Network Analytics ER Model – Towards a Conceptual View of Network Analytics

Qing Wang

Research School of Computer Science, The Australian National University, Australia,
qing.wang@anu.edu.au

Abstract. This paper proposes a conceptual modelling paradigm for network analysis applications, called the Network Analytics ER model (NAER). Not only data requirements but also query requirements are captured by the conceptual description of network analysis applications. This unified analytical framework allows us to flexibly build a number of topology schemas on the basis of the underlying core schema, together with a collection of query topics that describe topological results of interest. In doing so, we can alleviate many issues in network analysis, such as performance, semantic integrity and dynamics of analysis.

1 Introduction

Network analysis has proliferated rapidly in recent years, and it has useful applications across a wide range of fields, such as social science, computer science, biology and archaeology [2, 3, 10, 13, 15, 16]. One key aspect of network analysis is to understand how entities and their interaction via various (explicit or implicit) relationships take place within a network that is often represented as a graph with possibly millions or even billions of vertices. In practice, network data are often managed in a database system, e.g., Facebook uses MySQL to store data like posts, comments, likes, and pages. Network analysis queries are performed by extracting data from the underlying database, then analyzing them using some software tools that incorporate data mining and machine learning techniques [11]. Since different fragments of data may be of interest for different analysis purposes, network analysis queries are usually performed in ad hoc and isolated environments. Therefore, there is a divorce of data models and query languages between managing network data and analyzing network data in many situations, and several questions may arise.

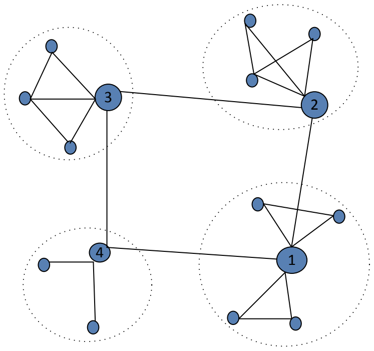
- *Semantic integrity*
  With more and more network analysis queries being performed, it becomes increasingly important to semantically align and mine their relationships. But how can we ensure that they are semantically relevant and consistent?
- *Analysis efficiency*
  Network analysis queries are often computationally expensive. Regardless of implementation details that different network analysis queries may have, the

need to capture semantics remains. Can the efficiency of network analysis queries be improved by leveraging their semantics at the conceptual level?

– *Network dynamics*
Network analysis applications are dynamic and evolving over time. Can network analysis be dynamically performed at different scales and over different time periods so as to predict trends and patterns?

The root of these questions stems from two different perspectives on networks - one is from the data management perspective (i.e., how to control data), and the other is from the data analysis perspective (i.e., how to use data). These two perspectives are closely related but have different concerns. We believe that conceptual modelling can play an important role in bridging these two perspectives, and contribute to answering the above questions. This paper aims to explore this, and in a broader sense, it also attempts to envision the role of conceptual modelling in the era of Big-data analytics since network analysis is at the core of Big-data analytics.

Example 1. Fig. 1.(a) depicts a simple network in which each vertex represents an author, and each edge represents that two authors have coauthored one or more articles. Suppose that we have two network analysis queries: (1) Q_c - find the collaborative communities of authors according to how closely they collaborate with each other to write articles together, and (2) Q_a - find the top-k influential researchers. With the results of Q_c and Q_a available (i.e., as shown in Fig. 1.(a) and (b)), we may further ask: (3) Q_{ca} - what are the collaborative communities of these top-k influential researchers? (4) Q_{ac} - are these top-k influential researchers the central ties in their collaborative communities? To answer Q_{ca} and Q_{ac}, we would like to know whether Q_c and Q_a are semantically consistent (i.e., use the same set of authors and articles). If they are, we



(a)

Influential Researchers	
AUTHOR-ID	INFLUENCE
8	2.4
6	2.2
2	1.8
4	1.5
10	1.3
3	1.3
...	...

(b)

Fig. 1. (a) a simple network with collaborative communities described by dashed circles; (b) a collection of influential researchers can leverage the results of Q_c and Q_a to efficiently answer Q_{ca} and Q_{ac}. Ideally, we would also like to analyze the changes of collaborative communities and influential researchers over time to discover unknown interactions and trends.

Contributions. The first contribution of this paper is the development of a conceptual modelling method for network analysis applications. We propose the Network Analytics ER model (NAER) that extends the concepts of the traditional ER models in three aspects: (a) the *structural* aspect - analytical types are added; (b) the *manipulation* aspect - topological constructs are added; and (c) the *integrity* aspect - semantic constraints are extended.

Then we introduce an analytical framework for network analysis applications, which has three components: a collection of query topics, a number of small topology schemas, and a relatively large core schema. The *core schema* consists of base types, while topology schemas consists of analytical types that have support from base types in the core schema. A query topic is a tree representing a hierarchy of object classes with each level being built from lower levels, and the leaves of such a tree can be specified using topological constructs over one or more topology schemas, or using the core schema. Topology schemas are usually small and dynamic, which describe topological structures of interest based on query requirements. The reason for having small topology schemas is to support flexible abstraction on topological structures.

We further develop the design guidelines of establishing such an analytical framework for network analysis applications. The key idea is that, in addition to data requirements, query requirements should also be taken into account in the modelling process. This enables an integrated view on the semantics of analysis tasks, and can thus provide a conceptual platform for sharing the theories and algorithms behind different analytical models. In doing so, such a conceptual model can circumvent the design limitations of conventional modeling techniques which do not consider analysis queries. It thus brings us several significant advantages for managing analysis tasks in networks, such as managing the complexity of computational models, handling the semantic integration of different data analysis results, and enabling comparative network analysis.

Outline. The remainder of the paper is structured as follows. We start with a motivating example in Section 2. Then we introduce the NAER model in Section 3. After that, we present a high-level overview for the analytical framework of network analysis applications, and discuss the general design principles that underlie the development of such an analytical framework in Section 4. We discuss the related works in Section 5 and conclude the paper in Section 6.

2 Motivating Example

We start with a bibliographical network, i.e., each article is written by one or more authors, an article is published in a conference or a journal, and one article may cite a number of other articles. Using the traditional ER approaches [5, 19], one can design a simple ER diagram as depicted in Fig. 2.

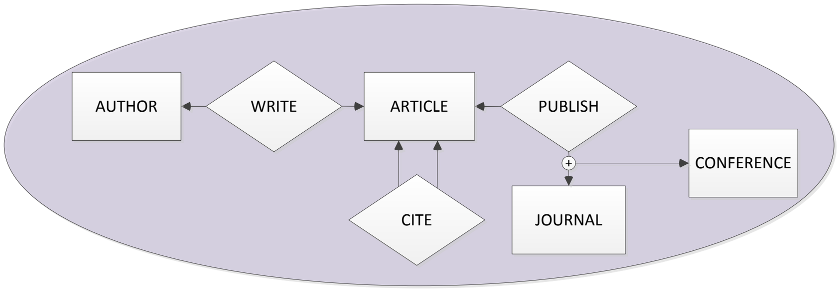


Fig. 2. An ER diagram

Based on this network, a variety of network analysis tasks can be performed. Typical examples include: community detection [8, 10] that is to identify sets of entities that have certain common properties, cocitation analysis [4] that is to identify sets of articles that are frequently cited together, and link predication [14] that is to find out links among entities which will probably appear in the future. We exemplify some of such analysis tasks by the following queries.

Q1: (Collaborative communities) *Find the communities that consist of authors who collaborate with each other to publish articles together.*

Q2: (Most influential articles) *Find the most influential article of each VLDB conference, together with the authors of the article.*

Q3: (Top-k influential researchers) *Find the top 10 influential researchers in terms of the influence of articles (i.e., the citation counts) they have published.*

Q4: (Correlation citation) *Find the correlation groups of journals which publish articles that are often cited by each other.*

Conceptually, these network analysis queries either require or generate some entities and relationships that are not explicitly represented in Fig. 2. For instance, the query Q1 generates a set of author groups, each being referred as a *collaborative community*, and the detection of such collaborative communities is based on the *coauthorship* relationships between authors, i.e., two authors have written an article together.

Capturing implicit entities and relationships, and represent them explicitly in a conceptual model can bring several benefits for network analysis applications: (1) It enables semantic integrity checking across different analysis results. (2) It supports comparative analysis on different dimensions in order to predict trends and discover new insights. (3) It can improve query performance by reformulating queries in a way that can leverage existing results whenever possible. Nevertheless, how should we specify such entities and relationships? Take the query Q1 for example, the question is how to model the concept of *collaborative community* and the relationship of *coauthorship* among authors. In most cases, they are algorithmically defined, without a precise a priori definition. Motivated by these questions, we will discuss the NAER model in Section 3.

3 Network Analytics ER Model

Our NAER model extends the concepts of the traditional ER models in three aspects: (a) the *structural* aspect - analytical types are added; (b) the *manipulation* aspect - topological constructs are added; and (c) the *integrity* aspect - semantic constraints are extended.

3.1 Base Types vs Analytical Types

Two kinds of entities and relationships are distinguished in the NAER model: (1) *base entity and relationship types* contain entities and relationships, respectively, as defined in the traditional ER models; (2) *analytical entity and relationship types* contain analytical entities and relationships, respectively, such that

- an *analytical entity* is an object of being analyzed, which may be a concrete thing or an abstract concept;
- an *analytical relationship* is a link among two or more analytical entities.

Base and analytical types serve rather different purposes. Base types specify first-class entities and relationships from the data management perspective, and analytical types specify first-class entities and relationships from the data analysis perspective. These two perspectives may lead to different decisions about which entities and relationships to emphasise, and which to ignore. For example, COAUTHORSHIP and AUTHOR are often interesting analytical types to consider in network analysis queries like Q1, but the corresponding base types AUTHOR, WRITE and ARTICLE are more natural and informative for managing what entities involve and how they interact.

In the NAER model, base types are the root from which analytical types can be derived. Let $\mathcal{B}(\Upsilon)$ be a set of base types that represent data in a network Υ. Then a set $\mathcal{A}(\Upsilon)$ of analytical types in Υ can be defined over $\mathcal{B}(\Upsilon)$ such that each $A \in \mathcal{A}(\Upsilon)$ is determined by a subset of base types in $\mathcal{B}(\Upsilon)$, and these base types that define A are called the *support* of A, denoted as $supp(A)$. To ensure that analytical types are well-defined, the following criteria must be applied:

- $supp(A) \subseteq \mathcal{B}(\Upsilon)$ for each analytical type A;
- $supp(A_E) \subseteq supp(A_R)$ for each analytical relationship type A_R, and every analytical entity type A_E that associates with A_R.

An analytical type A may have attributes, each of which must be derivable from the base types in its support $supp(A)$. To avoid redundant information, it is prohibited to have attributes in an analytical type as a copy of some attributes in base types. A *schema* S consists of a set of connected and well-defined types that are *complete*, i.e., if a relationship type $T_R \in S$, then for every type T that participates in T_R, we have $T \in S$.

Example 2. Suppose that all the entity and relationship types in Fig. 2 are base types in the network Υ_{bib} , then we can define several analytical types over these base types, as depicted in Fig. 3.(a)-(c), i.e.,

- $\mathcal{B}(\Upsilon_{bib}) = \{$AUTHOR, ARTICLE, CONFERENCE, JOURNAL, WRITE, CITE, PUBLISH$\}$;
- $\mathcal{A}(\Upsilon_{bib}) = \{$AUTHOR*, COAUTHORSHIP, ARTICLE*, CITATION, JOURNAL*, COCITATION$\}$.

Both COAUTHORSHIP and COCITATION may have an attribute WEIGHT, which respectively indicate how many articles two authors have written together, and how many times two journals are cocited by articles. The analytical types in $\mathcal{A}(\Upsilon_{bib})$ have the following support:

(a) *supp*(AUTHOR*) = $\{$AUTHOR$\}$ and
supp(COAUTHORSHIP) = $\{$AUTHOR, ARTICLE and WRITE$\}$;
(b) *supp*(ARTICLE*) = $\{$ARTICLE$\}$ and *supp*(CITATION) = $\{$ARTICLE and CITE$\}$;
(c) *supp*(JOURNAL*) = $\{$JOURNAL$\}$ and
supp(COCITATION) = $\{$ARTICLE, CITE, JOURNAL and PUBLISH$\}$.

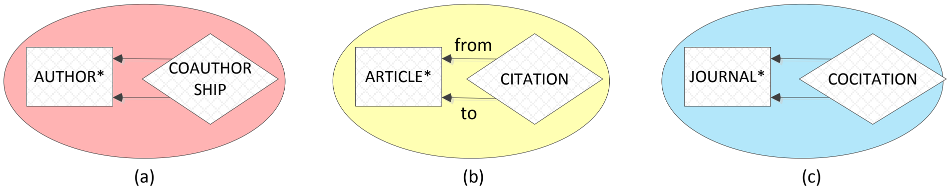


Fig. 3. (a) coauthorship schema S_{co}; (b) citation schema S_{ci}; (c) cocitation schema S_{jo}

3.2 Topological Constructs

A common scenario in network analysis is to analyze topological structures that are hidden underneath base entities and relationships. To explicitly represent a topological structure of interest, one can define analytical entities as vertices and analytical relationships as edges in a graph that may be directed or undirected, weighted or unweighted, etc. However, as illustrated by the following example, base and analytical types alone are still not sufficient to provide a clearly defined conceptual description for network analysis applications.

Example 3. To analyze collaborative communities as described in the query Q1, we may design the coauthorship schema S_{co}, i.e., Fig. 3.(a), consisting of the analytical entity type AUTHOR* and the analytical relationship type COAUTHORSHIP. Nevertheless, the problem of how to model the concept of *collaborative community* in terms of S_{co} still remains. Solving this problem requires us to take into account topological measures and operators, together with analytical types.

Topological measures play an important role in characterizing topology properties of a network [2, 12]. Two of the most commonly used topological measures are centrality and similarity. Let A be an analytical type.

- CENT: $A \mapsto \mathbb{N}$ is a centrality measure that describes how central elements are in A, and return a rank CENT(v) for an element v. This measure can be implemented in different ways, such as degree, betweenness and closeness centrality [9].
- SIMI: $A \times A \mapsto \mathbb{N}$ is a similarity measure that describes the similarity between two elements in A, and generates a rank SIMI(v_1, v_2) for a pair (v_1, v_2) of elements. This measure can also be implemented in different ways, such as q-gram, adjacency-based and distance-based similarity [8].

Based on topological measures, we introduce two families of topological constructs in the NAER model - clustering and ranking. Let S be a schema, $T \in S$, and m be a topological measure. Then we have

(1) CLUSTER-BY(S, T, m) that contains a set of clusters over T, according to the structure specified by S and the measure m;
(2) RANK-BY(S, T, m) that contains to a set of ranked elements over T, according to the structure specified by S and the measure m.

A CLUSTER-BY construct classifies a set of elements over A into a set of clusters (i.e., each cluster is a set of elements), while a RANK-BY construct assigns rankings to a set of elements over A. Both CLUSTER-BY and RANK-BY constructs need to be augmented with a topological measure. These topological constructs provide us an ability to specify existing prominent techniques of network analysis into the conceptual modelling process without being exposed to low-level implementation details.

Example 4. Consider the following concepts relating to the queries Q1-Q4.

- *Collaborative community* in the query Q1 can be modelled using

  $$\text{CLUSTER-BY}(S_{co}, \text{AUTHOR}^*, \text{CENT-CLOSENESS}).$$

  That is, each collaborative community is a group of authors in a network specified by S_{co}, and the measure for determining community membership is closeness centrality.
- *Influence of article* in the queries Q1-Q2 can be modelled using

  $$\text{RANK-BY}(S_{ci}, \text{ARTICLE}^*, \text{CENT-INDEGREE}).$$

  That is, each article is associated with a ranking that indicates its influence in terms of a network specified by S_{ci}, and the measure for determining rankings is indegree centrality.
- *Correlation group* in the query Q4 can be modelled using

  $$\text{CLUSTER-BY}(S_{jo}, \text{JOURNAL}^*, \text{CENT-BETWEENNESS}).$$

  That is, each correlation group contains journals that are correlated in a network specified by S_{jo} and the measure for determining the correlation among journals is betweenness centrality.

3.3 Integrity Constraints

In the NAER model, integrity constraints that are allowed in the traditional ER models can be extended to analytical entity and relationship types in a similar manner. Moreover, we can also define integrity constraints over topological constructs. The following are some typical constraints:

- DISJOINT *(resp.* OVERLAPPING*) constraints on* CLUSTER-BY
  Clusters identified by a CLUSTER-BY construct must be disjoint, i.e., no element can be a member of more than one cluster, (resp. can be overlapping).
- CONNECTED *constraints on* CLUSTER-BY
  For each cluster identified by a CLUSTER-BY construct, there is a path between each pair of its members, running only through elements of the cluster.
- EDGE-DENSITY *constraints on* CLUSTER-BY
  For each cluster identified by a CLUSTER-BY construct, its members have more edges inside the cluster than edges with other members who are outside the cluster.
- TOTAL *(resp.* PARTIAL*) constraints on* RANK-BY
  Every element in a given set must be (resp. may not necessarily be) ranked by a RANK-BY construct.

4 Analytical Framework

In this section, we discuss how to use the NAER model to establish an analytical framework for network analysis applications at the conceptual level.

4.1 High-level Overview

Fig. 4 illustrates an analytical framework of the bibliographical network described in our motivating example. In general, such an analytical framework has three components $\langle \mathcal{S}_q, \mathcal{S}_t, S_c \rangle$: (1) a collection of query topics $\mathcal{S}_q$, (2) a number of small topology schemas $\mathcal{S}_t$, and (3) a relatively large core schema S_c. The *core schema* S_c contains a set of base types. Each *topology schema* $S \in \mathcal{S}_t$ contains a set of analytical types, and the support of each analytical type in S is a subset of base types in S_c. Each *query topic* in $\mathcal{S}_q$ is a tree representing a hierarchy of object classes with each level being built from lower levels, and the leaves of such a tree can be specified using topological constructs CLUSTER-BY or RANK-BY over one or more topology schemas, or using the core schema if the attributes of base types need to be processed.

In Fig. 4, three topology schemas $\{S_{co}, S_{ci}, S_{jo}\}$ are built upon the core schema, which represent three topological structures that are of interest for network analysis queries over Υ_{bib}: (1) the coauthorship schema S_{co} for the query Q1, (2) the citation schema S_{ci} for the queries Q2 and Q3, and (3) the cocitation schema S_{jo} for the query Q4. Consequently, the four queries Q1-Q4 lead to four query topics, in which the query topics of the queries Q2 and Q3 are overlapping and having the same leave INFLUENCE OF ARTICLE (will be discussed in detail in the next subsection).

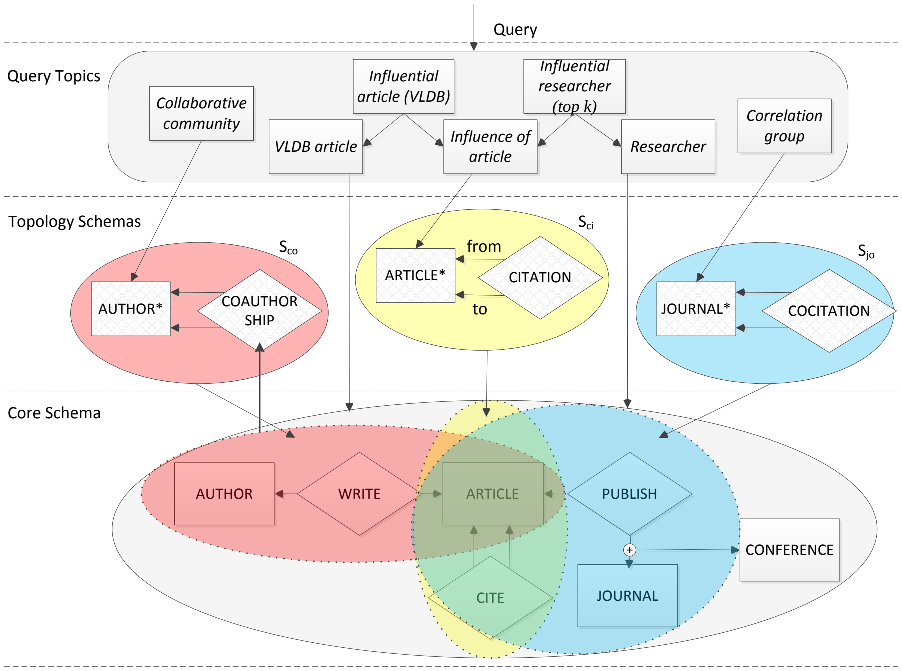


Fig. 4. An analytical framework

4.2 Design Principles

We now present the design guidelines that support the development of an analytical framework for network analysis applications. The central idea is to incorporate both data and queries into the conceptual modelling process. Generally, there are six steps involved:

(1) Identify data requirements (i.e., a set of business rules of interest);
(2) Design the core schema based on the data requirements;
(3) Identify query requirements (i.e., a set of analysis queries of interest);
(4) Design topology schemas based on the query requirements and query topics;
(5) Identify constraints on the query topics, and core and topology schemas.

The steps (1) and (2) are exactly the same as in the traditional ER models, the steps (3) and (4) are additional but critical for network analysis applications, and the step (5) extends integrity constraints of the traditional ER models to analytical types in topology schemas and topological constructs in query topics accordingly. In the rest of this section, we focus on discussing three key aspects: (i) what are data and query requirements; (ii) how are query requirements and query topics related; and (iii) how are the core and topology schemas designed.

Data and query requirements. Data and queries are two different kinds of requirements. Data requirements describe what information an application should manage, while query requirements describe how the information of an application should be used. Although our NAER model can conceptually represent both data and query requirements for network analysis applications, the questions to be clarified are: (a) Do we need to consider all queries? (b) If not, what are the query requirements of interest?

Queries in network analysis applications may exist in various forms. For example, *database queries* in the traditional sense, such as "find all journal articles published in 2013", often use a database language (e.g., SQL) to process data, and *analysis queries* from a topological perspective, such as the queries Q1, Q3 and Q4, often use certain data mining and machine learning techniques to process data. In a nutshell, database queries and analysis queries are fundamentally different in two respects:

- *Logical vs topological*: Database queries are concerned with the logical properties of entities and relationships, while analysis queries focus on the topological properties of entities and relationships. In most cases, analysis queries are formulated using software tools in a much more complicated way than database queries.
- *Indefinite vs definite*: Analysis queries often have indefinite answers, which depends on not only the underlying structure but also the choice of topological measures. It can be difficult to know which measure is better than the others, and which answer is optimal. In contrast, database queries have definite answers that are determined by the underlying database.

In many real-life applications, analysis and database queries are commonly combined in order to find useful information [17]. For example, the query Q2 can be viewed as the combination of an analysis query "*find the most influential articles*" and a database query "*find articles of each VLDB conference, together with the authors of the article*".

When designing a conceptual model for network analysis applications, we are only interested in analysis queries. There are two reasons: (1) analysis queries are often computationally expensive so that modelling analysis queries can help improve performance; (2) analysis queries are often isolated so that modelling analysis queries can help maintain their semantic integrity. Therefore, given a set $\mathcal{Q}$ of queries for modelling a network analysis application, which may contain database queries, analysis queries or a combination of both, queries in $\mathcal{Q}$ are first transformed into $\mathcal{Q}'$ by removing any database queries in $\mathcal{Q}$.

Query topics. After identifying query requirements, i.e., queries of interest, we need to analyze these queries to understand their semantics and required computations. Analyzing queries is to unravel the structures of queries, which has at least two aspects to consider: (1) the structure of a query, and (2) the structure among a set of queries. Since queries may be described in various syntactical forms, here we focus on exploiting the semantic structures of queries.

For each query Q, we associate it with a query topic $t(Q)$, which is a tree with each node C corresponding to an object class, and an edge from a node

C_1 to a node C_2 expressing that C_1 depends on C_2. This dependence relation between object classes is closed under transitivity, i.e., if C_1 depends on C_2, and C_2 depends on C_3, then C_1 depends on C_3. The query topic $t(Q)$ of a query Q can be defined at *a flexible level of abstraction*. That is, the level of granularity for nodes in a query tree is a design choice depending on individual applications.

For each query Q, we thus have a set of object classes that are in one-to-one correspondence with the nodes of $t(Q)$. A node $C_1 \in t(Q_1)$ in one query topic may have certain relationships with a node $C_2 \in t(Q_2)$ in a different query topic. Such relationships include that: (1) C_1 depends on C_2; or (2) C_1 and C_2 are the same. Nevertheless, it is impossible that C_1 depends on C_2, and meanwhile C_2 depends on C_1 or any of its descendant nodes. If two different query topics $t(Q_1)$ and $t(Q_2)$ contain the same node C, then $t(Q_1)$ and $t(Q_2)$ are connected by the node C, and merged as one tree.

Example 5. Consider the queries Q1-Q4 in our motivating example. We have one query topic for each of the queries as depicted in Fig. 5.(a), and three trees corresponding to the whole set $\{Q_1, Q_2, Q_3, Q_4\}$ as depicted in Fig. 5.(b).

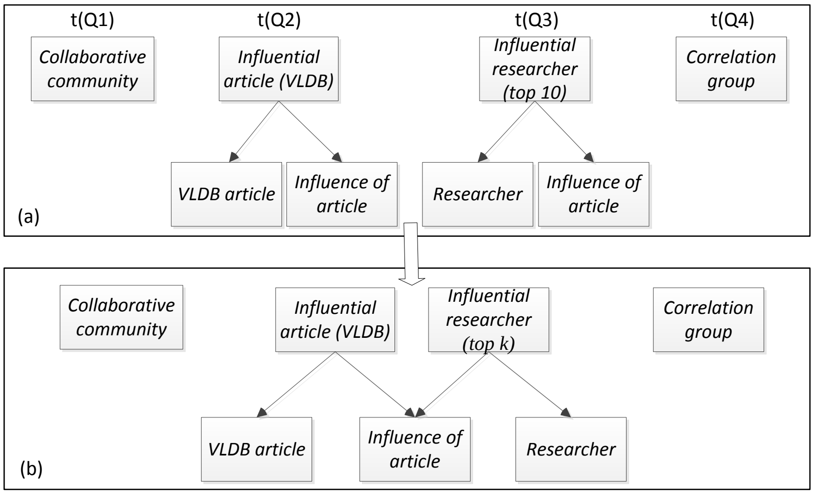


Fig. 5. Query topics (a). for individual queries; (b) for a set of queries

Core and topology schemas. For a network analysis application, the design of its core schema and topology schemas is carried out in two steps. First, the core schema is designed based on data requirements as in the traditional ER models. Second, the topology schemas are designed based on query requirements following a method of grouping the leaves of query topics that are associated with query requirements. All leaves that can be handled by database queries over the core schema are grouped together, while the other leaves are grouped in terms of what analytical types they need for analysis, and each of such groups correspond to one topology schema. It also implies that, each object class represented by a leave corresponding to some topology schema S_t can be specified by using a topological construct over S_t. In general, the central idea is that all data requirements should be captured by the core schema, and the analysis part of all query requirements should be captured by a collection of topology schemas.

Example 6. For the query topics of Q1-Q4, we can group their leaves as below, where C_i denotes the leave with the initials i. As a result, three topology schemas $\{S_{co}, S_{ci}, S_{jo}\}$ can be designed as described in Fig. 4.

Queries	Core schema	Topology schemas		
		S_{co}	S_{ci}	S_{jo}
Q1		C_{cc}		
Q2	C_{va}		C_{ioa}	
Q3	C_r		C_{ioa}	
Q4				C_{cg}

Example 4 showed that the object classes *collaborative community*, *influence of article* and *correlation group*, which are respectively represented by C_{cc}, C_{ioa} and C_{cg}, can be specified using topological constructs.

One distinguished feature of topology schemas is that, rather than taking objects in all their complexity, topology schemas only focus on specifying a simple but concise representation for objects. Therefore, topology schemas need to be designed in accordance with the following criteria:

1. *Topology schemas should be small.* Topology schemas are the basic building blocks of supporting analysis queries. The smaller topology schemas are, the easier they can be composed to support flexible modelling needs.
2. *Topology schemas should be dynamic.* Query requirements may be changing over time. Correspondingly, topology schemas need to be adaptive enough to reflect the dynamics of query requirements.

Two topology schemas in an analytical framework may be overlapping. In fact, certain degree of overlapping can facilitate comparative analysis over different topology schemas. Nevertheless, duplicate topology schemas should be avoided because this would cause redundant storage and inconsistence. The following example shows that our analytical framework supports an integrated and coherent view on core and topology schemas.

Example 7. The three topology schemas $\{S_{co}, S_{ci}, S_{jo}\}$ can be composed by leveraging base types in the core schema. Fig. 6 shows three possible compositions: (a) three topology schemas are composed by an analytical relationship type HAS that is determined by several base types; (b) two schemas are composed by a base relationship type (i.e., PUBLISH and WRITE) directly.

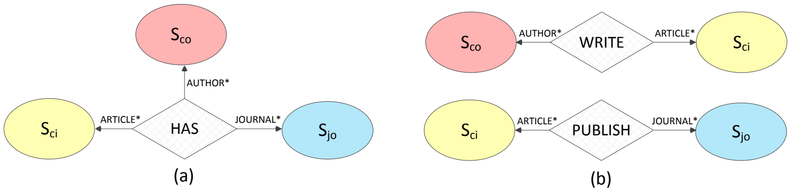


Fig. 6. Composing core and topology schemas

5 Related Works

Recently, a number of works have proposed to use database technologies for managing and analyzing network analysis [6, 7, 20]. However, they have mostly focused on designing logical data models and their corresponding query languages for supporting network analysis. So far, only very limited work has considered the design process of conceptual modeling [1]. In general, the previous works on modelling network analysis applications at the logical level fall into two lines of research:

(1) Extending traditional database technologies (i.e., the relational model and SQL) to support data mining algorithms, such as SiQL [20] and Oracle Data Miner.
(2) Extending object-oriented or graph database technologies to incorporate graph-theoretic and data mining algorithms, such as GOQL [18], and other works discussed in the survey paper [21].

Our work in this paper focused on the conceptual modelling of network analysis, and leaves the transformation to a logical model (e.g., the relational model, a graph model or a combination of several data models) as a decision of the user. For example, in [17], a hybrid memory and disk engine was developed for evaluating queries, which maintains topological structures in memory while the data is stored in a relational database. An analytical framework designed in our work can be well transformed into this data model and be implemented over the hybrid engine by separating topological structures specified by topology schemas from the database structure specified by the core schema.

6 Conclusions

In this paper, we proposed the NAER model and a conceptual modelling paradigm that incorporates both data and query requirements of network analysis. This was motivated by the rapid growth of network analysis applications. Such a conceptual view of network analysis applications can enable us to better understand the semantics of data and queries, and how they interact with each other. In doing so, we can avoid unnecessary computations in network analysis queries and support comparative network analysis in a dynamical modeling environment.

In the future, we plan to implement the NAER model, and based on that to establish an analytical framework for supporting network analysis applications, including the development of a concrete modelling language for network analysis and a query engine for processing topic-based queries.

References

1. Z. Bao, Y. Tay, and J. Zhou. sonSchema: A conceptual schema for social networks. In *Conceptual Modeling*, pages 197–211. 2013.
2. U. Brandes and T. Erlebach. *Network analysis: methodological foundations*, volume 3418. Springer, 2005.
3. T. Brughmans. Connecting the dots: towards archaeological network analysis. *Oxford Journal of Archaeology*, 29(3):277–303, 2010.
4. C. Chen, I.-Y. Song, and W. Zhu. Trends in conceptual modeling: Citation analysis of the ER conference papers (1979-2005).
5. P. Chen. The entity-relationship model – toward a unified view of data. *ACM TODS*, 1(1):9–36, 1976.
6. S. Cohen, L. Ebel, and B. Kimelfeld. A social network database that learns how to answer queries. In *CIDR*, 2013.
7. A. Dries, S. Nijssen, and L. De Raedt. A query language for analyzing networks. In *CIKM*, pages 485–494, 2009.
8. S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.
9. L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1979.
10. M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
11. M. Huisman and M. A. Van Duijn. Software for social network analysis. *Models and methods in social network analysis*, pages 270–316, 2005.
12. A. Jamakovic and S. Uhlig. On the relationships between topological measures in real-world networks. *Networks and Heterogeneous Media*, 3(2):345, 2008.
13. R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Link mining: models, algorithms, and applications*, pages 337–357. Springer, 2010.
14. L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.
15. M. E. Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical review E*, 64(1):016131, 2001.
16. G. A. Pavlopoulos, A.-L. Wegener, and R. Schneider. A survey of visualization tools for biological network analysis. *Biodata mining*, 1(1):1–11, 2008.
17. S. Sakr, S. Elnikety, and Y. He. G-SPARQL: a hybrid engine for querying large attributed graphs. In *CIKM*, pages 335–344, 2012.
18. L. Sheng, Z. M. Ozsoyoglu, and G. Ozsoyoglu. A graph query language and its query processing. In *ICDE*, pages 572–581, 1999.
19. B. Thalheim. *Entity-relationship modeling: foundations of database technology.* Springer, 2000.
20. J. Wicker, L. Richter, K. Kessler, and S. Kramer. SINDBAD and SiQL: An inductive database and query language in the relational model. In *Machine Learning and Knowledge Discovery in Databases*, pages 690–694. 2008.
21. P. Wood. Query languages for graph databases. *ACM SIGMOD Record*, 41(1):50–60, 2012.