*, 15. Elsevier.

[15] Liu, Y.; Wu, S.; Xiong, N.; Park, J.; Zhang, M. (2012): A cache-based search algorithm in unstructured P2P networks. *Journal of Intelligent Manufacturing*, 23(6), 2101–2107. Springer.

[16] Loser, A.; Staab, S.; Tempich, C. (2007): Semantic social overlay networks. *IEEE Journal on Selected Areas in Communications*, 25(1), 5–14. IEEE.

[17] Lv, Q.; Cao, P.; Cohen, E.; Li, K.; Shenker, S. (2002): Search and replication in unstructured peer-to-peer networks. In proc. of the 16th international conference on supercomputing, 84–95. ACM.

[18] Margariti, S. V.; Dimakopoulos, V. V. (2011): A Novel Probabilistic Flooding Strategy for Unstructured Peer-to-Peer Networks. In proc.of the 15th panhellenic conference on informatics, 149–153. IEEE Computer Society.

[19] Mawlood-Yunis, A.; Weiss, M.; Santoro, N. (2010): From p2p to reliable semantic p2p systems. *Peer-to-peer networking and applications*, 3(4), 363–381. Springer.

[20] Michlmayr, E. (2006): Ant algorithms for search in unstructured peer-to-peer networks. In Ph.D. Workshop, 22nd International Conference on Data Engineering, 1–7.

[21] Nebel, B. (1990): *Reasoning and revision in hybrid representation systems.* Springer-Verlag Heidelberg, Germany.

[22] Patel, J.; Gupta, I.; Contractor, N. (2006): JetStream: Achieving predictable gossip dissemination by leveraging social network principles. In proc. of the 5th IEEE international symposium on network computing and applications, 32–39. IEEE.

[23] Santillán C.; Reyes, L.; Conde, E.; Schaeffer, S.; Valdez, G. (2010): A Self-Adaptive Ant Colony System for Semantic Query Routing Problem in P2P Networks. *Computación y Sistemas*, 13(4), 433–448.

[24] Sozio, M.; Neumann, T.; Weikum, G. (2008): Near-optimal dynamic replication in unstructured peer-to-peer networks. In proc. of the 27th ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems, 281–290. ACM.

[25] Tempich, C.; Staab, S. (2006): Semantic query routing in unstructured networks using social metaphors. *Semantic Web and Peer-to-Peer*, 107–123. Springer.

[26] Vigfusson, Y.; Birman, K.; Huang, Q.; Nataraj, D. (2009): GO: Platform support for gossip applications. In proc. of the 9th IEEE international conference on peer-to-peer computing, 222–231. IEEE.

[27] Xu, M.; Zhou, S.; Guan, J.; Hu, X. (2010): A path-traceable query routing mechanism for search in unstructured peer-to-peer networks. *Journal of Network and Computer Applications*, 33(2), 115–127. Elsevier.