Content Aggregation on Knowledge Bases using Graph Clustering

Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme

Knowledge and Data Engineering Group, Universität Kassel
{lastname}@cs.uni-kassel.de
http://www.kde.cs.uni-kassel.de

Abstract. Recently, research projects such as PADLR and SWAP have developed tools like Edutella or Bibster, which are targeted at establishing peer-to-peer knowledge management (P2PKM) systems. In such a system, it is necessary to obtain provide brief semantic descriptions of peers, so that routing algorithms or matchmaking processes can make decisions about which communities peers should belong to, or to which peers a given query should be forwarded.

This paper provides a graph clustering technique on knowledge bases for that purpose. Using this clustering, we can show that our strategy requires up to 58% fewer queries than the baselines to yield full recall in a bibliographic P2PKM scenario.

1 Introduction: Ontology-Based P2PKM

Recently, a lot of effort has been spent on building peer-to-peer systems using semantic web technology [22, 5, 2, 15], based on a notion of peer-to-peer based, personal knowledge management (P2PKM for short). In such a scenario, users will model their knowledge in personal knowledge bases, which can then be shared with other users via a peer-to-peer network.

Many use cases for P2PKM have been implemented recently. In the PADLR and ELENA projects[1], a P2P infrastructure is established for the exchange of learning material; Bibster[2] is a tool for sharing BibTeX entries between researchers; the SCAM tool[3] for knowledge repositories connects to a P2P network. In these systems, each peer builds a knowledge base on top of a common ontology such as LOM and ACM CCS.

One crucial point in such a P2P network is that query messages need to be *routed* to peers which will be able to answer the query without flooding the network with unnecessary traffic. Several proposals have been made recently as to how the network can self-organize into a topology consisting of communities around common topics of interest, a structure which is beneficial for routing, and how messages can be routed in this topology [20, 21, 8, 23]. All of these are based on the idea of routing indices [3]. In a routing index, peers store an aggregated

[1] http://www.l3s.de/english/projects/projects_overview.html

[2] http://bibster.semanticweb.org

[3] http://scam.sourceforge.net/

view of their neighbors' contents, enabling them to make content-based routing decisions.

One missing link towards these self-organized network topologies is the extraction of expertises – semantic self-descriptions – of peers from the peers' knowledge bases. In this paper, a method of extracting these expertises using a clustering technique on the knowledge base is proposed and evaluated.

The remainder of this paper is structured as follows: After a brief review of an ontology-based P2P knowledge management scenario and related work, we will introduce technical preliminaries in Section 2. In Section 3 the automatic generation of self-descriptions of peers' knowledge bases through the use of graph clustering will be shown. Section 4 presents evaluation results for a bibliographic P2PKM scenario. Section 5 concludes and discusses future work.

1.1 Related Work

To the best of our knowledge, the exact problem discussed in this paper has not been treated before. There are, however, related areas which touch similar topics.

Knowledge-rich approaches from the text summarization area [10, 9] use algorithms on knowledge representation formalism to extract salient topics from texts in order to generate summaries. We compare our approach to the one in [10] in Section 4.

In semantic P2P overlays, peers need some means of obtaining a notion of other peers' contents for routing tables and other purposes. [13] and others rely on observing the past behavior of peers – queries sent and answered – to guess what kind of information peers contain, including some fallback strategies to overcome the bootstrapping problem. In [8], peers publish their expertise containing *all* topics they have information about without any aggregation, which will be a resource consumption problem for larger knowledge bases and networks.

Keyword-based P2P information retrieval systems can make use of the bag-of-words or vector-space models for IR. [19] proposes the use of Bloom filters to maintain compact representations of contents for routing purposes. These techniques, however, do not provide a semantically aggregated view of the contents, but rather a bitwise superposition of keywords which loses semantic relationships between related keywords.

Much work has been done on graph clustering (e. g. [16]) in a variety of areas. Most of these algorithms, though, do not readily yield representatives such as the centroids from the k-modes algorithm used in Section 3, and/or may not be naturally adapted to the shared-part/personal-part consideration used in Section 2.3.

2 Basics and Definitions

2.1 P2P Network Model

As in [20], the following assumptions are made about about peers in a P2PKM network:

- Each peer stores a set of *content items*. On these content items, there exists a *similarity function* called *sim*. We assume $sim(i,j) \in [0,1]$ for all items i, j, and the corresponding *distance function* $d := 1 - sim$ shall be a metric. For the purpose of this paper, we assume *content items* to be entities from a knowledge base (cf. Section 2.2), and the metric to be defined in terms of the ontology as described in Section 2.4.
- Each peer provides a self-description of what its knowledge base contains, in the following referred to as *expertise*. Expertises need to be much smaller than the knowledge bases they describe, as they are transmitted over the network and used in other peers' routing indices. A method of obtaining these expertises is outlined in Section 3. Formally, an expertise consists of a set $\{(c_i, w_i) | i = 1 \ldots k\}$ of pairs mapping content items c_i to real-valued weights w_i.
- There is a relation *knows* on the set of peers. Each peer knows a certain set of other peers, i. e., it knows their expertises and network address (e. g. IP address, JXTA ID, . . .). This corresponds to the routing index as proposed in [3]. In order to account for the limited amount of memory and processing power, the size of the routing index at each peer is limited.
- Peers query for content items on other peers by sending query messages to some or all of their neighbors; these queries are forwarded by peers according to some *query routing strategy*, which uses the *sim* function mentioned above to decide which neighbors to forward messages to.

2.2 Ontology Model

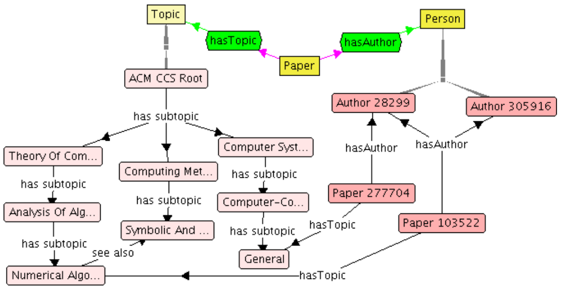

Fig. 1: Example Knowledge Base

For the purpose of this paper, we use the view on ontologies proposed by the KAON framework [6]. Following the simplified nomenclature of [6], an *ontology* consists of *concepts* with a subclassOf partial order, and relations between concepts. A *knowledge base* consists of an ontology and *instances* of concepts and relations. Concepts and instances are both called *entities* (for details cf. [6]).

Another important feature of KAON is the inclusion mechanism for knowledge bases, enabling the implementation of the shared and personal parts of knowledge bases as introduced in the next section.

2.3 Shared and Personal Parts of the Knowledge Bases

Based on the use cases mentioned in Section 1, all peers $P_i, i = 1 \ldots n$, in the system are assumed to *share* a certain part O of their ontologies: in the case of e-learning, this could be the Learning Object Metadata (LOM)[4] standard plus a classification scheme; when exchanging bibliographic metadata as in Bibster, this would be an ontology reflecting BIBTEX and a classification scheme such as ACM CCS[5], etc.

Additionally, the knowledge base KB_{i} of each peer P_i contains *personal* knowledge PK_{i} which is modeled by the user of the peer and is not known a-priori to other peers. Querying this knowledge efficiently and sharing it among peers is the main task of the P2PKM system. Formally, we can say that for all i, $KB_{\mathrm{i}} = \mathrm{O} \cup PK_{\mathrm{i}}$.

In Figure 1, the ontology used in the evaluation in Section 4 is shown. In this case, the shared part O comprises the concepts Person, Paper, Topic, and their relations, as well as the topics of the ACM CCS. The personal knowledge PK_{i} of each peer contains instantiations of papers and persons and their relationships to each other and the topics for the papers of each individual author in DBLP with papers in the ACM digital library (cf. 4.1 for details).

For the purpose of this paper, an agreement on a shared ontology O is assumed. The problem of ontologies emerging in a distributed KM setting [1], of ontology alignment, mapping, and merging [4], are beyond the scope of this work.

2.4 Ontology-Based Metrics

An ontology of the kind we use is a labeled, directed graph: the set of nodes comprises the entities, and the relations between entities make up the set of edges. An edge between entities in this graph expresses relatedness in some sense: the instance `paper37` may have an instanceOf edge to the concept `Paper`, `Paper` and `Topic` would be connected by an edge due to the `hasTopic` relation, etc.

On this kind of semantic structure, [17] has proposed to use the distance in the graph-theoretic sense (length of shortest path) as a semantic distance measure.

Metric Used in the Evaluation We follow this suggestion and apply it to the abovementioned graph as follows:

– To each edge, a length is assigned; taxonomic edges (instanceOf, subclassOf) get length 1, while non-taxonomic edges are assigned length 2. This reflects the fact that `subclassOf(PhDStudent, Person)` is a closer link between these concepts than, say, `rides(Person, Bicycle)`.

[4] `http://ltsc.ieee.org/wg12`

[5] `http://www.acm.org/class`

- Edge lengths are divided by the average distance of the incident nodes from the root concept. This reflects the intuition that top-level concepts such as `Person` and `Project` would be considered less similar than, e.g., `Graduate Student` and `Undergraduate` farther from the root.

Similarity, Relatedness, and Semantic Distances – Why Edge Counting? The notions of semantic similarity (things having similar features) and relatedness (things being associated with each other) have long been explored in various disciplines such as linguistics and cognitive sciences. Discussions about these phenomena and their respective properties have lasted for decades (cf. [24, 7]). While most of this discussion is outside the scope of this paper, some key points [7] are worth mentioning: Thematic relatedness and similarity are distinct phenomena, but both can get mixed up or influence each other.

In the context of this paper, where the goal is to provide self-descriptions of knowledge in a P2PKM system, some more influences on the choice of the semantic distance should be noted:

- The ontologies to be used in P2PKM will be engineered specifically for KM purposes. Thus, regarding a relation between two concepts as an indication that these two have something to do with each other reflects the intention of a knowledge engineer to express relatedness.
- In a P2PKM system, domain specific ontologies will be used. These represent a conceptualization of a small part of the world which is relevant for the given domain, so that stray associations such as *lamp – round glowing object – moon – . . .*, which might occur in a "world ontology", will be avoided.
- Modeling idiosyncrasies of certain tools and formalisms such as described in the next section need to be anticipated. This can be done by allowing for flexible weighting and filtering strategies.

Various constraints are present on other kinds of metrics which have led to the use of an edge-counting metric for the purpose of this paper. Approaches such as [18] or [24] assume the presence of full text or linguistic background knowledge; others such as [14] only use concepts and an instanceOf relationship, neglecting instances and non-taxonomic relationships altogether. To yield maximum flexibility and to use as much of the modeled content as possible, an edge counting approach was chosen for this paper.

Keeping this discussion in mind, one needs to be aware of what kinds of similarity and/or relatedness should be expressed in modeling the ontology and parameterizing the metric.

Pitfalls on Real-World Ontologies While the edge-counting metric seems straightforward, applying it to real-world ontologies turned out to be non-trivial:

Noise and Technical Artifacts. Often not all of the content of a knowledge base is used to model a certain domain as such; e. g., in KAON, lexical information is represented as first-class entities in the knowledge base. This yields entities which are not relevant for the semantic distance computation. There is also a root class which every entity is an instance of, which would render our approach to calculating distances useless.

Modeling Idiosyncrasies. Engineering an ontology implies design decisions, e. g. whether to model something as an instance or as a concept [25]. These decisions carry implications for the weighting of edges, e. g. when taxonomic relationships are expressed by a relation which is not one of `instanceOf`, `subclassOf`.

To overcome these problems, we have implemented extensive entity filtering and weighting customization strategies which are applied prior to the metric computation itself.

Choice of Parameters One obvious question is where the parameters, weighting schemes and filtering rules necessary for this kind of metric should come from. These can be agreed upon just like the ontology to be used itself. When stakeholders deside that there should be a "see also" relation between topics, they could also agree on its importance or non-importance for retrieval tasks (cf. the discussion about the value of non-taxonomic relations in [17]).

Secondly, this kind of semantic metric will not primarily be used to reflect human judgment of similarity or relatedness directly, but to structure a network topology. For this type of use, optimal parameters can be determined in simulation experiments or might be learned over the lifetime of the system.

2.5 *k*-Modes Clustering

In Section 3, we will use an extension of k-modes clustering [11] to obtain aggregations of knowledge bases. The basic version of k-modes clustering for partitioning a set S of items into k clusters $S_1, \ldots, S_k$ such that $S = \dot{\bigcup}_i S_i$ works as follows:

1. Given k, choose k elements $C_i, i = 1 \ldots k$ of S as *centroids*
2. Assign each $s \in S$ to the cluster S_i with $i = \arg\min_j d(C_j, s)$
3. For $i = 1 \ldots k$, recompute C_i such that $\sum_{s \in S_i} d(C_i, s)$ is minimized.
4. Repeat steps 2 and 3 until centroids converge.

This algorithms yields (locally) optimal centroids which minimize the average distance of each centroid to its cluster members. A variation we will use is *bisection k-modes clustering*, which produces k clusters by starting from an initial cluster containing all elements, and then recursively splitting the cluster with the largest variance with 2-modes until k clusters have been reached.

As the algorithm is randomized, it may happen that a cluster cannot be split although k clusters have not been reached. In that case, we retry a fixed number of times before accepting the clustering.

3 Graph Clustering for Content Aggregation

As mentioned in the motivation, a peer needs to provide an expertise in order to be found as an information provider in a P2PKM network. From the discussion above, the following requirements for an expertise can be derived:

- The expertise should provide an aggregated account of what is contained in the knowledge base of the peer, meaning that using the similarity function, a routing algorithm can make good a-priori guesses of what can or cannot be found in the knowledge base. More specifically, the personal part PK_{i} should be reflected in the expertise.
- The expertise should be much smaller than the knowledge base itself, preferably contain only a few entities, because it will be used in routing indices and in computations needed for routing decisions.

With these requirements in mind, we propose the use of a clustering algorithm to obtain an expertise for each peer.

3.1 Clustering the Knowledge Base

We use a version of bi-section k-modes clustering for the extraction of such an expertise. As mentioned before, k-modes clustering yields centroids which are locally optimal elements of a set regarding the average distance to their cluster members.

Using the semantic metric, these centroids fulfill the abovementioned requirements for an expertise: We can compute a *small* number of centroids, which are – on the average – *semantically close* to every member of their respective clusters, thus providing a good *aggregation* of the knowledge base.

In order to apply this algorithm in our scenario, however, some changes need to be made:

- The set S to be clustered has to consist only of the *personal parts* PK_{i} of the knowledge bases. Otherwise, the structure of the shared part (which may be comparatively large) will shadow the interesting structures of the personal part.
- The *centroids* C_i will not be chosen from the whole knowledge base, but only from the shared part O of the ontology. Otherwise, other peers could not interpret the expertise of a peer.

The expertise for each knowledge base is obtained by clustering the knowledge base as described, obtaining a set $\{C_i \mid i = 1 \dots k\} \subseteq O$ of entities from the ontology as centroids for a given k. The expertise then consists of the pairs $\{(C_i, |S_i|) | i = 1 \dots k\}$ of centroids and cluster sizes. Because we restricted the choice of centroids to be from O, we get expertises that other peers can interpret from clustering the elements of KB_i.

3.2 Determining the number of centroids

One problem of the k-modes algorithm is that one needs to set the value of k beforehand. As the appropriate number of topics for a given knowledge base may not be known a-priori, we use the *silhouette coefficient* [12], which is an indicator for the quality of the clustering. In short, it determines how well clusters are separated in terms of the distances of each item to the nearest and the second nearest centroid: if each item is close to its own centroid and far away from the others, the silhouette coefficient will be large, indicating a good clustering.

4 Experimental Evaluation

In the following sections, we will try to verify three hypotheses:

1. Extracting a good expertise from a knowledge base is harder for large knowledge bases.
2. With larger expertises, the retrieval results improve.
3. The clustering strategy extracts expertises which are useful for retrieval.

The intuition is as follows: Extracting a good expertise from a large knowledge base is harder than from a small one, as the interests of a person interested in many areas will be more difficult to summarize than those of someone who has only few fields of interest. With larger expertises, the retrieval results improve, because if we spend more space (and processing time) for describing someone's interests, we can make better guesses about what his knowledge base contains. As the clustering strategy tries to return the centroids which are as close as possible to all cluster members, we assume that it gives a good approximation of what a knowledge base contains.

4.1 Setup

To evaluate the usefulness of the expertise extraction approach from the previous sections, we consider a P2PKM scenario with a self-organized semantic topology as described in [20, 8, 23]: the expertises of peers are stored in routing tables, where similarity computations between queries and expertises in the routing indices are used to make greedy routing decisions when forwarding queries.

If the routing strategy of this network works as intended, the peers which published an expertise closest to a given query will be queried first. In the following experiment, the quality of the expertises is evaluated in isolation based on that observation: An expertise was extracted for each peer. All of the shared entities of the ontology were used in turn as queries. For each query, the authors were sorted in descending similarity of the closest entity of the expertise to the query. Ties were resolved by ordering in decreasing weight order.

The evaluation is based on the bibliographic use case mentioned in Section 1: there are scientists in the P2P network sharing bibliographic information about their publications. An ontology according to Figure 1 is used. Only the top level concepts (Person, Topic, Paper) and the ACM classification hierarchy are shared among the peers. Each user models a knowledge base on his peer representing his own papers.

We instantiated such a set of knowledge bases using the following data:

- For 39067 papers from DBLP which are present in the ACM Digital Library, the topics were obtained from the ACM website. There are 1474 topics in the ACM Computing Classification System. Details on the construction of the data set and the conversion scripts can be found on `http://www.kde.cs.uni-kassel.de/schmitz/acmdata`.
- To yield non-trivial knowledge bases, only those authors who wrote papers on at least 10 topics were considered. This left 317 authors. A discussion of this pruning step can be found in Section 4.3.

For each of the summarization strategies described below, we show the number of authors which had to be queried in order to yield a given level of recall. This is an indicator for how well the expertises capture the content of the authors' knowledge bases: the better the expertises, the fewer authors one needs to ask in order to reach a certain level of recall.

This is a variation of the usual precision-against-recall evaluation from information retrieval. Instead of precision – how many of the retrieved documents are relevant? – the relative number of the queried authors which are able to provide papers on a given topic is measured.

4.2 Expertise Extraction Strategies

In comparison with the clustering technique from Section 3, the following strategies were evaluated. The expertise size was fixed to be 5 except where noted otherwise.

Counting (#5): The occurrences of topics in each author's knowledge base were counted. The top 5 topics and counts were used as the author's expertise.

Counting Parents (#P5): As above, but each topic did not count for itself, but for its parent topic.

Random (R5): Use 5 random topics and their counts.

Wavefront (WFL7/WFL9): Compute a wavefront of so-called *fuser concepts* [10]. A fuser concept is a concept many descendants of which are instantiated in the knowledge base. The intuition is that if many of the descendants of a concept occur, it will be a good summary of that part of the knowledge base. If only few children occur, a better summarization would be found deeper in the taxonomy.

There are two parameters in this computation: a threshold value between 0 and 1 for the *branch ratio* (the lower the branch ratio, the more salient the topic), and a minimal depth for the fuser concepts. There are some problems in comparing this strategy with the other strategies named here:

- It is not possible to control the number of fuser concepts returned with the parameters the strategy offers.
- Leaves can never be fuser concepts, which is a problem in a relatively flat hierarchy such as ACM CCS, where many papers are classified with leaf concepts.
- All choices of parameters yielded very few fuser concepts.

The expertise consisted of the fuser concepts as returned by the wavefront computation with the inverse of the branch ratio as weights. If the number of fuser concepts was less than 5, the expertise was filled up with the leaf concepts occurring most frequently. We examined thresholds of 0.7 (WFL7) and 0.9 (WFL9) with minimal depth 1.

Clustering (C5/C37): The expertise consisted of centroids and cluster sizes determined by a bisection-k-modes clustering as described in Section 3. C5 used a fixed k of 5, while C37 selected the best $k \in \{3, \ldots, 7\}$ using the silhouette coefficient. 20 retries were used in the bi-section k-means computation.

4.3 Results

In this section, results are presented for the different strategies. The values presented are averaged over all queries (i. e. all ACM topics), and, in the cases with randomized algorithms (C5, C37, R5), over 20 runs.

Note that all strategies except C37 returned expertises of size 5, while in C37, the average expertise size was slightly larger at 5.09. Table 3 shows the distribution of expertise sizes for C37.q

Pruning of the Evaluation Set In order to yield interesting knowledge bases to extract expertises from, we pruned the ACM/DBLP data set as described in Section 4.1. Thus, only the knowledge bases of authors which have written papers on at least 10 topics were considered.

Table 1: Full vs. pruned data: Fraction of authors (%) queried to yield given recall, C5 strategy

Recall	full data	pruned data
10%	0.01	4.09
30%	0.04	4.93
50%	0.07	6.43
70%	0.16	12.53
90%	0.55	18.73
100%	3.45	22.88

Table 1 presents a comparison of the full and the pruned dataset for the C5 strategy. It can be seen that the full data require querying only a fraction of the authors which is one or two orders of magnitude *smaller* than the pruned data. This indicates that the first hypothesis holds; the pruning step yields the “hard” instances of the problem.

Influence of the Expertise Size Intuitively, a larger expertise can contain more information about the knowledge base than a smaller one. In the extreme case, one could use the whole knowledge base as the expertise.

To test the second hypothesis, Figure 2 and Table 2, show the influence of the expertise size on retrieval performance for the C5 clustering strategy.

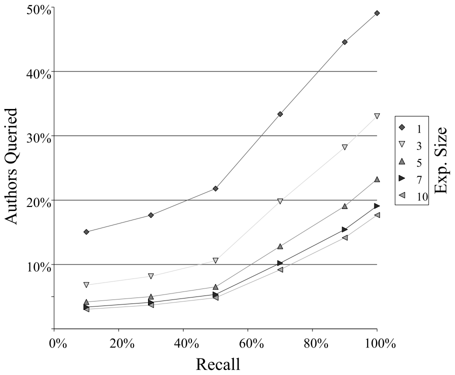

Fig. 2: Influence of Expertise Size (C5 Strategy)

Table 2: Percentage of Authors Queries against Expertise Size (C5 Strategy)

	Expertise Size				
Recall	1	3	5	7	10
10%	15.06	6.80	4.09	3.38	3.03
30%	17.66	8.16	4.93	4.12	3.69
50%	21.79	10.59	6.43	5.35	4.82
70%	33.37	19.79	12.53	10.21	9.18
90%	44.57	28.20	18.73	15.44	14.15
100%	49.07	33.04	22.88	19.10	17.67

Table 3: Distribution of Expertise Sizes for C37

Exp. Size	Percentage of Authors
3	20%
4	15%
5	21%
6	23%
7	21%
Avg.: 5.09	

While the small number of data points for each recall level do not lend themselves to a detailed quantitative analysis, it is clear that the expertise size has the expected influence in the clustering technique: the larger the expertise is, the more detail it can provide about the knowledge base, and the better the retrieval performance is.

Note that the resources a peer would be willing to spend on storing routing tables and making routing decisions are limited, so that a trade-off between resources set aside for routing and the resulting performance must be made, especially as network and knowledge base sizes grow larger.

Influence of the Summarization Strategy Finally, we evaluate the performance of the clustering strategies against the other strategies mentioned above.

Table 4 and Figure 3 show that the k-modes clustering compares favorably against the other strategies: fewer authors need to be asked in order to find a given proportion of the available papers on a certain topic. This is an indication that the clustering technique will yield expertises which can usefully be applied in a P2PKM system with a forwarding query routing strategy based on routing indices. For example, to yield 100% recall, 58% fewer (18.42% vs. 44.15%) peers would have to be queried when using C37 instead of the #5 strategy. With C37 and a routing strategy that contacted best peers first, $100\% - 18.42\% \approx 81\%$ of the peers could be spared from being queried while still getting full recall.

The standard deviations σ of the randomized strategies given in Table 4 show that while the actual results of the C5, C37, and R5 runs may vary, the quality of the results for querying is stable.

Table 4: Percentage of Authors Queried against Recall; σ: Standard Deviation

	Authors Queried						
Recall	WFL7	WFL9	C5 (σ)	C37 (σ)	#5	#P5	R5 (σ)
10%	6.11	6.37	4.09 (.28)	**3.10** (.18)	10.69	9.25	6.96 (.48)
30%	7.16	7.43	4.93 (.28)	**3.80** (.19)	12.15	10.72	8.26 (.52)
50%	9.61	9.86	6.43 (.32)	**5.01** (.21)	15.33	13.67	11.33 (.61)
70%	19.06	19.67	12.53 (.52)	**9.65** (.33)	27.43	23.38	24.04 (.82)
90%	28.97	29.78	18.73 (.64)	**14.78** (.47)	39.45	33.91	35.16 (.93)
100%	34.35	35.37	22.88 (.75)	**18.42** (.48)	44.15	39.27	39.65 (.83)

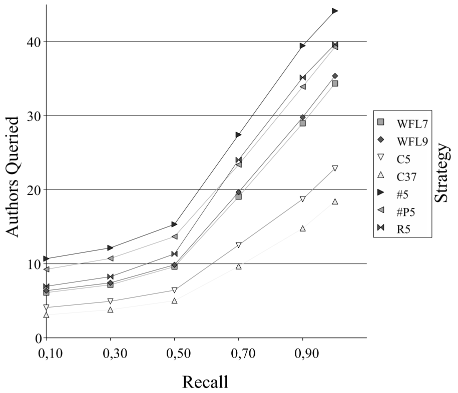

Fig. 3: Percentage of Authors Queried against Recall

To get an impression about why the clustering strategies work better than the others, consider one author whose papers are labelled with the following topics[6]: B.5, B.6, B.6, B.6.1.a, B.6.1.a, B.6.3.b, B.7, B.7.1.c, B.8, B.8, C.0.d, C.3.e, C.5.3.f, D.3.2, G.1, I.5.4.g, J.

The different strategies delivered the results shown in Table 5. It can be seen that the clustering strategies find the best balance between spreading the expertise over all occuring topics, and on the other hand generalizing so that many occuring topics are subsumed under one expertise entry. This happens due to the way the clustering strategy spreads the clusters over the ontology graph, maxminizing the coherence within clusters. Most other strategies, e. g. , did not consider any of the topics outside the B and C parts of ACM CCS.

Table 5: Sample Results for Different Strategies

#P5	#5	R5	WFL7	WFL9	C5	C37
B. (6)	B.6.1.a (2)	B.6.1.a (2)	C. (3)	C. (3)	B. (11)	B.6 (10)
B.6.1 (2)	B.6 (2)	B.6.3.b (1)	B.6.1.a (2)	B.6.1.a (2)	C. (3)	C. (3)
B.6.3 (1)	B.8 (2)	D.3.2 (1)	B. (2)	B. (2)	I.5.4.g (1)	J. (1)
C.0 (1)	B.6.3.b (1)	C.5.3.f (1)	B.6 (1.5)	B.6 (1.5)	D.3.2 (1)	G.1 (1)
B.7.1 (1)	B.5 (1)	B.7 (1)	B.6.3.b (1)	B.6.3.b (1)	G.1 (1)	D.3.2 (1)
						I.5.4.g (1)

5 Summary and Outlook

5.1 Conclusion

In this paper, an algorithm which can be used to extract semantic summaries – called *expertises* – from knowledge bases is proposed. A motivation for the necessity of this kind of summary is given, namely, that such summaries are needed for routing tables in semantic P2P networks.

We demonstrate that the clustering method outperforms other strategies in terms of queries needed to get a given recall on a set of knowledge bases from a bibliographic scenario. We also show qualitatively that larger knowledge bases are harder to summarize, and that larger expertises are an advantage in determining which peers to query.

5.2 Outlook and Work in Progress

Evaluation in Context. This paper provides evidence that the clustering procedure extracts suitable expertises for a P2PKM setting. The next step will be combining the clustering with self-organization techniques for P2PKM networks as described in [20]. Note that usually the value of aggregations or summaries is measured by evaluating it against human judgment. In our case, however, the aggregations will be evaluated with regard to their contribution to improving the performance of the P2P network.

[6] Note that the fourth level topics do not have names of their own originally; we attached artificial IDs to distinguish them

Scalability Issues. Computing the metric as described above is very expensive, as it needs to compute all-pairs-shortest-paths. For large ontologies having tens or hundreds of thousands of nodes, this is prohibitively expensive. In the current evaluation, the shortest paths needed are computed on the fly, but for a real-world P2PKM implementation, some faster solution needs to be found. The obvious idea of pre-computing the metric does not mitigate the problem very much, because maintaining the shortest path lengths requires $O(n^2)$ storage.

On possible direction of investigation is to look at the actual usage of the metric in a P2PKM system. If the community structure of the network leads to a locality in the use of the metric, caching and/or dynamic programming strategies for the metric computation may be feasible.

Test Data and Evaluation Methodology. Other than in Information Retrieval, for example, there are neither widespread testing datasets nor standard evaluation methods available for Semantic Web and especially P2PKM applications. In order to compare and evaluate future research in these areas, standardized data sets and measures need to be established.

Acknowledgement. Part of this research was funded by the EU in the Nepomuk project (FP6-027705).

References

1. K. Aberer, P. Cudré-Mauroux, M. Hauswirth. The Chatty Web: Emergent Semantics Through Gossiping. In *Proc. 12th International World Wide Web Conference*. Budapest, Hungary, May 2003.
2. M. Bonifacio, R. Cuel, G. Mameli, et al. A peer-to-peer architecture for distributed knowledge management. In *Proc. 3rd International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Businesses MALCEB'2002*. Erfurt, Germany, October 2002.
3. A. Crespo, H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Proc. International Conference on Distributed Computing Systems (ICDCS)*. Vienna, Austria, July 2002. ISSN 0734-2071.
4. J. de Bruijn, F. Martin-Recuerda, D. Manov, et al. State-of-the-art survey on ontology merging and aligning (SEKT project deliverable 4.2.1). `http://sw.deri.org/~jos/sekt-d4.2.1-mediation-survey-final.pdf`, 2004.
5. M. Ehrig, P. Haase, F. van Harmelen, et al. The SWAP data and metadata model for semantics-based peer-to-peer systems. In M. Schillo, M. Klusch, J. P. Müller, et al. (eds.), *Proc. MATES-2003. First German Conference on Multiagent Technologies*, vol. 2831 of *LNAI*, pp. 144–155. Springer, Erfurt, Germany, SEP 2003. ISSN 0734-2071.
6. M. Ehrig, S. Handschuh, A. Hotho, et al. KAON - towards a large scale Semantic Web. In K. Bauknecht, A. M. Tjoa, G. Quirchmayr (eds.), *Proc. E-Commerce and Web Technologies, Third International Conference, EC-Web 2002*, no. 2455 in LNCS. Springer, Aix-en-Provence
7. D. Gentner, S. K. Brem. Is snow really like a shovel? Distinguishing similarity from thematic relatedness. In M. Hahn, S. C. Stoness (eds.), *Proc. Twenty-First Annual Meeting of the Cognitive Science Society*. Mahwah, NJ, 1999. ISBN 3-540-40317-5.

8. P. Haase, R. Siebes. Peer selection in peer-to-peer networks with semantic topologies. In *Proc. 13th International World Wide Web Conference*. New York City, NY, USA, May 2004.
9. U. Hahn, U. Reimer. Knowledge-based text summarization: Salience and generalization operators for knowledge base abstraction. In I. Mani, M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. MIT Press, 1999.
10. E. Hovy, C.-Y. Lin. Automated text summarization in SUMMARIST. In I. Mani, M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. MIT Press, 1999.
11. Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Discov.*, 2(3):283–304, 1998. ISSN 1384-5810. doi:http://dx.doi.org/10.1023/A:1009769707641.
12. L. Kaufman, P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990. ISBN 1-58133-109-7.
13. A. Löser, C. Tempich, B. Quilitz, et al. Searching dynamic communities with personal indexes. In Y. Gil, E. Motta, V. R. Benjamins, et al. (eds.), *Proc. 4th International Semantic Web Conference, ISWC 2005*. Galway, Ireland, Nov. 2005.
14. A. Maedche, S. Staab. Measuring similarity between ontologies. In *Proc. Of the European Conference on Knowledge Acquisition and Management - EKAW-2002. Madrid, Spain, October 1-4, 2002*, vol. 2473 of *LNCS/LNAI*. Springer, 2002. ISBN 1-58133-109-7.
15. W. Nejdl, B. Wolf, C. Qu, et al. Edutella: A p2p networking infrastructure based on rdf. In *Proc. 11th International World Wide Web Conference (WWW 2002)*. Honolulu, Hawaii, May 2002.
16. A. Pothen. Graph partitioning algorithms with applications to scientific computing. In D. E. Keyes, A. Sameh, V. Venkatakrishnan (eds.), *Parallel Numerical Algorithms*, pp. 323–368. Kluwer, 1997.
17. R. Rada, H. Mili, E. Bicknell, et al. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, January/February 1989.
18. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95*. Montreal, Canada, August 1995.
19. P. Reynolds, A. Vahdat. Efficient peer-to-peer keyword searching. In M. Endler, D. C. Schmidt (eds.), *Middleware*, vol. 2672 of *Lecture Notes in Computer Science*. Springer, 2003. ISBN 3-540-40317-5.
20. C. Schmitz. Self-organization of a small world by topic. In *Proc. 1st International Workshop on Peer-to-Peer Knowledge Management*. Boston, MA, August 2004.
21. C. Schmitz, S. Staab, C. Tempich. Socialisation in peer-to-peer knowledge management. In *Proc. International Conference on Knowledge Management (I-Know 2004)*. Graz, Austria, June 2004.
22. J. Tane, C. Schmitz, G. Stumme. Semantic resource management for the web: An elearning application. In *Proc. 13th International World Wide Web Conference*. New York, May 2004.
23. C. Tempich, S. Staab, A. Wranik. Remindin': Semantic query routing in peer-to-peer networks based on social metaphors. In W3C (ed.), *Proceedings of the 13th International World Wide Web Conference (WWW 2004)*, pp. 640–649. ACM, New York, USA, MAY 2004.
24. A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.
25. C. A. Welty, D. A. Ferrucci. What's in an instance? Tech. Rep. #94-18, RPI Computer Science Dept., 1994.