ABSTRACT
The star schema is widely accepted as the de facto data model for data warehouse design. A popular approach for developing a star schema is to develop it from an entity-relationship diagram with some heuristics. Most of the existing approaches analyze the semantics of an ERD to generate a star schema. In this paper, we present the SAMSTAR method, which semi-automatically generates star schemas from an ERD by analyzing its semantics as well as structure. The novel features of SAMSTAR are (1) the use of the notion of Connection Topology Value (CTV) in identifying the candidates of facts and dimensions and (2) the use of Annotated Dimensional Design Patterns (A_DDP) as well as WordNet to extend the list of dimensions. We illustrate our method by applying it to the examples from existing literature. We prove that the outputs of our method are a superset of those of the existing methods. The SAMSTAR method simplifies the work of experienced designers and gives a smooth head-start to novices.

Categories and Subject Descriptors
H.2.1 [Logical Design] Data Models; H.2.7 [Database Administration] Data warehouse and repository.

General Terms

Keywords
ER Diagram, Star Schema, Data warehouse Design, Connection Topology Value, WordNet, Dimensional Design Patterns.

1. INTRODUCTION
In this age of knowledge, data warehousing is undoubtedly a blooming field. The most popular data structure used in data warehouses (DWs) is the star schema or the dimensional model popularized by Kimball [18]. Most companies already have entity-relationship diagrams (ERDs) for their online transaction processing (OLTP) systems. Since an ERD contains more information about a particular business domain than a star schema, it is possible to convert an ERD to a star schema.

To develop a star schema using an ERD, most of existing approaches analyze the attributes of interesting business entities. The entities that have numerical measure attributes are assumed to be the candidates of facts and the entities that have non-numerical and descriptive attributes are assumed to be the candidates of dimensions. These approaches are qualitative in nature and focus only on the semantics of an ERD to generate a star schema.

In this paper, we present a Semi-Automated lexical Method for generating STAR schemas from an entity-relationship diagram (SAMSTAR), which semi-automatically generates star schema(s) from an ERD by analyzing the semantics as well as the structure of the given ERD. The novel features of SAMSTAR are (1) the use of the notion of Connection Topology Value in identifying the candidates of facts and dimensions and (2) the use of Annotated Dimensional Design Patterns as well as WordNet [6] to extend the list of dimensions. We illustrate our method by applying it to the examples from existing literature. We demonstrate that the outputs of our method are a superset of those of the existing methods. The SAMSTAR method simplifies the work of experienced designers and gives a smooth head-start to novices.

The remainder of this paper is organized as follows. Section 2 reviews various approaches used to design a multidimensional model for data warehouses. Section 3 describes the key concepts and the algorithm of the SAMSTAR method. Section 4 illustrates the case studies and Section 5 evaluates our method. Section 6 suggests possible research directions in this field.

2. RELATED LITERATURE REVIEW
In this section we review the existing research work on DW design methods. Our subject of interest is logical schema design and therefore, we review only this phase for each piece of related literature. The survey shows that these methods are mainly driven by the following three factors:

User/Demand-Driven: This approach gives the highest priority to user needs. Different user groups are interviewed to get an extensive list of requirements. User requirements could vary across organizational levels and could change over time. Hence,
such approaches often result in unstable schemas. Domain experts are needed to perform such kind of analysis.

**Source/Supply/Data-Driven:** It ensures that the structure of the data warehouse reflects the semantic structure of OLTP data. This structured approach could be highly automated. The resulting schema is resilient to changes in analysis over time. It also helps in developing dimension hierarchies and simplifying the ETL process design. Minimal time is required to start the project. Pure data-driven approaches, however, could ignore business goals and user needs and hence, could generate inadequate schemas.

**Goal-Driven:** This approach focuses on business goals and aligns itself to business processes, rather than user needs or available data sources.

A DW design, however, cannot be merely based on a user-requirements list or an ERD or business goals. In real world, most of the design methods suitably blend two or more of these factors to engender hybrid methods.

Giorgini, Rizzi, and Garzetti [8] present a DW design method that primarily goal driven and secondarily supply and demand-driven. It uses an existing ERD and employs them during the design process; this makes the process primarily goal driven. It uses an existing ERD and user needs to design a logical schema. The requirements are derived directly from the analysis of stakeholders’ and decision makers’ goals. However, this approach relies heavily on the expertise of the analyst.

Prat, Akoka, and Comyn-Wattiau [23] present a UML-based data warehouse design method, which is primarily demand-driven and secondarily data-driven. It considers user needs and applies UML transformations to design a data warehouse. It provides a multidimensional metamodel, which could be used in the context of MOLAP and ROLAP tools. This semi-automated method has to be further tested on extensive case studies, and mapping transformations need to be enriched.

Chen and Hsu [4] propose a methodology to convert an ERD into a snowflake schema, which is primarily demand-driven and secondarily data-driven. It uses an existing ERD and applies an algorithm on it. The method is primarily demand-driven because it first requires the designer to manually identify the fact table and then proceeds with adding dimensions to the fact table. One major drawback of this paper is that it requires a fact table to be manually identified before making use of the proposed algorithm. The SAMSTAR method overcomes this drawback by being able to automatically determine the candidates of facts from an ERD.

Guo et al. [13] proposes a triple driven approach. It is primarily data-driven and secondarily goal and user-driven. The design begins with requirements analysis phase and goes through conceptual and logical phases thereafter. The steps of the method are mostly subjective and qualitative.

The SAMSTAR method is also primarily data-driven and secondarily user and goal driven. This method is different from the aforementioned methods in two aspects. First, it relies upon connection structure of an ERD and not just the attributes of various entities in the ERD. Second, it is quantitative in nature and thus could be highly automated.

A growing interest in automating such methods has been observed in the last few years. Jensen, Holmgren, and Pedersen [16] present an automatable method for discovering correct multidimensional structure in the relational data source ensuring the general quality of the output. However, the procedure to identify fact tables is semi-automated. Phipps and Davis [22] and Prat, Akoka, and Comyn-Wattiau [23] also propose semi-automated methods for generating multidimensional schemas. Our method is different from these semi-automated and other manual methods, in that it automatically identifies the candidates of facts from a large ERD.

Moreover, we have incorporated the work by Jones and Song [17], in our method. This paper presents the Dimensional Design Patterns (DDPs) and their usefulness in designing dimensional models. These patterns provide a framework for listing all the core dimensions used in various business domains. The categories are shown in Figure 1. Each of the six sub-classes (‘Temporal,’ ‘Action,’ ‘Location,’ ‘Object,’ ‘Stakeholder,’ and ‘Qualifier’) of the domain dimension could be instantiated to an appropriate dimension entity e.g. in an order processing system, ‘Stakeholder’ could be instantiated to “Customer” and “Sales Representative.” ‘Temporal’ could be instantiated to “Time,” “Period” or “Date”.

Chen and Hsu [4] propose a methodology to convert an ERD into a snowflake schema, which is primarily demand-driven and secondarily data-driven. It uses an existing ERD and applies an algorithm on it. The method is primarily demand-driven because it first requires the designer to manually identify the fact table and then proceeds with adding dimensions to the fact table. One major drawback of this paper is that it requires a fact table to be manually identified before making use of the proposed algorithm. The SAMSTAR method overcomes this drawback by being able to automatically determine the candidates of facts from an ERD.

Guo et al. [13] proposes a triple driven approach. It is primarily data-driven and secondarily goal and user-driven. The design begins with requirements analysis phase and goes through conceptual and logical phases thereafter. The steps of the method are mostly subjective and qualitative.

The SAMSTAR method is also primarily data-driven and secondarily user and goal driven. This method is different from the aforementioned methods in two aspects. First, it relies upon connection structure of an ERD and not just the attributes of various entities in the ERD. Second, it is quantitative in nature and thus could be highly automated.

A growing interest in automating such methods has been observed in the last few years. Jensen, Holmgren, and Pedersen [16] present an automatable method for discovering correct multidimensional structure in the relational data source ensuring the general quality of the output. However, the procedure to identify fact tables is semi-automated. Phipps and Davis [22] and Prat, Akoka, and Comyn-Wattiau [23] also propose semi-automated methods for generating multidimensional schemas. Our method is different from these semi-automated and other manual methods, in that it automatically identifies the candidates of facts from a large ERD.

Moreover, we have incorporated the work by Jones and Song [17], in our method. This paper presents the Dimensional Design Patterns (DDPs) and their usefulness in designing dimensional models. These patterns provide a framework for listing all the core dimensions used in various business domains. The categories are shown in Figure 1. Each of the six sub-classes (‘Temporal,’ ‘Action,’ ‘Location,’ ‘Object,’ ‘Stakeholder,’ and ‘Qualifier’) of the domain dimension could be instantiated to an appropriate dimension entity e.g. in an order processing system, ‘Stakeholder’ could be instantiated to “Customer” and “Sales Representative.” ‘Temporal’ could be instantiated to “Time,” “Period” or “Date”.
3. THE SAMSTAR METHOD
The overall structure of SAMSTAR method is described by the following component diagram. Our method is a conglomeration of three techniques: Annotated DDP (A_DDP), described in Section 3.1.7; WordNet, a lexical database for English language; and Quantitative Analysis.

SAMSTAR is based on relationships and cardinalities. Instead of determining and classifying the attributes of all entities, we focus on the relationships between the entities. In other words, we are interested in the structure of the ERD.

A very large ERD would produce a long list of candidate facts belonging to diverse business processes. This method requires users to decide facts (related to a single business process such as sales, inventory, deliveries, and orders) from the list of candidate facts and lets the algorithm find the related dimensions of the decided facts.

A DW is an essential component of decision support systems in large organizations and critical decisions require human input. In the same lines, a star schema design should not be solely based on an ERD. Our method automates the process as much as possible. The algorithm has 11 steps, out of which 4 are manual and 7 are automated.

3.1 Key Concepts and Ideas
SAMSTAR is based on certain observations and concepts, which can be listed up in the form of following ideas:

1. Facts and Dimensions
2. Direct and Indirect many-to-one relationships
3. Connection Topology Value (CTV)
4. High CTV and Threshold
5. Candidates of Dimensions
6. Pre-processing the ERD
7. Annotated DDP (A_DDP)
8. Post-processing of star schemas

3.1.1 Facts and Dimensions
The first idea is related with facts and dimensions. Figure 3 shows a star schema. Entities in the middle of schema are fact entities and entities surrounding facts are dimension entities.
It has been observed that there usually exists a many-to-one (M:1) relationship between a fact and a dimension. So, we make two heuristic rules related to ERDs.

- Entities, lying on many side of an M:1 relationship, are the candidates of fact.
- Entities, lying on one side of an M:1 relationship, are the candidates of dimensions.

Each M:1 relationship defines a dimension path. Therefore, a fact table comes from an entity which has a relatively high number of M:1 relationships.

### 3.1.2 Direct and Indirect many-to-one Relationships

There are two different types of M:1 relationships; direct and indirect relationships. See Figures 4a and 4b for direct and indirect relationships respectively. The target of an arrow points to the entity on the 1-side of the relationship.

![Figure 4a. Direct many-to-one relationship](image)

![Figure 4b. Indirect many-to-one relationship](image)

This idea is based on transitive cardinality relationships. If there are direct relationships between D and E and between E and F (Figure 4b); then there is surely an indirect relationship between D and F.

Table 1 shows various transitive relationships resulted from multiple direct relationships in Figure 5. The quantitative portion of the SAMSTAR method provides an equation to calculate how many M:1 relationships (both direct and indirect) an entity has.

![Figure 5. Transitive Indirect Relationships](image)

<table>
<thead>
<tr>
<th>Table 1. Transitive relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>A:B (direct)</td>
</tr>
<tr>
<td>1:M</td>
</tr>
<tr>
<td>1:M</td>
</tr>
<tr>
<td>M:1</td>
</tr>
<tr>
<td>M:1</td>
</tr>
</tbody>
</table>

A direct M:1 relationship represents stronger semantics than an indirect M:1 relationship. All entities connected directly or indirectly to an M-side entity through a single chain of one or more M:1 relationships, represent the same dimension path. For example, the entity A in Figure 4(a) has more dimension paths than the entity D in Figure 4(b). Therefore, we give a higher weighting to direct relationships than indirect relationships, while identifying the candidates of facts and dimensions.

### 3.1.3 Connection Topology Value

The Connection Topology Value (CTV) of an entity is a composite function of the topology value of direct and indirect M:1 relationships. There might be many possible functions that could be used to calculate the CTV of an entity. One could be to add all direct and indirect M:1 relationships the entity has with other entities. But this function does not differentiate a direct from an indirect M:1 relationship. However, we give a higher weighting factor to direct relationships because of the reasons stated in Section 3.1.2.

Hence, the CTV of an entity $e$ could be defined by the following recursive function.

$$CTV(e) = weight_d \times \text{Count}(\text{Node}(e)) + weight_i \times \sum_i CTV(\text{Node}(i)),$$

where
- $CTV(e)$ is the CTV of an entity $e$.
- $Node(e)$ is an entity lying on the 1-side and having direct M:1 relationship with $e$.
- $\text{Count}(\text{Node}(e))$ is the total number of entities having direct M:1 relationships with $e$.
- $weight_d$ is the percentage of the weight given to a direct M:1 relationship.
- $weight_i$ is the percentage of the weight given to an indirect M:1 relationship.

As a direct relationship weighs higher than an indirect relationship for fact entities of star schema structure, we can have a general rule as follows:

$$weight_d > weight_i$$

Suppose we use the following weights and notation:

- $weight_d = 100\%$
- $weight_i = 80\%$
- $\text{Count}(\text{Node}(e)) = n$

Then, the formula for calculating CTV for an entity $e$ becomes:

$$CTV(e) = 1 \times n + 0.8 \sum_{i=1}^{n} CTV(\text{Node}(i))$$

where $i$ represents an entity having a direct M:1 relationship with $e$.

**Example:** The above-mentioned formula will be clarified by following calculations: Suppose we have the following ERD with entities A, B, C, D, E, F, G and H.

![Figure 6. An Example ERD for CTV Calculation](image)
The CTV for each entity is:

CTV(H) = 1*0 + 0.8 * 0 = 0

CTV(G) = 0

CTV(E) = 1*1 + 0.8 * CTV(H) = 1

CTV(B) = 1*1 + 0.8 * (CTV(E)) = 1.8

CTV(C) = 1*2 + 0.8 * (CTV(G) + CTV(F)) = 2

CTV(D) = 1*1 + 0.8 * (CTV(C)) = 1 + 0.8 * 2 = 2.6

CTV(A) = 1*2 + 0.8 * (CTV(B) + CTV(C)) = 2 + 0.8 * (1.8 + 2.6) = 5.04

### 3.1.4 High CTV

We define High CTV value as any CTV value which is higher than the threshold Th.

For an entity e,

CTV(e) > Th, where Th is threshold

Threshold is calculated by following equation, which we adopted from research in power engineering [5].

\[
Th = X + k \cdot \sigma
\]

\[
Th = \frac{\sum_{i=1}^{N} CTV(i)}{N} + k \cdot \frac{\left( \sum_{i=1}^{N} (CTV(i) - X)^2 \right)^{1/2}}{N}
\]

X = Mean

\( \sigma = \text{Standard Deviation} \)

N = total number of entities

k = adjustable parameter

All entities are numbered 1, 2, 3, ..., N.

This equation is used to find out how far an entity’s CTV goes above the average CTV. The entities having the CTVs higher than this threshold value are considered to have a high CTV. Christie [5] has used the same equation to identify the major outlier event days in a power distribution system. In this work, it is believed that the farther an event is from the average, the rarer it is. The paper has used the value of k as 1.5. The value of k is adjustable and can be varied accordingly to desired degree of rarity.

### 3.1.5 Candidates of Dimensions

Once all the facts have been identified, the dimension entities for each fact entity are determined. An entity is a dimension entity if it falls in one of the following two categories:

a. It has a direct 1:M relationship with the fact entity.

b. It has an indirect 1:M relationship with fact entity and one of its synonym entity name matches with one of the entities listed by DDPs [17] or Annotated DDP (Section 3.1.7).

c. Every pair of 1:1 relationship entities is combined into a single entity to avoid any loss of semantics. This rule has been created because dimensions are usually denormalized.

#### 3.1.6 Pre-processing input ERD

The SAMSTAR method takes only binary ERDs as one of its inputs. To convert an ERD into its binary counterpart, we pre-process input ERDs as follows:

1. Pre-process the input ERD to convert it into a Binary ERD.
2. Store Entities and Relationships.

### 3.1.7 Annotated DDP

Following the framework of DDPs described in Section 2, we have created a list of dimensions of each of the six classes by referring to four sources [1, 15, 18, 19], and have instantiated the six classes of DDP to produce a list of 131 commonly used dimension entities. We refer to these entities as Annotated DDP (A_DDPs). These entities are frequently used entities in the business processes such as: Retail Sales and e-commerce, Inventory management, Procurement, Order Management, CRM, HR management, Accounting, Financial Services, Telecommunications and utilities, Education, Transportation, Health care and insurance, Quality control, Strategic business analysis, tracking website user actions, Managing and scaling the webhousing, designing the Webhouse user interface, building clickstream datamarts, designing the website to support webhousing, capturing data points about customer behavior, determining whether a particular web ad is working, and determining whether a customer is able to switch to a competitor.

#### 3.1.8 Post-processing of star schemas

In this manual step, the user is given a choice to post-process the system generated star schemas. This step is required because even though the user selects facts and dimensions from the respective candidate lists; certain issues remain unresolved. These issues are listing of related entities as separate dimensions; redundancy of the ‘time’ dimension; and inappropriate naming of facts and dimensions. Therefore, the user is allowed to merge two or more dimensions; to get rid of ‘time’ or its synonymous dimension; and to rename the inappropriately named facts and dimensions. However, this step is optional and can be conveniently skipped by the user.

### 3.2 SAMSTAR Algorithm

#### Assumptions:

1) A structurally valid ERD [7] is available.
2) A problem statement having consolidated DW users’ requirements is available. Statement should clearly state the following:
   a. Business Goals: Business Process for which data warehouse has to be designed
   b. User Requirements: Primary users of future system and the main measures they would be interested in.

#### Input: an ERD, Problem Statement

#### Output: Star Schema(s)

/* Steps in italics are manual, requiring user-input. */

#### Algorithm SAMSTAR

Begin
1. Pre-process the input ERD to convert it into a Binary ERD.
2. Store Entities and Relationships.
3. Let user choose weighting factors for direct and indirect relationships.
4. Calculate the CTV for all entities.
5. Calculate the threshold value, Th, for CTV.
6. Identify the entities having CTV higher than the threshold Th. These are the candidates for fact tables.
7. Decide and shortlist the fact entities based on the results from Step No. 6 and the problem statement. There could be more than 1 fact table for a business process.
8. For each fact entity, perform the following steps:
   (i) Identify the entities having direct M:1 link with a fact entity.
   (ii) Identify entities having indirect M:1 link with the fact entity.
   Out of these entities, identify synonyms of entity names from WordNet. Extract the terms which match the DDP entity list or the A_DDP List.
   (iii) Combine the results to Steps 8(i) and 8(ii) to prepare a list of candidate dimensions for a given fact.
   (iv) Add ‘time’ dimension to this list.
9. Decide the dimension entities based on problem statement and the result of Step 8.
10. Let the user post-process the ERD:
   (i) Check if ‘time’ is a redundant dimension.
   (ii) Merge two or more related dimensions.
   (iii) Rename the fact and dimension tables.
11. Generate the star schema(s).

We implemented SAMSTAR algorithm using JAVA with the Dragon toolkit [25].

4. EXAMPLE CASE STUDIES
We present the results of applying SAMSTAR to a few examples in the existing literature in this section.

The values of direct (weight_d) and indirect (weight_i) weighting factors were set as 1 and 0.8, respectively. The value of $k$ used in the calculation for threshold CTV was set to 1.5. During the course of the research, we performed several experiments modifying the value of $k$. In the test example by Moody and Kortnik (2000), when $k$ was chosen as 1.5, our program generated 3 fact tables. Multiple fact tables were generated when the condition $0.25 < k \leq 1.5$ held true; and a single fact table was generated when $k$ lied between 1.5 and 1.75. We found similar results with other examples and hence, for all calculations, this paper uses 1.5 as the value of $k$.

The DDP entity list (“calendar,” “time,” “fiscal calendar,” “special period,” “purpose,” “management,” “location,” “contact,” “configuration,” “organization,” “role,” “profile,” “state,” “cause,” and “unit”) and the A_DDP list (131 entities) were used for identification of dimension candidates. WordNet was used to find synonyms of a candidate entity.

4.1 Case Study: Moody and Kortnik [21]
We first tested our algorithm on the example data model of the paper by Moody and Kortnik [21].The input ERD is shown in Figure 7; the star schemas given by the authors are shown in Figure 8; and the output from SAMSTAR is illustrated in Figure 9.

The algorithm identified three entities as the candidates of fact: Sale, Sale Item and Sale Fee, because they had high CTVs. As shown in the figures, our algorithm generated an extra fact entity ('Sale Fee') which was not generated by the method suggested by the authors. ‘Sale Fee’ is not a very commonly occurring fact entity. But as per the structure of the input ERD, it has a high number of M:1 relationships and hence it can be analyzed from multiple point of views. For example, it would be interesting to see the alterations in ‘Sale Fee’ with changes in ‘Location’, ‘Customer’ or ‘Time’ of sale. This shows that SAMSTAR is also capable of discovering those facts, which are not very commonly used but which might be really good candidates of subject of analysis.
For each of these fact candidates, the entities having direct $M:1$ relationships were the obvious candidates of dimensions.

Additional dimensions were found by navigating through each direct $M:1$ link and searching for A-DDPs. For example, for ‘Sale’ fact entity, SAMSTAR identified four obvious candidates for dimensions: ‘Period(1),’ ‘Period(2),’ ‘Customer’ and ‘Location’. These entities have a direct relationship with the Sale entity. Other entities such as ‘Region’ and ‘State’ were added by navigating the direct links from fact entity and verifying their presence in our exhaustive list of A-DDPs. The ‘Time’ dimension candidate was added as it is typically important to measure the fact on the basis of time. The user can get rid of extraneous dimensions during Step9. The post processing step (Step 10 (i)) lets the user choose one of the two synonymous dimensions (‘time’ and ‘period’) and Step 10 (ii) gives the user freedom to merge two similar dimensions (e.g. ‘Region’ and ‘State’).

The diagrams depict that the schemas generated by SAMSTAR are similar to those generated by the manual steps in Moody’s paper [21]. It is obvious that SAMSTAR generated star schemas are a superset (in terms of both the number of facts and the number of dimensions) of those depicted in Figure 8.

4.2 Other Examples
SAMSTAR algorithm was also applied on the examples given in [3, 4, 11]; and quite appropriate results were found. We skip them due to the lack of space.

These case studies reinstate that the schemas generated by SAMSTAR are the superset of the ones given by the authors. This fortifies our claim that SAMSTAR schemas are inclusive of all possible facts and dimensions. Moreover, our schemas have a higher number of facts and dimensions; this gives the designer a helpful aid and he/she could further prune the schema as per the business and user requirements.

5. DISCUSSION OF SAMSTAR
In this section we evaluate the functionality of SAMSTAR. As observed in the case studies in Section 4, the SAMSTAR algorithm semi-automatically generates star schema(s) from a complex ERD. The output is a superset of the one generated manually using some other approach on the same ERD, business goals and user needs. Hence, our method is significant in many design situations and could help a wide variety of designers. By using our method, model designers greatly reduce their work and even the inexperienced designers can produce a robust star schema. SAMSTAR generates the candidates of facts on the basis of the connection topology structure of the entire diagram and identifies the candidates of dimensions on the basis of topology of the diagram and dimensional design patterns. This adds additional dimension entities to the generated star schema. SAMSTAR makes use of WordNet which ensures that the synonyms of DDP and A-DDP entities existing in the ERD are not missed out while preparing the list of candidate dimensions. Most of the existing approaches analyze the semantics of an ERD to generate a star schema, but the SAMSTAR method analyzes the semantics as well as the structure of the input diagram.

Moreover, this method automatically identifies candidates of fact entities from an ERD no matter how complex or simple it is. Therefore, our method is particularly well suited for systems where a source ER schema is available, rich, complex and substantial in size. Also, it can be automated up to a large extent as this method is quantitative in nature.

6. CONCLUSION & FUTURE RESEARCH
In this paper, we have presented the SAMSTAR method, which uses a quantitative approach of semi-automatically generating star schema(s) from an ERD. The novel features of SAMSTAR are that we use Connection Topology Value to identify facts that have the highest number of $M:1$ relationships and then identify dimensions by using the topology of the input ERD as well as Annotated Dimensional Design Patterns and WordNet.

We illustrated the SAMSTAR algorithm with examples. It was observed that our algorithm is significant in the following four aspects: (1) It is a universal method to generate star schema(s) in that we have used generalized DDPs and WordNet to identify dimensions of a fact table. (2) It is quantitative in nature in that we analyze the structure of an input ERD. (3) It can be used to identify a set of fact candidates from a large and complex ERD. (4) This method could be automated up to a large extent and thus simplifies the work of experienced designers and gives a smooth head-start to novices.

A future research direction is to identify the appropriate relationship between the value of $k$ (adjustable parameter) and the number of fact tables and the size of the ERD. SAMSTAR could be extended to identify the measures of facts and the attributes of dimensions. Also, it could be further refined to verify that the grains of the fact table and each of the dimensions tables are
compatible to each other, and hence, could reassure that the query results are valid.

7. REFERENCES


