Actors–media–qualities: a generic model for information retrieval in virtual communities

Gregor Heinrich

Natural Language Processing Group
Department of Computer Science
University of Leipzig
gregor@arbylon.net

Abstract: The article presents a model of the structural properties of virtual communities and the information they can access. It argues that a large part of the information – and actually knowledge – present in virtual communities can be identified by a graph structure that consists of three node types – actors, media and qualities – as well as the relations that connect them. Based on these relations, information retrieval and other inference mechanisms can be mapped into the model.

1 Introduction

Virtual communities have become a major factor for the design of information systems and Web-based applications, giving rise to community-based information infrastructure. Recently, this development is apparent for instance in the emergence of social computing and the "Web 2.0" [WCZM07], as well as in the paradigm shift in enterprise knowledge management from techno-centric approaches towards tacit knowledge and social capital [HWW03, Got05]. It even extends to the increasing importance of peer-to-peer systems [WB05] and collaboration grids [Sto07] for information management and processing, where peers or grid nodes act in lieu of humans and can be considered members of generalised virtual communities.

A major reason for this shift towards community-based systems is that the information available from in such approaches includes an added value that is impossible to generate or capture using classical, purely content-based approaches, because it directly gains from the intelligence, creativity and social behaviour of people. On the "supply" or authoring side, this added value consists of processes to generate content interesting for the community, which is done by providing infrastructure to easily author or make available contributions, to ask or answer questions (Web 2.0, peer-to-peer), to supply and exchange explicit and tacit knowledge (knowledge sharing infrastructure) or to offer computational services (grid). On the other, the "demand" or retrieval side, the added value consists of processes to provide social decision support, which is done by collaborative filtering, feedback mechanisms and importance measures that often use the relational structure of the community (social recommenders, reputation systems). Further, by offering suitable infrastructure to facilitate and amplify social processes, community-based systems tend to reinforce the identification of their users as members of a community and thus create the motivation to contribute to the community.

Objective. In this article, we investigate the question of how to capture some of the added values that community-based approaches offer and how to combine them with content-based approaches in a reasonably compact model. Such a model is envisioned to describe the structure of community-based information either for comparison and classification of existing systems, or as a structural basis for new developments.

The final goal that this work contributes to is to build infrastructure for *access* to the knowledge that exists within a community, i.e., the demand side of the infrastructure. However, one of the features that makes users of community-based systems unique compared to those of others, is that they are on both sides of the system, supply and demand, i.e., are contributors and retrievers alike. Looking at retrieval, or, more generally, inference approaches cannot therefore completely exclude the content creation processes in communities.

Outline. This paper continues with a more detailed discussion of the requirements needed for the envisioned model in Section 2. As the core contribution of this article, the model itself is proposed in Section 3 and applied to a practical example in Section 4. Finally, after a short statement on related work in Section 5, the present approach and future research is discussed in Section 6.

2 Requirements

This section derives the properties of the envisioned model of virtual communities by specifying requirements. There are three major types of requirements: First we need to identify the scenarios that should be covered (Section 2.1) and need to define how specific the model should be to them (Section 2.2). Finally, the types of community knowledge of the scenarios that should be addressed in the model need to be considered (Section 2.3).

2.1 Requirement 1: Usage scenarios covered

The main purpose of the envisioned model is to *represent a virtual community to support information access scenarios and associated inference tasks*. Such tasks are for instance to find community members and documents according to certain criteria. The scenarios for information access include:

- Expert finders: Systems that allow to find people who have expert knowledge in a given topic, based on profile information, document content and authoring information (e.g., MITRE [MDH00], AnswerGarden [AM96], XperT [Hei04]);

- Digital libraries: Systems that allow to access documents where the community consists of the authors who mutually cite their articles or monographies (CiteSeer [GBL98][1], the ACM Digital Library [Whi01]);
- Collaborative authoring: Systems that allow community members to contribute to a collaborative information repository, either *ad hoc* asynchronous communication (mailing lists, forums) or as "Web 2.0" tools like blogs, wikis [WCZM07]) and other approaches that make accessible and allow users to contribute content (Twitter, flickr and YouTube) or meta-content (structured data as in IMDB, or tags as in CiteULike and del.icio.us);
- Social network platforms: Systems that offer self-authored personal profiles and connections as dynamic contact and friends lists (Xing, myspace);
- Peer-to-peer systems: Systems that distribute content over a community of peer modules (that can themselves represent a community of people) and can be considered "generalised communities" (SemPIR [WB05]).

2.2 Requirement 2: Scope and specificity

The model is primarily used in early stages of system design where the working concepts are decided and basic algorithmic considerations are undertaken (cf. [BFHV03]). Therefore, the model should be generic enough to be *independent of scenario specifics* like particular types of persons or documents. This also makes it suitable for representation and comparison of a wide variety of existing and new systems and scenarios. In fact, the result may be a form of data model or ontology whose structure can be specialised for particular scenarios in question, similar to meta-modelling methodologies (cf. [Bez06]).

2.3 Requirement 3: Information and knowledge types addressed

Many systems for community-based information infrastructure are not only used to retrieve information but actually knowledge. For these cases, it is inevitable to not only *represent documented information as explicit knowledge*, but to also *integrate tacit knowledge* [NT95, Boi99] and possibly *social capital* [Les00] in the model. This way, a great part of the added value can be captured that is ascribed to community-based approaches compared to purely content-based ones, as suggested by the literature (see, e.g., [Wen98, HWW03] covering Communities of Practice).

The access to knowledge needs a few more remarks. By definition, tacit knowledge – or "knowing" as a process rather than a state [Pol74] – is restricted to individuals or groups of individuals and it is in most cases difficult to write down (to "externalise" [NT95]) because it depends on intangible factors like experience, procedural knowledge, special

[1] Registration of articles with CiteSeer is actually a community-based process, as well.

talents, cultural background and norms that can only be made available to others by direct interaction. Social capital as a form of collective tacit knowledge [DGKT03] supports this interaction by holding together communities and being the basis for offers of help needed to achieve shared goals or to solve problems [Put00].

To give an example, in a newsgroup the experience of an expert answering a complex question cannot be written down in its entirety; it is tacit. Further, the fact that such exchange works at all is often a result of the social capital established within the community, which is tacit as well. The predominant way to approach the problem of making available such tacit "assets" is to identify the expert, e.g., from a profile, from previous answers or articles, by explicit recommendation, possibly confirmed by the location within a social network that reflects the communication within the community.

Therefore, "cues" to tacit knowledge (and social capital) in the community are important auxiliaries. Locating both tacit and explicit knowledge extends the notion of information retrieval, and in the following we use the term information retrieval to refer to this more general form, avoiding new definitions like "knowledge retrieval" because at their core, the basic approaches are those of information retrieval [BYRN99], and tacit assets should be represented in the model as explicit cues that point to them. The types of such cues are manifold, and we take an "inductive" approach and summarise sources that are commonly used and need to be represented in the envisioned model:

- Authoring and reference information: Tacit knowledge leaves traces in documents that are created in the community, either by experts themselves ("authoring") or via reference in documents by other authors, such as in reports and in scientific citations ("reference"). This is not retricted to text content, like scientific authoring, but also in non-textual media, for instance in the way a movie is edited by an expert editor. Authoring and reference information is captured "en passant" from the existing processes in the community.

- Profile information: The existence of expertise can be catalogued using questionnaires, interviews, structured CVs and other means that are "actively" or explicitly applied to capture the existence of tacit knowledge, as in many knowledge and skills managment approaches. However, many tacit skills are unknown even to the expert, and in these cases, only by interaction and problem solving can tacit knowledge be located and may be "implicitly" captured by tracking collaboration. Both active and implicit ways to capture specific traits and properties of users contribute to profiles.

- Social network information[2]: Important pieces of tacit community knowledge are identified from the structure of the community, i.e. the position of individuals and groups in the social network. Depending on the type of relations available as a representation of the real social network, this may allow identification of experts by their embedding into clusters of other experts, as well as possibly capture cues of social capital, such as trust, recommendation and reputation, which are important prerequisites to sharing of tacit knowledge through collaboration.

[2] In "genereralised communities" like peer-to-peer networks, social network information does not represent social capital proper but similarly, relational properties of the network are used as cues to identify items of interest.

3 The actors–media–qualities model

The requirements collected in Section 2 yield a set of qualitative input factors to develop the model. With the focus on information access, Requirement 1 implies the need for a semantic representation of the items in the model that can be used to evaluate the relevance to a query, i.e., some sort of profile or set of "qualities" that can be associated with the items. The generic scope (avoidance of scenario specifics) from Requirement 2 implies what in ontology design is called "minimal ontological commitment", i.e., the restriction of the model to a minimum of elements [Gru94]. Requirement 3 implies on one hand the representation of explicit knowledge items, which can be modelled as a documents or, more general, "media", on the other cues of tacit knowledge, i.e., the types of sources in the list in Section 2 need to be included. Authoring and reference demand for a connection of documents or media with community members, and profiles can be considered special cases of documents. Finally, social network information indicates the appropriateness of a graph-based representation of the community.

Fortunately, such a graph representation is flexible enough to be the structural basis for the entire model. Authoring information etc. can be expressed by including into the network media items and connections with the authors. Moreover, semantic or other qualities associated with the items can be directly included as nodes in the graph representation.

In the next subsections, we define the model in terms of a graph structure. We first introduce node types in Section 3.1, its edge types in Section 3.2 and finally the complete model structure in Sec.3.3.

3.1 Defining entities

We define AMQ entity types similar to classes in ontology or software analysis and denote them by calligraphic letters like $\mathcal{A}$. Our model supports subsumption/inheritance (*is_a*) relationships, i.e., entity types may have a hierarchy of subtypes that are denoted in italic type $A \in \mathcal{A}$ etc. Further, it supports aggregation (*has_a*) relationships. Instances, i.e., objects that represent the actual data, will then be denoted in lower case $a \in A \in \mathcal{A}$. Considering type hierarchy, for simplicity we will use the shorthand $a \in \mathcal{A}$ to denote that a is an instance of some type $A \in \mathcal{A}$. Three root entity types are proposed to model a community according to the above requirements: actors, media and qualities.

Actors, $a \in \mathcal{A}$**,** are entities that represent everyone/everything acting in an autonomous way, which implies actions like to *write*, *collaborate*, *query*, *study* or to *assess*, in addition to explicit (verbal) knowledge (to *know*). These actions will later be defined as relations. Actors bear tacit knowledge, and naturally represent people and groups of people that engage in knowledge sharing and interaction. As a special case, intelligent agents can be considered actors although they are often represented via explicit rule sets. Subsumption of actors is defined (an author *is a*n actor), as is aggregation (a group *has a* number of authors).

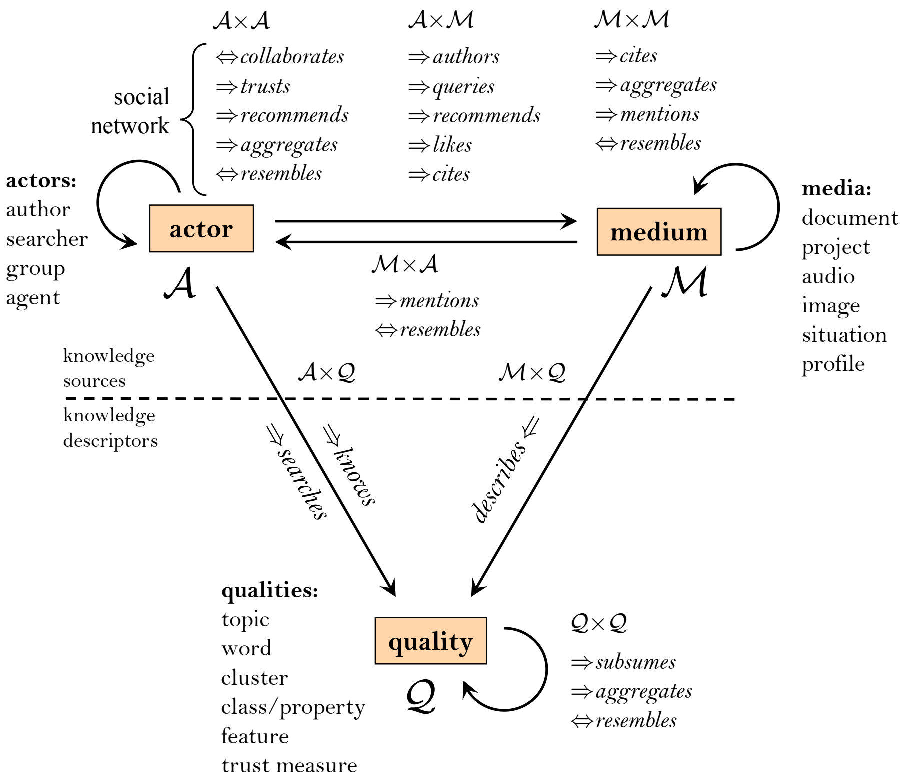

Figure 1: Structure of the AMQ model, with example entity and relation (italics) types. Directed relations between classes denoted with ⇒, undirected ones with ⇔.

Media, $m \in \mathcal{M}$**,** are entities that contain information and "react" to actors. Media bear explicit, i.e., verbal or numerical knowledge, and can be thought of as a generalisation of documents to all formats that can contain explicit knowledge or serve as cues to knowledge, including for example project and course documentation, audio tracks and images, video clips and other artefacts. In addition, user queries fit into this scheme as "reciprocal" media (requesting rather than providing knowledge). Further, user profiles behave like media. Like actors, media allow subsumption (a document *is a* medium) and aggregation (a book *has a* number of chapters).

Qualities, $q \in \mathcal{Q}$**,** are entities that provide a set of attributes to describe actors, media and other qualities in a way that inference can be performed on them. This inference includes comparison using a distance or similarity metric and consequently retrieval. Qualities as the units of knowledge representation are one key to mapping existing inference and information retrieval methods into the model and can be used develop new models. Considering existing metadata frameworks, full-text approaches like [AM96] use inverted indices as representations of qualities. In Semantic Web-based information retrieval, reasoning requires qualities to be defined with formal semantics, i.e., $\mathcal{Q}$ is the representation of an ontology that can be queried or reasoned upon. And when using latent-semantic models like latent Dirichlet allocation [BNJ02], automatically inferred latent variables (commonly referred to as latent topics) are the qualities to describe the entities in the model. Finally, when dealing with multimedia data, content-based features can be used as qualities.

For a schematic view of these entities, see Fig. 1, where they are displayed as rectangles, along with examples of typical classes they subsume. With regard to its three root entity types, we call the model the *actors–media–qualities model* or *AMQ model*.

3.2 Defining relations

With the entities allowing to represent real-world items in the model as instances of one of the actors, media and qualities types (or subtypes), relations between them provide the actual information used for inference and retrieval. Relations in the AMQ model are quantified by some weighting function $w(x, y, R) : \mathcal{X} \times \mathcal{Y} \rightarrow I_R \subset \mathbb{R}$ with I_R the set of allowed weighting values for relation of type $R \in \mathcal{R}$ and $\mathcal{R}$ being the set of all relation types. Depending on the relation type, this interval can be discrete and binary, $I_R = \{0, 1\}$, probablistic, $I_R = [0, 1]$, some distance or similarity measure, $I_R = [0, \infty)$, or any other set depending on the semantics of the relation. Typically, relations restrict the node types $\mathcal{X}$ and $\mathcal{Y}$ they map to each other (domain and range), and may be directed or undirected. For brevity, we represent relations by their weighting function, using the entities a, m and q as defined above. Typical relation types as given in Fig. 1 will be discussed in the next paragraphs.

Media–quality relations, $w(m, q, R) : \mathcal{M} \times \mathcal{Q} \rightarrow I_R$, describe the semantics of media. Explicit knowledge cues in media items use the media–quality (*mq*-) relation *describes*, denoting explicit knowledge in a particular subject. Here, $I_R = [0, 1]$ is a function that maps to a relevance value for the subject, and to describe one this subject, weights are established from the medium to all possible qualities q_i that this subject is composed of. Combining the weightings of the *mq*-relation over several qualities $\vec{q} = \{q_i\}_i, i \in [1, K]$ can be expressed by introducing a shorthand for a vector weighting function $w(m, \vec{q}, R) : \mathcal{M} \times \mathcal{Q} \rightarrow (I_R)^K$ over all qualities q_i. This can for instance represent a vector of topic probabilities or a set of binary association functions with the range of ontology classes. When trying to retrieve relevant documents, the subject must be expressed as a set of qualities, which themselves are extracted from a query text or other object (as a “reciprocal” medium).

Actor–quality relations, $w(a, q, R) : \mathcal{A} \times \mathcal{Q} \rightarrow I_R$, describe the semantics of knowledge associated with an actor. The central relation with respect to knowledge cues is the actor–quality (*aq*-) relation *knows*, which, however, is not directly observable. In most approaches in the literature, the *knows* relation is inferred, for instance from *author*ship: For example, in the MITRE [MDH00], AnswerGarden [AM96] and XperT [Hei04] systems, knowledge cues from documents are used via the *describes* relation, and experts are inferred via the actor–media (*am*-) relation *authors*. The Author-Topic Model [RZGSS04], however, directly extracts latent topics for actors, implementing the *knows* relation directly. Opposite to this "supply" dimension of knowledge, the "demand" of specific knowledge can be evaluated for actors, which is reflected by the actor–quality relation *searches* that can be inferred via the *describes* and *queries* relations (for all explanations, see Fig. 1). Comparing the qualities of both supply and demand dimensions enables the functionality of matchmaking systems like that in [RSW05].

Actor–media relations, $w(a, m, R) : \mathcal{A} \times \mathcal{M} \rightarrow I_R$ or $w(m, a, R) : \mathcal{M} \times \mathcal{A} \rightarrow I_R$, describe the association of an actor with a medium or vice versa. Actor–media (*am*-) relations usually derive from authoring and reference information (*authors*, *cites*, *recommends* etc.). Further, query actions by actors are special types of AM relations (*queries*). Typically, such information is often explicit and can be extracted a priori as a basis for inference. Inferred *am*-relations are used in collaborative approaches to express recommendation (*recommends*) and preference (*likes*).

Media–media relations, $w(m, m', R) : \mathcal{M} \times \mathcal{M} \rightarrow I_R$, describe mutual relationships between media. Media–media (*mm*-) relations play an important role in citation networks and digital libraries, both as references and aggregation. Like *am*-relations, they are often explicit and can be used as basis for inference. An important inferred relation for retrieval is similarity (*resembles*).

Actor–actor relations, $w(a, a', R) : \mathcal{A} \times \mathcal{A} \rightarrow I_R$, describe social structure of the community and other relations between actors. Actor–actor (*aa*-) relations can represent information on social capital in the community. ReferralWeb [KSS97], for example, uses relational cues such as friends, colleagues, and co-workers, Opal [HKJ$^+$05] in addition ratings between collaboration candidates [DM04]. Depending on the application case, different types of inference are possible. An example is to find an actor who is an expert in a topic and trusted by reputable actors. The inference based on explicit cues is the same as described for *aq*-relations, but the set of relevant actors now is filtered via appropriate network criteria, such as shortest path or reputation measures that aggregate weighted ratings (see [KSS97, PSD03] and references therein). An alternative way is to perform inference in an integrated manner is to use statistical relational learning techniques that integrate semantic and relational steps of inference (see, e.g., [Nev06]).

Quality–quality relations, $w(q, q', R) : \mathcal{Q} \times \mathcal{Q} \rightarrow I_R$, map knowledge description frameworks into the AMQ model. For instance, for ontologies quality–quality (*qq*-) relations may include RDFS or OWL relations (e.g., *rdfs:subClass*, properties or aggregations). The AMQ model does not make any commitment on the formalism for *qq*-relations, allowing to include axiomatic descriptions of qualities, for instance to use the results of Semantic-Web inference, or hierarchical relations between latent topics. Further, *qq*-relations are the place in the model where distance measures fit in to compare actors and media as the knowledge sources in the model, with ontology approaches on one hand (mostly leading to binary results) and real-valued retrieval functions modelling relevance on the other. For instance, two actors in an expert finder system may be similar in terms of their knowledge if the qualities they are described with are similar.

3.3 AMQ graphs and inference

In order to complete the AMQ model, all data are joined in a graph structure, which is the basis for inference algorithms. More specifically, taking ideas from ontology modelling, e.g., [MvH04], the schema and instance structures are distinguished.

Schema graph. The structure that combines the entities and relations discussed in the last sections is defined as an AMQ schema graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with the vertex set consisting of the three entity types (possibly their subtypes), $\mathcal{V} = \mathcal{A} \cup \mathcal{M} \cup \mathcal{Q}$, and the edge set mapping to the various relation types between them, $\mathcal{E} : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{R}$, with $\mathcal{R} \ni R$ denoting the set of relation types. Fig. 1 can be understood as a simplified example of a schema graph where the different node and edge types are collapsed into the clique of root entity types, which will be extended by a more expressive graphical notation in Section 4.

Instance graph. While the schema graph reflects the structure of the data about the community, an instance graph $G(V, E)$ fills this AMQ schema with data, leading to a kind of generalised co-citation graph or social network [WF94]. The instance graph contains typed objects as vertices, $v \in V$, where each vertex v has a type that is a member of $\mathcal{V}$ in the schema, as well as edges between the objects, $e \in E$, where each e maps to a relation type $R \in \mathcal{R}$ and a weight, i.e., $E : V \times V \rightarrow \mathcal{R} \times \mathbb{R}$ with $\mathbb{R} \supset I_R$ subsuming the range of all weighting functions $w(x, y, R)$. Note that this definition can be easily extended to hypergraphs by allowing edge sets with different vertex counts per edge.

In order to represent a virtual community for a retrieval scenario, typically only a small set of entity and relation types need to be included in the schema, depending on the available information on the community and the retrieval mechanisms required to fulfill a particular set of retrieval tasks.

Inference in AMQ models is the process of identifying or creating entities or relations in the AMQ instance graph by analysis of its semantic or structural properties. Semantics here refers to the qualities associated with entities (e.g., topics associated with a document), and structure to the general topology of the AMQ graph (e.g., co-citation, social network) spanned by the different data available.

More formally, inference algorithms can be defined as transformations from a given instance graph structure $G(V, E)$ to another structure that adds the inferred items to G: $G' = G(V, E) \cup G(\hat{V}, \hat{E})$.

In this way, standard methods of information retrieval and inference may be expressed in the AMQ model, providing a method to classify or unify existing algorithms, possibly creating a library of standard algorithms for re-use. Further, the method allows to define novel inference schemes that may use combinations of existing algorithms or lead to completely new approaches. As the AMQ model itself makes no commitment on the type of inference used, the range of possibilities is wide. In the next section, this will be explained with an example.

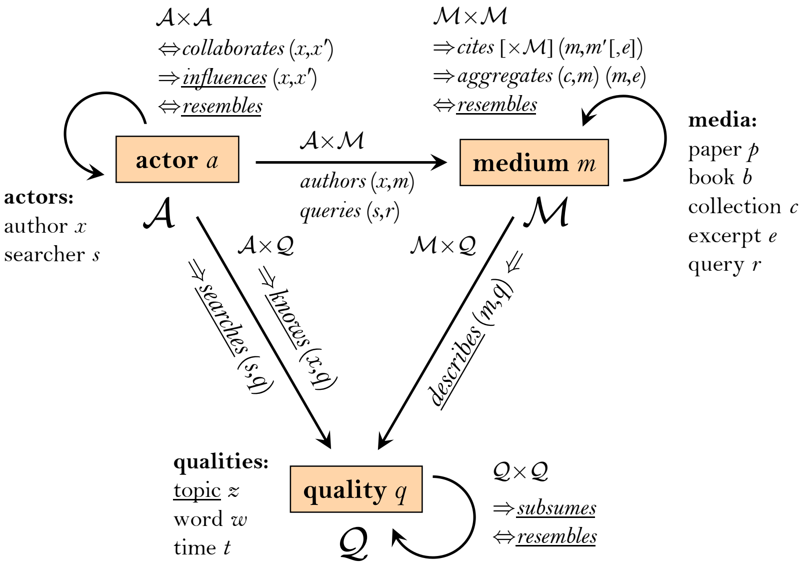

Figure 2: AMQ schema considered for CiteSeer and other digital library corpora.

4 Example: CiteSeer as an AMQ model

An illustrative application for an AMQ model is the CiteSeer digital library [GBL98], which offers research publications online. The authors of these publications can be considered to form a scientific virtual community whose members venture to create new knowledge by using and extending existing sources.

Being only one example among a set of scientific digital libraries (cf. the CORA dataset[3] or the digital libraries in Section 2), beside the semantic content of titles, abstracts and full-texts, CiteSeer gives access to authorship, co-citation, publication time (and partly venue) information and thus can be used to track the knowledge creation process in the community or to identify relevant and influential papers. However, the CiteSeer portal has only limited retrieval features, allowing document and author name search while restricting structural analysis to importance measures in the co-citation graph. On the other hand, a snapshot of the CiteSeer portal data exists that may be used to extend this: the CiteSeer corpus[4] with approx. 564k document abstracts and titles, 229k authors and 1272k intra-corpus citations.

Before we analyse inference and retrieval tasks possible on these data in Section 4.2, we characterise the structure of the CiteSeer data as an AMQ model in Section 4.1.

[3] http://www.cs.umass.edu/~mccallum/code-data.html

[4] http://citeseer.ist.psu.edu/oai.html.

4.1 Schema

The AMQ schema structure is shown in Fig. 2. Compared to Fig. 1, only the entities and relations relevant to CiteSeer are displayed, but on the other hand, Fig. 2 adds some additional information on the schema:

- Entities are added symbols, like x for an author as a special type of actor, a.
- These symbols are used to specify the domain and range of the different relations, like $\mathit{authors}(x, m)$, which means that this relation applies to authors (as opposed to searchers, s) but to all possible media types, m. This notation allows to retain the collapsed simple graphic representation of the AMQ schema. If no constraints are specified, a relation applies to the root types.
- Underlined types are considered inferred, whereas the others can directly be extracted from the available data.

Regarding actors, there are authors and searchers. The data in the CiteSeer corpus in fact do not include any "demand" data; the searcher entity $s \in \mathcal{A}$ is rather included to show the querying process of someone retrieving data from the corpus than to represent data contained in it. Looking at media entities seems self-explanatory, an excerpt $e \in \mathcal{M}$ being part of a medium ($\mathit{aggregates}(m, e)$) that can serve as context for a reference to another medium. Then optionally the binary $\mathit{cites}(m, m')$ relation between two documents is extended to a ternary $\mathit{cites}(m, m', e)$ relation with additional excerpt e as part of m (leading to a hypergraph structure). Finally, there are three different types of quality: topic, word and time, and while the latter two can be directly read from the corpus by indexing or from metadata, topics are themselves inferred entities, e.g., extracted by latent Dirichlet allocation or the author–topic model (see qualities definition in Section 3.1). The qq-relation *subsumes* can be applied to a topic for a hierarchical approaches, to a word when using some semantic hierarchy and to nest periods of time.

4.2 Inference tasks

On the CiteSeer data, various existing and novel inference and retrieval methods can be applied, and here we view them from the perspective of how they are expressed in terms of the AMQ model rather than the actual algorithms.

Media retrieval is to find documents etc. for a given query. This requires initial indexing, which creates *describes* relations between media and qualities (words for inverted index or vector-space models [BYRN99], inferred topics for latent semantics, etc.). During search, the same is done for a query, completing the graph with the respective weighting. The actual ranking of relevant documents is then based on an appropriate distance measure between the weightings of the *describes* relation, using vector weighting function $w(m, \vec{q}, \text{“}\mathit{describes}\text{”}) : \mathcal{M} \times \mathcal{Q} \to \mathbb{R}^K$ with K qualities for a given medium m. In the AMQ schema, such distances may be mapped to a *resembles* relation as a basis for ranking:

$$w(m, m', \text{“}\textit{resembles}\text{”}) = f(\text{distance}\{w(m, \vec{q}, \text{“}\textit{describes}\text{”}), w(m', \vec{q}, \text{“}\textit{describes}\text{”})\}) \ .$$

In the example, both words and topics as qualities allow to combine literal and latent-semantic search in an appropriate retrieval function or distance measure. For latent semantics, the weighting function $w(m, \vec{q}, \text{“}\textit{describes}\text{”})$ represents the probability distribution $p(z|m)$ over K topics and implies the existence of topic distributions $p(w|z)$ that map words to topics (cf. [BNJ02]).

Expert finding. Extending document retrieval to scenarios like expert finding (see Section 2) is simple: Ranking for retrieval is then done via distances between *knows aq*-relations or *describes mq*-relations of documents authored by a particular person, identified via the *authors am*-relation that infers a *knows* relation:

$$w(a, \vec{q}, \text{“}\textit{knows}\text{”}) = f(\{w(m, \vec{q}, \text{“}\textit{describes}\text{”})\}_m) \ \forall \ \{m : w(a, m, \text{“}\textit{authors}\text{”}) > 0\} \ .$$

For the actual implementations of the associated algorithms, numerous possibilities exist in the literature, e.g., the mentioned [RZGSS04] or [Hei04] that make use of latent topic distributions $p(z|x)$ for authors.

Advanced tasks. Beyond this, various other inference and retrieval tasks can be performed with the CiteSeer schema, for instance:

- Semantic matching: For a given document, the distance to other documents is inferred based on the *describes* relation, inferring the $\textit{resembles}(m, m')$ relation.
- Co-citation matching: The similarity between the subgraph structures spanned by *cites* relations around two documents is inferred, e.g., based on intersection.
- Document citation influence: The influence of a document along $\textit{cites}(m, m')$ relations is inferred, e.g., using graph importance measures like PageRank.
- Author influence: Document citation influence may be mapped to their authors using $\textit{authors}(a, m)$.
- Actor matching: Combinations of searcher and author are ranked by their *knows* or *searches* relations, inferring *resembles* relations.
- Sub-community detection: Similar interests *searches* and/or knowledge *knows* can be clustered into communities with an entity group $g \in \mathcal{A}$ and $\textit{aggregates}(g, a)$.
- Semantic citation influence: Combining the influence of citations with their semantics ($\textit{describes}(m, q)$), possibly exploiting citation context via $\textit{cites}(m, m', e)$.
- Dynamic models: Combining the above models with temporal information and its temporal derivatives, e.g., to analyse the evolution of *describes* or *searches* relations.

Moreover, with relevance or preference information included in the data, this list could be extended by collaborative approaches like recommender systems where actors rate media via $\textit{recommends}(s, m)$ and inference yields a $\textit{likes}(s, m)$ relation, possibly creating profile clusters similar to the groups g above.

5 Related work

Regarding previous approaches to define some generic structural basis of information access that uses the typical data available in virtual communities, existing work turned out to be surprisingly scarce.

Nevertheless, several research strands are relevant, mostly considering the way data is represented. From this perspective, the AMQ model can be viewed as an extension of social networks [WF94] by documents and items of knowledge representation. On the other hand, there are close relationships with ontology modelling [Gru94], particularly the Web Ontology Language OWL [MvH04]: Regarding entities and their types, OWL defines individuals that belong to classes that themselves support subsumption and aggregation as well as other relations called properties to link individuals to each other (object properties) or to data values (datatype properties). The AMQ model takes up the individual and class concepts but is limited to the object properties as the basis for its relations. Because OWL and other ontology languages are focussed on logical reasoning, the concept of weighted relations cannot be modelled in a simple and expressive way but rather requires workarounds like reification. Because the possibility of weighted relations is at the core of the AMQ model and it does not restrict inference methods to logic reasoning, the AMQ model has been defined as a more generic graph structure.

Moreover, the AMQ model can be considered an application of entity-relationship modelling [Che76] to community-based information retrieval tasks, providing a "template" to designing domain-specific database schemes. In a similar direction, the approach has some relations to meta-modelling [Bez06] as it can be used to derive models from a template model structure.

All of these viewpoints focus on the data structure that the AMQ model defines. Considering its objective to classify inference tasks and retrieval algorithms for community knowledge reveals no specific work beyond the general treatments of information retrieval methods like [BYRN99].

6 Conclusions

In this article, the "AMQ model" was developed, a representation of virtual communities that can be used as the basis for information systems to support retrieval and inference on their data. The model can be considered an attempt to characterise the domain of virtual communities from a data structure viewpoint considering the most important factors of explicit and tacit knowledge. Yet the model stays conceptually simple and does not claim to cover all imaginable scenarios. Rather, it attempts to pragmatically characterise a domain of applications of community knowledge access, namely such that infer similarity, relevance, classifications and other information from relations between people, documents and semantics.

By formalising information structure of community-based retrieval and inference tasks, the proposed model opens a new perspective on how to develop such approaches, and research can depart from this into various directions. First, as this paper only discussed the structure of the data and associated tasks, one of the foremost future research topics is to fill the tasks with concrete algorithms. Here it is of special interest to explore combinations of existing approaches to obtain better retrieval tools, e.g., merging semantic with collaborative approaches. Second, an empirical study of the properties of the AMQ graphs of real-world scenarios may reveal interesting features that may be exploited for novel inference methods, e.g., based on suspected small-world properties of different relations and their combinations.

References

[AM96] M.S. Ackerman and D.W. McDonald. AnswerGarden 2: Merging organizational memory with collaborative help. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, pages 97–105, 1996.

[Bez06] J. Bezivin. On the Unification Power of Models. *Software and System Modeling (SoSym)*, 4(2):171–188, 2006.

[BFHV03] Peter Barna, Flavius Frasincar, Geert-Jan Houben, and Richard Vdovjak. Methodologies for Web Information System Design. In *Proc. ITCC*, 2003.

[BNJ02] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[Boi99] Max H. Boisot. *Knowledge assets – securing competitive advantage in the information economy*. Oxford University Press, 1999.

[BYRN99] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press & Addison-Wesley, 1999.

[Che76] Peter P. Chen. The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

[DGKT03] E. Davenport, M. Graham, J. Kennedy, and K. Taylor. Managing Social Capital as Knowledge Management – Some Specification and Representation Issues. In *Proc. American Society for Information Science and Technology (ASIS&T)*, pages 101–108, 2003.

[DM04] Elisabeth Davenport and Leo McLaughlin. *Trust in knowledge management and systems in organizations*, chapter Interpersonal Trust in Online Partnerships: The Challenge of Representation, pages 107–123. Idea Group, 2004. ISBN:1-59140-220-4.

[GBL98] C. Lee Giles, Kurt Bollacker, and Steve Lawrence. CiteSeer: An Automatic Citation Indexing System. *Proc. 3rd ACM Conf. on Digital Libraries*, pages 89–98, June 23–26 1998.

[Got05] Petter Gottschalk. *Strategic knowledge management technology*. Idea Group, 2005.

[Gru94] Thomas Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies*, 43(5/6):907–928, 1994.

[Hei04] Gregor Heinrich. Teamarbeit nach Mass – Expertisemanagement in Organisationsnetzwerken (in German). In Anette Weisbecker, Thomas Renner, and Stefan Noll, editors, *Electronic Business – Innovationen, Anwendungen und Technologien*, pages 52–59. Fraunhofer IRB-Verlag, Stuttgart, Sep 2004. ISBN 3-8167-6621-8.

[HKJ+05] Gregor Heinrich, T. Keim, C. Jung, U. Krafzig, and S. Noll. Smart collaboration networks – a toolkit and a vision for creating and predicting partnership. In *Proc. Int. Conf. eChallenges*, 2005.

[HWW03] Marleen Huysman, Etienne Wenger, and Volker Wulf, editors. *Communities and Technologies*. Dordrecht: Kluwer, 2003.

[KSS97] H. Kautz, B. Selman, and M. Shah. ReferralWeb: Combining Social Networks and Collaborative Filtering. *Communications of the ACM*, 40(3), March 1997.

[Les00] E. Lesser, editor. *Knowledge and social capital: foundations and applications*. Oxford: Butterworth-Heinemann, 2000.

[MDH00] M. Maybury, R.D. D'Amore, and House. *Beyond Knowledge Management: Sharing Expertise*, chapter Automated Discovery and Mapping of Expertise. Cambridge: MIT Press, 2000.

[MvH04] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. W3c recommendation, W3C, Feb. 2004.

[Nev06] Jennifer Neville. *Statistical Models and Analysis Techniques for Learning in Relational Data*. PhD thesis, Stanford University, 2006.

[NT95] Ikujiro Nonaka and Hirotaka Takeuchi. *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford Univ. Press New York/Oxford, 1995.

[Pol74] Michael Polanyi. *Personal knowledge: Towards a Post-Critical Philosophy*. University of Chicago & Press, Chicago, 1974. Originally published in 1958.

[PSD03] J.M. Pujol, R. Sangüesa, and J. Delgado. *Web Intelligence*, chapter A Ranking Algorithm Based on Graph Topology to Generate Reputation or Relevance, pages 382–395. Springer, 2003.

[Put00] R.D. Putnam. *Bowling Alone: The Collapse and Revival of American Community*. New York: Simon & Schuster, 2000.

[RSW05] Tim Reichling, Kai Schubert, and Volker Wulf. Matching Human Actors based on their Texts: Design and Evaluation of an Instance of the ExpertFinding Framework. In *Proceedings of GROUP 2005*. New York: ACM-Press, 2005.

[RZGSS04] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The Author-Topic Model for authors and documents. In *20th Conference on Uncertainty in Artificial Intelligence*, 2004.

[Sto07] Heinz Stockinger. Defining the grid: a snapshot on the current view. *J Supercomput*, (to be published), 2007.

[WB05] H.F. Witschel and T. Böhme. Evaluating profiling and query expansion methods for p2p information retrieval. In *Proc. of the 2005 ACM Workshop on Information Retrieval in Peer-to-Peer Networks (P2PIR)*, 2005.

[WCZM07] Fei-Yue Wang, Kathleen M. Carley, Daniel Zeng, and Wenji Mao. Social Computing: From Social Informatics to Social Intelligence. *Intelligent Systems, IEEE*, 22:79–83, 2007.

[Wen98] E.C. Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 1998.

[WF94] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[Whi01] John White. ACM Opens portal. *Commun. ACM*, 44(7):14–ff, 2001.