

Paper #579 Generating Preview Tables for Entity Graphs

Dear Chairs and Reviewers,

We thank you all for offering us the opportunity to revise the paper and respond to your comments. We would like to thank all reviewers and the meta reviewer for their detailed and insightful comments, which helped us to substantially improve the paper. In this letter, we first summarize the revisions we have made to the paper. We then provide more detailed responses to the reviewers' individual comments. We use notations [RxAy], [RxWy] and [RxDy] to denote a reviewer's requested revisions, weak points, and detailed comments, respectively. In the reviews the three reviewers were named as "Reviewer_1", "Reviewer_3", and "Reviewer_5". We will use the same.

(1) Summary of Changes

1.1 We conducted an extensive user study using 84 recruited students at our institution. In the user study each participant was requested to use one of seven alternative systems, answer several test questions, as well as provide feedback to a user experience questionnaire. The seven systems were compared based on the user study results. The new Section 6.3 explains the user study setup and results in detail. The related new figures/tables are Table 5, Table 6, Table 7, and Figure 10.

1.2 We also organized a panel for a group of experts (graduate students doing research in related areas) to manually produce preview tables. The panel participants individually worked on the task and then discussed together to come up their preview tables for several domains of the Freebase dataset. The outcome becomes one of the seven systems that were compared in user study. Section 6.3 also discusses how the expert panel worked.

1.3 In the original manuscript, for the results of comparison with gold standard on key attribute scoring (Section 6.1.2), we presented Figure 5 which uses measure Precision-at-K. We now include Figures 6 and 7 as well, which use Average Precision and nDCG. The two measures are also briefly explained in the middle of Section 6.1.2.

1.4 We expanded Section 7 Related Work to discuss a few more related studies, most of which were suggested by the reviewers. The new materials are in the last two paragraphs of this section. Note that more directly relevant studies on schema summarization were discussed in Section 1 (page 2) of the manuscript.

1.5 We also extended the Conclusion section to discuss a few directions of future work.

1.6 We included in the Appendix details of the gold standard collected from Freebase (Table 12), as well as the results of comparing the Freebase gold standard with the hand-crafted preview tables produced by experts (Tables 10 and 11).

1.7 We moved the original Section 6.3 "Sample Optimal Previews" to Appendix, to comply with the space limitation.

(2) Detailed Responses

Assigned Reviewer_1 (R1)

R1A1: The focus of this paper could be improved significantly if the authors focused on the qualitative aspects and the human curation aspects instead of the performance experiments. Conceptually, the task is not

time-intensive since once the preview is generated, it won't change frequently even there is more new edits coming in considering the size of entity graph.

The current user study is not enough to tell how this will indeed help users to make a decision to select the entity graph. It is only used to evaluate the ranking. The paper needs to show that the technique provided is a good qualitative alternative to the baseline / naive case.

R1W2: The paper could do with more comparisons against reasonable baselines / comparable algorithms; the evaluations (Fig 6 & 7) simply compare the two proposed algorithms, but do not compare this against a reasonable existing alternative.

Response to R1A1 and R1W2: We conducted an extensive user study using 84 recruited students to compare the qualitative aspects of seven approaches, including a set of preview tables coming out of manual investigation and discussion of an expert panel. For the new materials, please refer to the new Section 6.3, Table 5, Table 6, Table 7, and Figure 10.

R1W1: While the paper is extremely well thought out, the overall motivation for this problem needs to be fleshed out more. While this problem itself may be NP-hard, the authors do not articulate how common this problem is, or what the overall impact is. As the authors mention, "Freebase offers an entrance page showing 6 major entity domains" then perhaps first demonstrate that intuitive alternatives do not suffice, such as Pagerank representation of the graph (ObjectRank, Balmin @ VLDB 2004) and schema summarization etc? Alternatively, perhaps show that this method is *faster*(i.e. time-to-task* than human curation?

Response to R1W1:

The new user study (Section 6.3) compares the proposed preview tables with several alternative approaches, including using schema graph itself, schema summarization, Freebase ground truth, as well as the preview tables manually produced by human experts.

It is indeed verified that producing preview tables using human experts is a very time-consuming process, and thus an automatic algorithm is important. The details are described in Section 6.3.

We extended Section 7 Related Work to discuss a few related studies, including ObjectRank. ObjectRank applies authority-based ranking to database keyword search. Part of its ranking formula is extended from PageRank. The table importance measure in YPS09 [19] and our random-walk based scoring measure in Section 3.2 bear similar ideas. (This was mentioned at the end of Section 3.2 and the newly added 2nd-to-last paragraph of Section 7.) Note that the ranking measures are not the focus of this paper, and the results on the optimization problems in section 4 and the algorithms in Section 5 are not affected by the specific ranking measures. (This was mentioned in the last paragraph of Section 3.1.)

R1D1: The details of the user study need to be fleshed out. For example, what does it mean to say "important" in "The workers were asked which of the 2 entity types in the pair is more important."; all responses are valid (no outliers) ?, etc.

Response to R1D1: When we gave pairs of entity types to each Amazon Mechanical Turk worker, we directly asked them to, for each pair, "choose the one that is more important than the other". We also provided a few examples to explain what are considered more important in common sense. The workers were asked to answer a few screening questions that test their common knowledge. They must answer the screening questions correctly, otherwise their responses were not considered. These details are now included in Section 6.2.

R1D2 : Other papers to look at, esp. in the context of search:

* "Qunits: queried units in database search", Nandi and Jagadish CIDR 2009

* "Query Biased Snippet Generation in XML Search", Chen et al SIGMOD 2009

Response to R1D2: We extended Section 7 Related Work to discuss these and other studies.

Assigned Reviewer 3 (R3)

R3A1: The user evaluation has to be more convincing. Specially, it would make sense to quantitatively compare the preview tables computed by the algorithms against hand-crafted schemas created by a set of experts.

R3W2: User study and evaluation are weaker in comparison to the rest of technical presentation.

Response to R3A1 and R3W2: We conducted an extensive user study using 84 recruited students to compare seven alternative approaches, including hand-crafted preview tables by experts. For the new materials, please refer to the new Section 6.3, Table 5, Table 6, Table 7, and Figure 10.

R3A2: The differences between the variations in criterion (concise + diverse + dense) would ideally also be evaluated.

R3D7: Though the paper emphasizes the ability to obtain multiple variations of a preview schema (concise, concise + dense, and concise + diverse), the experimental section does not address the objective / subjective quality of the previews computed over the variations (except for the example in Table 6).

Response to R3A2 and R3D7: These variations are also compared in the new user study in Section 6.3.

R3D1: It is hard to appreciate the quality comparison, when the Freebase schemas used are unknown (as the author suggests, the published pages are no longer accessible). At the very least the authors should list the entity types and the non-key attributes for a few of the tested domains.

Response to R3D1: The schema of the tables in the Freebase gold standard can be found in Table 12 in the Appendix.

R3D2: The authors assume that the six entity types and their attributes as published by Freebase can be used a golden set. Is there a justification for this? Freebase does not necessarily have the preview goals laid out in the paper. It would be more convincing if the paper included a evaluation in which the Freebase schema was also rated in the user study and then the overlap measure with the solutions generated by the different algorithms.

Response to R3D2: We used precision-at-k to compare freebase gold standard against experts-produced preview tables. The results are presented in Tables 10 and 11 in Appendix.

R3D3: The entity type importance correlation (Section 6.1.3) does not appear to necessarily relevant to the author's preview table goals. There has to be justification for why that comparison is a reasonable one.

Response to R3D3: To directly evaluate the preview tables, we have conducted a user study using 84 participants. Please refer to Section 6.3.

R3D4: Why is Precision@K a good measure for preview quality? At least an informal justification is necessary. Further, the charts in Figure 5 are hard to read and understand.

Response to R3D4: Precision-at-K, Average Precision, and nDCG are widely used for measuring accuracy of ranked results. The intuition is that an approach that ranks accurate results higher is expected to receive better values under these measures. We have added this explanation to Section 6.1.2.

We have also improved the aspect ratio of Figure 5. Hopefully it is now easier to read and understand.

R3D6: An alternate evaluation strategy might be to have a set of 10 experts (e.g., grad students with some data management understanding) independently create preview tables for the said datasets under the outlined design constraints (k tables, n attributes). These can be instead compared with the results of the algorithm (and correlation/overlap measured).

Response to R3D6: Thank you very much for the suggestion. We followed this suggestion to produce hand-crafted preview tables by experts, and such hand-crafted result was one of the seven approaches in the user study. Please refer to the new Section 6.3 for details and results.

Assigned_Reviewer_5 (R5)

R5A1: In the experiments, compare the usefulness of the preview tables vs schema graph (or summary of the schema graph). This can be done by a user study among data workers (and not AMT users).

R5W1: In the experiments, there should be a direct comparison between the schema graph (or summary of the schema graph) and the preview tables from the usefulness and understandability of the users (i.e., data workers) point of view.

R5D1: The problem studied in this paper looks interesting. The concept of preview tables are modeled mathematically and the right parameters to control the size of the preview tables are introduced. However, the motivation of using preview tables is not supported in the experiment.

R5D2: The contribution of this paper is that it claims providing preview tables is more useful than providing schema graph (or summary of the schema graph) for the purpose of understanding the entity graph. However, the authors fail to show this in the experiments. In the experiments, the accuracy of the solution is considered only by looking at the ranking orders of candidate keys. Even in the user study, users are asked to compare which pair of the 2 entity types are more important. However, the right experiment would show how useful the schema graph (or summary of the schema graph) is versus the preview tables. Instead of asking the users of AMT, actual data workers should be asked to evaluate the usefulness of the preview tables.

Response to R5A1, R5W1, R5D1, R5D2: Thank you very much for the suggestion. We conducted an extensive user study using 84 recruited students to compare seven alternative approaches, including schema graph and summarization of schema graph. For the new materials, please refer to the new Section 6.3, Table 5, Table 6, Table 7, and Figure 10.

R5A2: In the experiments, a sample of the schema graph summary should also be shown beside each preview table sample.

Response to R5A2: Since we studied 7 different approaches in the user study, we couldn't find enough participants and allocate enough participant stipend for all pairwise side-by-side comparisons. Hence, we used A/B testing in Section 6.3 to compare these 7 approaches.

R4A3: The related work should be expanded. More discussions about similar work that has been done in the past should be added.

R5A4: Since the concept of the preview tables for entity graphs are introduced for the first time in this paper, the conclusion should include a thorough discussion about the possibilities of future work.

R5W3: The related work section is too short and shallow. The conclusion should include the possible future work.

Response to R5A3, R5A4, R5W3: We expanded Section 7 Related Work to discuss a few more related studies, most of which were suggested by the reviewers. The new materials are in the last two paragraphs of this section. Note that more directly relevant studies on schema summarization were discussed in Section 1 (page 2) of the manuscript.

We also extended the Conclusion section to discuss several directions of possible future work.

Generating Preview Tables for Entity Graphs

ABSTRACT

Users are tapping into massive, heterogeneous entity graphs for many applications. It is challenging to select entity graphs for a particular need, given abundant datasets from many sources and the oftentimes scarce information for them. We propose methods to produce preview tables for compact presentation of important entity types and relationships in entity graphs. The preview tables assist users in attaining a quick and rough preview of the data. They can be shown in a limited display space for a user to browse and explore, before she decides to spend time and resources to fetch and investigate the complete dataset. We formulate several optimization problems that look for previews with the highest scores according to intuitive goodness measures, under various constraints on preview size and distance between preview tables. The optimization problem under distance constraint is **NP-hard**. We design a dynamic-programming algorithm and an Apriori-style algorithm for finding optimal previews. Results from experiments, comparison with related work and user studies demonstrated the scoring measures’ accuracy and the discovery algorithms’ efficiency.

1. INTRODUCTION

We witness an unprecedented proliferation of massive, heterogeneous *entity graphs* that represent entities and their relationships in many domains. For instance, in Fig. 1—a tiny excerpt of an entity graph, the edge labeled *Actor* between nodes Will Smith and Men in Black captures the fact that the person is an actor in the film. Real-world entity graphs include knowledge bases (e.g., DBpedia [2], YAGO [16], Probase [18], Freebase [4] and Google’s Knowledge Vault [8]), social graphs, biomedical databases, and program analysis graphs, to name just a few. Numerous applications are tapping into entity graphs in domains such as search, recommendation systems, business intelligence and health informatics.

Entity graphs are often represented as RDF triples, due to heterogeneity of entities and the often lacking schema. The Linking Open Data community has interlinked billions of RDF triples spanning over several hundred datasets.¹ Many other entity graph datasets are also available from data repositories such as Amazon’s Public

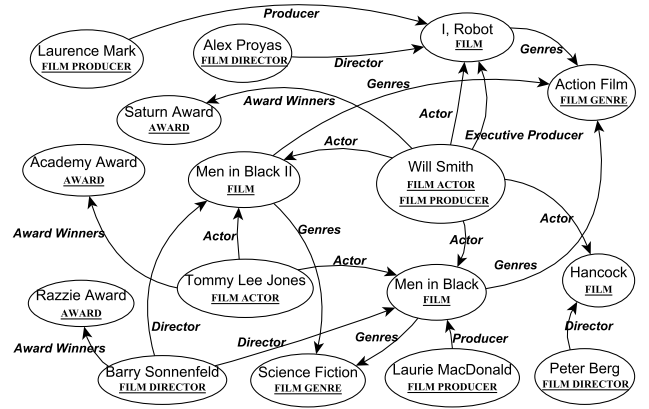


Figure 1: An Excerpt of an Entity Graph.

Data Sets,² Data.gov³ and NCBI’s databases.⁴

It is challenging to select entity graphs for a particular need, given abundant datasets from many sources and oftentimes scarce information available about them. While sources such as the aforementioned data repositories often provide dataset descriptions, one cannot get a direct look at an entity graph before fetching it. Downloading a dataset and loading it into a database can be a daunting task. A data worker may need to tackle many challenges before they can start any real work on an entity graph.

In this paper, we propose methods to automatically produce *preview tables* for entity graphs. Given an entity graph with a large number of entities and relationships, our methods select from the many entity types a few important ones and produce a table for each chosen entity type. Such a table comprises a set of attributes, selected among many candidates, each of which corresponds to a relationship associated with the corresponding entity type. A tuple in the table consists of an entity belonging to the entity type and its related entities for the table attributes.

Fig. 2 is a possible preview of the entity graph in Fig. 1. It consists of two preview tables—the upper table has attributes FILM, Director and Genres, and the lower table has attributes FILM ACTOR and Award Winners. In this preview, entities of types FILM and FILM ACTOR are deemed of central importance in the entity graph. Hence, FILM and FILM ACTOR are the *key attributes* of the two tables, respectively, marked by the underlines beneath them. Attributes *Director* and *Genres* in the upper table are considered highly related to FILM entities. Similarly, *Award Winners* in the lower table is highly related to FILM ACTOR entities. The two tables contain 4 and 2

¹ <http://linkeddata.org/>

² <http://aws.amazon.com/publicdatasets/>

³ <http://www.data.gov/>

⁴ <http://www.ncbi.nlm.nih.gov/>

	FILM	Director	Genres
t_1	Men in Black	Barry Sonnenfeld	{Action Film, Science Fiction}
t_2	Men in Black II	Barry Sonnenfeld	{Action Film, Science Fiction}
t_3	Hancock	Peter Berg	-
t_4	I, Robot	Alex Proyas	{Action Film}

	FILM ACTOR	Award Winners
t_5	Will Smith	Saturn Award
t_6	Tommy Lee Jones	Academy Award

Figure 2: A 2-Table Preview of the Entity Graph in Fig. 1. (Upper and lower tables for subgraphs #1 and #2 in Fig. 3, respectively.)

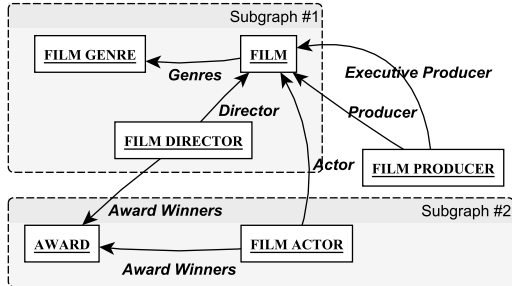


Figure 3: The Schema Graph for the Entity Graph in Fig. 1.

tuples, respectively. For instance, the first tuple of the upper table is $t_1 = \langle \text{Men in Black}, \text{Barry Sonnenfeld}, \{\text{Action Film}, \text{Science Fiction}\} \rangle$. The tuple indicates that entity Men in Black belongs to type FILM and has a relationship Director from Barry Sonnenfeld and has relationship Genres to both Action Film and Science Fiction.

Data workers browse and explore data under inevitable display space constraints on mobile devices and desktop monitors. The proposed preview tables are for compact presentation of important types of entities and their relationships in an entity graph. They assist data workers in attaining a quick and rough preview of the schema of the data, before they decide to spend more time, money and resources to fetch and investigate the complete entity graph. The tuples in the tables further facilitate an intuitive understanding of the data. (Our approach shows a few randomly sampled tuples in each preview table. How to selectively display important tuples is left to future study.)

To this end, two other approaches are arguably less adequate for gaining a quick overview of an entity graph.

(1) One solution is to show a schema graph corresponding to the data graph. Fig. 3 is the schema graph for the entity graph in Fig. 1. While its definition is given in Sec. 2, we note that it is generated by merging same-type entity graph vertices (i.e., entities) and edges (i.e., relationships). Although a schema graph is much smaller than the corresponding entity graph, it is not small enough for easy presentation and quick preview. For instance, in a snapshot of the “film” domain of Freebase, there are 190K vertices and 1.6M edges. The corresponding schema graph consists of 50 entity types and 136 relationship types.

(2) Another approach is to present a summary of the schema graph, by schema summarization techniques [19, 20, 21, 17, 22]. Some of these methods [19, 20, 21] work on relational and semi-structured data, instead of graph data. Some [21, 17, 22] produce trees or graphs as output instead of flat tables. It is unclear how to apply these methods on an entity graph or its schema graph, due to differences in data models. Although it is plausible that some of these approaches can be adapted for entity graphs, several reasons can render them ineffective. First, schema summary can still be quite large. The most closely related work, [19, 20], clusters the tables in a database but does not reduce the number of tables or the

complexity of database schema. If we treat each entity type as a table and its neighboring entity types in the schema graph as the table attributes, the number of tables would equal the number of entity types. For the aforementioned “film” domain in Freebase, it means one would have to understand the result of clustering 50 tables. Second, schema summarization is for helping database administrators and programmers in gaining a detailed understanding of a database in order to form queries. Our goal is to assist data workers in attaining a quick and rough understanding of an entity graph, before they decide to grasp such a detailed understanding. Therefore, our work can be viewed as an approach of finding a succinct representation of the schema graph (instead of clustering it). We are not aware of such an approach in previous studies.

In our definition (details in Sec. 2), a *preview* is a set of preview tables, each of which has a *key attribute* (corresponding to an entity type) and a set of *non-key attributes* (each corresponding to a relationship type). Given an entity graph and its schema graph, there is thus a large space of possible previews. Our goal is to find an “optimal” preview in the space. To this end, we tackle several challenges: (1) We discern what factors contribute to the goodness of a preview and propose several scoring functions for key and non-key attributes as well as preview tables. The scoring functions are based on several intuitions related to how much information a preview conveys and how helpful it is to users. (2) Based on the scoring measures, a preview’s score is maximized when it includes as many tables and attributes as possible. However, the purpose of having a preview is to help users attain a quick understanding of data and thus a preview must fit into a limited display space. Considering the tradeoff, we enforce a constraint on preview size. Furthermore, we consider enforcing an additional constraint on the pairwise distance between preview tables. Given the spaces of all possible previews, we formulate the optimization problem of finding an preview with the highest score among those satisfying the constraints. The optimization is non-trivial, as we prove that it is NP-hard under distance constraint. (3) The search space of previews grows exponentially by data size and the constraints. A brute-force approach is thus too costly. For efficiently finding optimal previews, we designed a dynamic programming algorithm and an Apriori [1]-style algorithm.

In summary, this paper makes the following contributions:

- We motivated a novel concept of preview for entity graphs.
- We proposed ideas for measuring the goodness of previews based on several intuitions. (Sec. 3)
- We formulated optimal preview discovery problem, and proved its NP-hardness under distance constraint. (Sec. 4)
- We developed a dynamic-programming algorithm and an Apriori-style algorithm for finding optimal previews. (Sec. 5)
- Extensive experiments, comparison with related work, and user study verified the scoring measures’ accuracy, the algorithms’ efficiency, and the effectiveness of discovered previews. (Sec. 6)

2. PREVIEW DISCOVERY PROBLEM

An *entity graph* is a directed multigraph $G_d(V_d, E_d)$ with vertex set V_d and edge set E_d . Each vertex $v \in V_d$ represents an entity and each edge $e(v, v') \in E_d$ represents a directed relationship from entity v to v' . G_d is a multigraph since there can be multiple edges between two vertices. (E.g., in Fig. 1, there are two edges Actor and Executive Producer from entity Will Smith to entity I, Robot.)

Each entity is labeled by a name. For simplicity and intuitiveness of presentation, we shall mention entities by their names, assuming all entities have distinct names, although in reality they are distinguished by unique identifiers such as URIs. Each entity belongs

$G_d(V_d, E_d)$	an entity graph
$v \in V_d$	an entity
$e(v, v') \in E_d$	a directed relationship from entity v to entity v'
$G_s(V_s, E_s)$	a schema graph
$\tau \in V_s$	an entity type
$\gamma(\tau, \tau') \in E_s$	a relationship type from entity type τ to entity type τ'
T	a preview table
$T.key$	the key attribute of T
$T.nonkey$	the non-key attributes of T
$T.\tau$	the set of entities of type τ —the key attribute of T
$t \in T$	a tuple t in preview table T
$t.\tau$	t 's value on τ which is the key attribute of T
$t.\gamma$	t 's value on non-key attribute γ
$\mathcal{P} = \{\mathcal{P}[1], \dots, \mathcal{P}[k]\}$	a preview, which consists of k preview tables
\mathcal{P}_{opt}	an optimal preview
$S(\mathcal{P})$	the score of preview \mathcal{P}
$S(T)$	the score of preview table T
$S_{cov}(\tau), S_{walk}(\tau)$	score of key attribute τ based on coverage/random-walk
$S_{cov}^r(\gamma), S_{ent}^r(\gamma)$	score of non-key attribute γ based on coverage/entropy
\mathbb{T}	the space of all possible preview tables
\mathbb{P}	the space of all possible previews
$dist(\tau, \tau')$	distance between τ and τ' in schema graph G_s

Table 1: Notations

to one or more *entity types*, underlined in Fig. 1. (E.g., Will Smith belongs to types FILM ACTOR and FILE PRODUCER and I, Robot belongs to type FILM.) Each relationship belongs to a *relationship type*. (E.g., the edge from Will Smith to Men in Black has type Actor.) The type of a relationship determines the types of its two end entities. For instance, an edge of type Actor is always from an entity belonging to FILM ACTOR to an entity belonging to FILM. We will mention edges by the surface names of their relationship types. Two different relationship types may have the same surface name for intuitively expressing their meanings, although underlyingly they have different identifiers. For instance, the *Award Winners* edge from Will Smith to Saturn Award and the *Award Winners* edge from Barry Sonnenfeld to Razzie Award belong to two different relationship types. The former is for relationships from FILM ACTOR to AWARD, while the latter is for relationships from FILM DIRECTOR to AWARD.

Given an entity graph $G_d(V_d, E_d)$, its *schema graph* is a directed graph $G_s(V_s, E_s)$, where each vertex $\tau \in V_s$ represents an entity type and each directed edge $\gamma(\tau, \tau') \in E_s$ represents a relationship type from entity type τ to τ' . An edge $\gamma(\tau, \tau') \in E_s$ if and only if there exists an edge $e(v, v') \in E_d$ where e has type γ , v has type τ and v' has type τ' . Fig. 3 shows the schema graph corresponding to the entity graph in Fig. 1. A schema graph is a multigraph as there can be multiple relationship types between two entity types. (E.g., two relationship types—*Producer* and *Executive Producer*—are from entity type FILM PRODUCER to FILM.) It is clear from the above definitions that, given a data graph, the corresponding schema graph is uniquely determined.

Definition 1 (Preview Table and Preview). Given an entity graph $G_d(V_d, E_d)$ and its schema graph $G_s(V_s, E_s)$, a *preview table* T has a mandatory *key attribute* (denoted $T.key$) and at least one *non-key attributes* (denoted $T.nonkey$). T corresponds to a star-shape subgraph of the schema graph $G_s(V_s, E_s)$. The key attribute corresponds to an entity type $\tau \in V_s$, and each non-key attribute corresponds to a relationship type $\gamma(\tau, \tau') \in E_s$ or $\gamma(\tau', \tau) \in E_s$. Note that the edges from and to an entity are both important. Hence, a non-key attribute corresponds to either $\gamma(\tau, \tau')$ or $\gamma(\tau', \tau)$.

The preview table T consists of a set of tuples. The number of tuples equals the number of entities of type τ (the key attribute of T), i.e., $|T| = |T.\tau|$ and $T.\tau = \{v | v \in V_d \wedge v \text{ has type } \tau\}$. Given an arbitrary tuple $t \in T$, we denote t 's key attribute value by $t.\tau$. Each tuple t attains a distinct value of $t.\tau$. Its value on a non-key attribute $\gamma(\tau, \tau')$, denoted $t.\gamma(\tau, \tau')$ or simply $t.\gamma$, is a set—the set of entities in entity graph G_d incident from $t.\tau$ through an edge of

type $\gamma(\tau, \tau')$. More formally, $t.\gamma(\tau, \tau') = \{u | u \in V_d \wedge e(t.\tau, u) \in E_d \wedge u \text{ belongs to type } \tau'\}$. Symmetrically, its value on a non-key attribute $\gamma(\tau', \tau)$ is the set of entities in G_d incident to $t.\tau$ through an edge of type $\gamma(\tau', \tau)$, i.e., $t.\gamma(\tau', \tau) = \{u | u \in V_d \wedge e(u, t.\tau) \in E_d \wedge u \text{ belongs to type } \tau'\}$.

A *preview* \mathcal{P} is a set of preview tables, i.e., $\mathcal{P} = \{\mathcal{P}[1], \dots, \mathcal{P}[k]\}$, where $\forall i \neq j, \mathcal{P}[i].key \neq \mathcal{P}[j].key, k \leq |V_s|$ is the total number of preview tables. Note that $|V_s|$ is the number of vertices in G_s , i.e., the number of entity types in G_d . \square

According to Definition 1, the upper and lower tables in Fig. 2 correspond to the star-shape subgraphs #1 and #2 in Fig. 3, respectively. The key attribute in the upper table is FILM and the non-key attributes are *Director* and *Genres*. The key attribute in the lower table is FILM ACTOR and its non-key attribute is *Award Winners*. It is worth noting that, although each tuple's value on the key attribute is non-empty, unique and single-valued, its value on a non-key attribute can be empty (e.g., $t_3.Genres$ in Fig. 2), duplicate (e.g., $t_1.Director$ and $t_2.Director$ in Fig. 2) and multi-valued (e.g., $t_1.Genres$ and $t_2.Genres$ in Fig. 2). It also follows that a preview table is not a relational table.

By Definition 1, every vertex τ in a schema graph can serve as the key attribute of a candidate preview table, which also includes at least one non-key attribute—an edge incident on τ . We use \mathbb{T} to denote the space of all possible preview tables. A preview is a set of preview tables. We use \mathbb{P} to denote the space of all possible previews. Note that $\mathbb{P} \subset 2^{\mathbb{T}}$, i.e., not every member of the power set $2^{\mathbb{T}}$ is a valid preview, because by Definition 1 preview tables in a preview cannot have the same key attribute.

Problem Statement: Given an entity graph $G_d(V_d, E_d)$ and its corresponding schema graph $G_s(V_s, E_s)$, the *preview discovery problem* is to find \mathcal{P}_{opt} —the optimal preview among all possible previews. We shall develop the notions of goodness and optimality for a preview and define goodness measures in Sec. 3.

Note that the preview discovery problem focuses on selecting key and non-key attributes for preview tables. It does not select tuples. As our goal is to help users attain a good initial understanding of the schema of an entity graph, we argue that it is only necessary to show a small number of tuples instead of all. Our current approach is to randomly select a few. How to define representativeness of tuples and choose the most representative tuples is left for future work.

3. SCORING MEASURES FOR PREVIEWS

In this section, we discuss the scoring functions for measuring the goodness of previews for entity graphs. The measures are based on the intuition that a good preview should 1) relate to as many entities and relationships as possible and 2) help users understand an entity graph and its schema graph. The first intuition is obvious, as a preview relating to only a small number of entities or relationships will inevitably lose lots of information and thus lead to poor comprehensibility of the original graph. The second intuition models the goodness of previews according to users' behavior in browsing entity and schema graphs.

3.1 Preview Scoring

The score of a preview $\mathcal{P} = \{\mathcal{P}[1], \dots, \mathcal{P}[k]\}$ is simply aggregated from individual preview tables' scores, by summation:

$$S(\mathcal{P}) = \sum_{i=1}^k S(\mathcal{P}[i]), \quad (1)$$

where $S(\mathcal{P}[i])$ is the score of a preview table $\mathcal{P}[i]$, defined as:

$$S(\mathcal{P}[i]) = S(\tau) \times \sum_{\gamma \in \mathcal{P}[i].nonkey} S^r(\gamma), \quad (2)$$

where $S(\tau)$ is the score of the key attribute of $\mathcal{P}[i]$ (i.e., $\mathcal{P}[i].key=\tau$) and $S^\tau(\gamma)$ is the score of a non-key attribute γ in $\mathcal{P}[i]$. $S(\tau)$ and $S^\tau(\gamma)$ are defined and elaborated in Sec. 3.2 and Sec. 3.3.

In the above definition, the score of a preview table equals the product of its key attribute's score and the summation of its non-key attributes' scores. The definition gives the key attribute τ much higher importance than any individual non-key attribute, because the preview table centers around the entities of type τ and describes their non-key attributes, i.e., their relationships with other entities.

It is possible to propose many viable scoring functions for previews, key attributes and non-key attributes. Furthermore, techniques such as learning-to-rank [12] may be applied in ranking previews by features related to key and non-key attributes, although the feasibility of collecting many labelled data is less clear in this case. We leave it to future work to explore this direction. Nevertheless, we note that the results on the optimization problems in Section 4 and the algorithms in Section 5 will stand, as long as the scoring function replacing Eq. 1 and Eq. 2 is monotonic with regard to $S(\tau)$ and $S^\tau(\gamma)$, and the measures defining $S(\tau)$ and $S^\tau(\gamma)$ do not affect the results.

3.2 Key Attribute Scoring

Coverage-based scoring measure: Given an entity graph $G_d(V_d, E_d)$ and its corresponding schema graph $G_s(V_s, E_s)$, the key attribute τ of a candidate preview table T corresponds to an entity type, i.e., $\tau \in V_s$. If the entity graph consists of many entities of type τ , including T in the preview makes the preview relevant to all those entities. The coverage-based scoring measure thus defines the score of τ as the number of entities bearing that type:

$$S_{cov}(\tau) = |\{v|v \in V_d \wedge v \text{ has type } \tau\}|$$

For example, given the entity graph in Fig. 1 and the corresponding schema graph in Fig. 3, the coverage-based score of the key attribute FILM is $S_{cov}(\text{FILM}) = 4$.

Random-walk based scoring measure: We consider a *random-walk process* over a graph G converted from the schema graph $G_s(V_s, E_s)$, inspired by the PageRank algorithm [5] for Web page ranking and many related ideas. Similar to G_s , vertices in G are entity types and edges are relationship types. Different from G_s , the edges are undirected. As explained in Def. 1, the edges from and to an entity are both important to the entity. The edge between τ_i and τ_j in G is weighted by the number of relationships (i.e., the number of edges) in the entity graph between entities of types τ_i and τ_j . We denote the weight by w_{ij} , defined as follows.

$$w_{ij} = w_{ji} = \sum_{\gamma(\tau_i, \tau_j) \in E_s} |\{e|e \in E_d \wedge e \text{ has type } \gamma(\tau_i, \tau_j)\}| \\ + \sum_{\gamma(\tau_j, \tau_i) \in E_s} |\{e|e \in E_d \wedge e \text{ has type } \gamma(\tau_j, \tau_i)\}|$$

In the $|V_s| \times |V_s|$ transition matrix M , an element M_{ij} corresponds to the *transition probability* from τ_i to τ_j in G . M_{ij} equals the ratio of w_{ij} to the total weight of all edges incident on τ_i in G :

$$M_{ij} = w_{ij} / \sum_k w_{ik}$$

For example, based on Fig. 3, the transition probability from FILM to FILM GENRE is $M_{\text{FILM}, \text{FILM GENRE}} = w_{\text{FILM}, \text{FILM GENRE}} / (w_{\text{FILM}, \text{FILM GENRE}} + w_{\text{FILM}, \text{FILM ACTOR}} + w_{\text{FILM}, \text{FILM DIRECTOR}} + w_{\text{FILM}, \text{FILM PRODUCER}}) = 5/(5+6+4+3) = 0.28$. The transition probability from FILM to FILM PRODUCER is $M_{\text{FILM}, \text{FILM PRODUCER}} = w_{\text{FILM}, \text{FILM PRODUCER}} / (w_{\text{FILM}, \text{FILM GENRE}} + w_{\text{FILM}, \text{FILM ACTOR}} + w_{\text{FILM}, \text{FILM DIRECTOR}} + w_{\text{FILM}, \text{FILM PRODUCER}}) = 3/(5+6+4+3) = 0.17$.

Suppose a random walker traverses in G , either by going from an entity type τ_i to another entity type τ_j through the edge between

them with probability M_{ij} or by jumping to a random entity type. Entity types that are more likely to be visited by the user are of higher importance. The random walk process will converge to a stationary distribution which represents the chances of entity types being visited. The stationary distribution π of the random walk process is given as follows. Note that a similar idea was applied in [19] for ranking relational tables by importance.

$$\pi = \pi M$$

The random-walk based score of a candidate key attribute τ_i is:

$$S_{walk}(\tau_i) = \pi_i, \text{ where } \pi_i \text{ is the stationary probability of } \tau_i.$$

3.3 Non-Key Attribute Scoring

Coverage-based scoring measure: The coverage-based scoring measure for non-key attribute is similar to that for key attribute. Given an entity graph $G_d(V_d, E_d)$ and its schema graph $G_s(V_s, E_s)$, consider a candidate preview table T with key attribute τ . A non-key attribute γ of T corresponds to a relationship type, i.e., $\gamma \in E_s$. If the entity graph contains many edges (i.e., relationships) belonging to type γ , incorporating such a relationship type into table T makes it relevant to all those relationships and their corresponding entities. The coverage-based scoring measure thus defines the score of γ as the number of relationships bearing that type:

$$S_{cov}^\tau(\gamma) = |\{e|e \in E_d \wedge e \text{ has type } \gamma\}|$$

For example, given the entity graph in Fig. 1 and the schema graph in Fig. 3, the coverage-based scores of non-key attributes *Director* and *Genres* are $S_{cov}^{\text{FILM}}(\text{Director}) = 4$ and $S_{cov}^{\text{FILM}}(\text{Genres}) = 5$.

The coverage-based scoring measure for non-key attribute is symmetric, i.e., given $\gamma(\tau, \tau')$ (or $\gamma(\tau', \tau)$) $\in T.nonkey$, $S_{cov}^\tau(\gamma) \equiv S_{cov}^{\tau'}(\gamma)$. Both τ and τ' can be the key attribute of a different preview table, in which γ is a non-key attribute. The scores of γ in the two tables are equal.

Entropy-based scoring measure: For a preview table T with key attribute τ , we measure the goodness of a non-key attribute $\gamma(\tau, \tau')$ (or $\gamma(\tau', \tau)$) by how much information it provides to T , for which the *entropy* of γ ($H(\gamma)$) is a natural choice of measure:

$$S_{ent}^\tau(\gamma) = H(\gamma) = \sum_{j=1}^{n_j} \frac{n_j}{|t.\gamma|} \log\left(\frac{|t.\gamma|}{n_j}\right),$$

where n_j is the number of tuples in T that attain the same j th attribute value u on non-key attribute $\gamma(\tau, \tau')$ (or $\gamma(\tau', \tau)$), i.e., $u \in V_d \wedge u \text{ has type } \tau'$ and $n_j = |\{v|v \in T.\tau \wedge e(v, u) \in E_d \text{ (or } e(u, v) \in E_d) \wedge e \text{ has type } \gamma\}|$. $|t.\gamma|$ is the number of tuples in T with non-empty values on $\gamma(\tau, \tau')$ (or $\gamma(\tau', \tau)$). Continue the running example. The entropy-based scores of non-key attributes *Director* and *Genres* are $S_{ent}^{\text{FILM}}(\text{Director}) = (2/4) \log(4/2) + (1/4) \log(4/1) + (1/4) \log(4/1) = 0.45$, and $S_{ent}^{\text{FILM}}(\text{Genres}) = (2/3) \log(3/2) + (1/3) \log(3/1) = 0.28$. Note that for two values on a multi-valued attribute (e.g., {Action Film, Science Fiction} and {Action Film} for FILM.Genres in Fig. 2), we consider them equivalent if and only if they have the same set of component values. By definition, the entropy-based scoring measure for non-key attribute is asymmetric, i.e., given $\gamma(\tau, \tau')$ (or $\gamma(\tau', \tau)$) $\in T.nonkey$, $S_{ent}^\tau(\gamma) \neq S_{ent}^{\tau'}(\gamma)$.

4. OPTIMAL PREVIEWS UNDER SIZE AND DISTANCE CONSTRAINTS

In this section, based on the scoring measures defined in Sec. 3, we formulate several optimization problems that look for the optimal previews with best scores under various constraints on preview size and distance between preview tables. We prove that some of these optimization problems are NP-hard.

By Eq. 1 (or any other monotonic aggregate function), the score of a preview monotonically increases by its member preview tables—the more preview tables in a preview, the higher its score. Similarly by Eq. 2, the score of a preview table monotonically increases by its non-key attributes. The properties are formally stated in the following two propositions. Recall that \mathbb{P} and \mathbb{T} denote the space of all possible previews and all possible preview tables.

Proposition 1. Given previews $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{P}$, if $\mathcal{P}_1 \supseteq \mathcal{P}_2$, then $S(\mathcal{P}_1) \geq S(\mathcal{P}_2)$.

Proposition 2. Given preview tables $T_1, T_2 \in \mathbb{T}$, if $T_1.key = T_2.key$ and $T_1.nonkey \supseteq T_2.nonkey$, then $S(T_1) \geq S(T_2)$.

By the above propositions, a preview’s score is maximized when it includes as many tables and attributes as possible. However, a preview must fit into a limited display space, due to constraints posed by mobile devices and desktop monitors. Therefore the size and the goodness score of a preview present a tradeoff. Considering the tradeoff, we enforce a constraint on preview size, given by a pair of integers (k, n) , where k is the number of allowed preview tables and n is the number of allowed non-key attributes in the tables. Their values may be either manually chosen by interactive users or automatically suggested based on the size of a display space. The previews satisfying the size constraint are called *concise previews*.

An alternative size constraint is a maximally allowed number of attributes per preview table. However, we do not consider such a constraint in this paper. We argue that forcing each preview table to have the same width can cause two problems—on the one hand, the allocated space for some preview tables may be wasted because they do not have as many important non-key attributes; on the other hand, the fixed space is insufficient for other preview tables with more important non-key attributes.

Further, for obtaining either a coherent or a diverse preview, we enforce an additional constraint on the pairwise distance between preview tables. The distance between two preview tables T_1 and T_2 (denoted $dist(T_1, T_2)$) is the length of the shortest undirected path⁵ between their key attributes $T_1.key$ and $T_2.key$ in schema graph G_s . (Recall that the key attributes are vertices (i.e., entity types) in G_s .) For example, the distance between the two tables in Fig. 2 is 1, which is the shortest path length between FILM and FILM ACTOR in the schema graph in Fig. 3. Similarly, for the two tables whose key attributes are FILM and AWARD, their distance would be 2.

Based on the above notion of distance, the constraint on table distance is given by an integer d , which is the maximum (resp. minimum) distance between preview tables. The previews satisfying the distance constraint are called *tight* (resp. *diverse*) previews. Intuitively speaking, the preview tables in a tight preview are highly related to each other due to their short pairwise distance, while the preview tables in a diverse preview are not tightly related to each other and cover different types of concepts. Arguably, both types of previews are useful for understanding an entity graph. We shall compare them empirically in Sec. 6.

Given the spaces of all possible concise, tight and diverse previews, we formulate three optimization problems—finding an *optimal preview* with the highest score in the corresponding space of previews. Below we formally define the three types of previews and the corresponding optimization problems. Note that we assume the constraints k, n, d are given. While it is intuitive for a user to specify desired values for these constraints, it is helpful if a system can automatically suggest values. We leave it to future work.

Definition 2 (Concise, Tight and Diverse Previews). Given the size constraint (k, n) , a *concise preview* has k preview tables (i.e., key

attributes) and no more than n non-key attributes in the tables.⁶ The space of all concise previews is

$$\mathbb{P}_{k,n} = \{\mathcal{P} \mid \mathcal{P} \in \mathbb{P}, |\mathcal{P}| = k, \sum_{i=1}^k |\mathcal{P}[i].nonkey| \leq n\}.$$

Given the size constraint (k, n) and the distance constraint d , a *tight preview* (diverse preview) is a concise preview in which the distance between any pair of preview tables is smaller (greater) than or equal to d . The space of all tight previews is

$$\mathbb{P}_{k,n,\leq d} = \{\mathcal{P} \mid \mathcal{P} \in \mathbb{P}_{k,n}, \forall T_1, T_2 \in \mathcal{P}, dist(T_1, T_2) \leq d\}.$$

The space of all diverse previews is

$$\mathbb{P}_{k,n,\geq d} = \{\mathcal{P} \mid \mathcal{P} \in \mathbb{P}_{k,n}, \forall T_1, T_2 \in \mathcal{P}, dist(T_1, T_2) \geq d\}. \quad \square$$

Definition 3 (Optimal Preview Discovery Problem). The optimization problem of finding an *optimal preview* is defined as follows, where \mathbb{P} can be any of the aforementioned three spaces— $\mathbb{P}_{k,n}$, $\mathbb{P}_{k,n,\leq d}$ and $\mathbb{P}_{k,n,\geq d}$.

$$\mathcal{P}_{opt} \in \arg \max_{\mathcal{P} \in \mathbb{P}} S(\mathcal{P}) \quad (3)$$

Note that the $\arg \max$ function may return a set of optimal previews due to ties in scores. \square

For example, given the entity graph in Fig. 1, using coverage-based scoring measures for both key and non-key attributes, an optimal concise preview consisting of 2 tables and 6 non-key attributes (i.e., $k=2, n=6$) is $\mathcal{P} = \{T_1 : \text{FILM}, \text{Actor}, \text{Genres}, \text{Director}, \text{Producer}; T_2 : \text{FILM ACTOR}, \text{Actor}, \text{Award Winners}\}$. The edge *Actor* is a non-key attribute in both T_1 and T_2 , in different directions. An optimal diverse preview under the same size constraint ($k=2, n=6$) and distance constraint $d=2$ is $\mathcal{P} = \{T_1 : \text{FILM}, \text{Actor}, \text{Genres}, \text{Director}, \text{Producer}, \text{Executive Producer}; T_2 : \text{AWARD}, \text{Award Winners}\}$.

4.1 NP-hardness of the Optimal Tight and Diverse Preview Discovery Problems

The optimal preview discovery problem is non-trivial. Particularly, the problem in the spaces of both tight previews ($\mathbb{P}_{k,n,\leq d}$) and diverse previews ($\mathbb{P}_{k,n,\geq d}$) is NP-hard.

Theorem 1. The optimal tight preview discovery problem is NP-hard.

Proof. The decision version of the optimal tight preview discovery problem is *TightPreview*(G_s, k, n, d, s)—Given a schema graph G_s , decide whether there exists such a preview \mathcal{P} that (1) \mathcal{P} has k tables and no more than n non-key attributes; (2) the distance between every pair of preview tables is not greater than d ; and (3) the preview’s score is at least s , i.e., $S(\mathcal{P}) \geq s$.

We construct a reduction, in polynomial-time, from the NP-hard Clique problem to *TightPreview*(G_s, k, n, d, s). Recall that the decision version of *Clique*(G, k) is to, given a graph $G(V, E)$, decide whether there exists a clique in G with k vertices. The reduction is by constructing a schema graph G_s from G . For simplicity of exposition, in both this proof and the proof of Theorem 2, we assume the schema graph G_s is undirected and every edge γ in G_s corresponds to the same relationship type. This assumption is made without loss of generality. Note that our following proof casts no requirement on the score of a preview (i.e., $s = 0$) and thus no requirement on the scores of key and non-key attributes in G_s . Hence, edge orientation and its corresponding relationship type bears no significance in the proof.

⁶ A preview with less than n non-key attributes may outscore another preview with exactly n non-key attributes. Further, a set of k entity types may have only less than n edges in the schema graph. Hence, the condition $|\mathcal{P}[i].nonkey| \leq n$ instead of $|\mathcal{P}[i].nonkey| = n$. On the other hand, it is safe to assume that an entity graph with practical significance always has more than k entity types under any reasonably small k . Therefore an optimal preview always should have exactly k preview tables, given the monotonic scoring function (cf. Eq. 1).

⁵ An undirected path in a directed graph is a path in which the edges are not all oriented in the same direction.

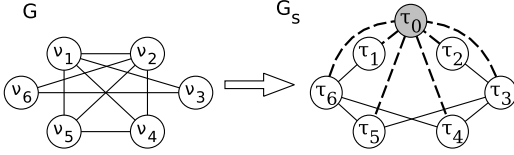


Figure 4: Construction of G_s from G , for Reduction from the Clique Problem to the Optimal Diverse Preview Discovery Problem.

Formally, we construct a schema graph $G_s(V_s, E_s)$ from G through a vertex bijection $f : V \rightarrow V_s$:

- $\forall e(v, v') \in E$, there exists an edge (i.e., relationship type) $\gamma(\tau, \tau') \in E_s$, where $\tau = f(v)$ and $\tau' = f(v')$.
- $\forall \gamma(\tau, \tau') \in E_s$, there exists an edge $e(v, v') \in E$, where $v = f^{-1}(\tau)$ and $v' = f^{-1}(\tau')$.

$\text{Clique}(G, k)$ is thus reduced to $\text{TightPreview}(G_s, k, k, 1, 0)$ by the above bijections. \square

The NP-hardness of the optimal diverse preview discovery problem is also based on a reduction from the Clique problem, although the proof is more complex.

Theorem 2. The optimal diverse preview discovery problem is NP-hard.

Proof. The decision version of the optimal diverse preview discovery problem is $\text{DiversePreview}(G_s, k, n, d, s)$ —Given a schema graph G_s , decide whether there exists such a preview \mathcal{P} that (1) \mathcal{P} has k tables and no more than n non-key attributes; (2) the distance between every pair of preview tables is not smaller than d ; and (3) the preview’s score is at least s , i.e., $S(\mathcal{P}) \geq s$.

We construct a reduction, in polynomial-time, from the NP-hard $\text{Clique}(G, k)$ to $\text{DiversePreview}(G_s, k, n, d, s)$. The reduction is also by constructing a schema graph $G_s(V_s, E_s)$ from G . It is similar to the reduction for $\text{TightPreview}(G_s, k, n, d, s)$ in Theorem 1, but also bears two important differences. (1) G_s contains a special vertex, denoted τ_0 , that is directly connected to every other vertex in G_s . (2) Barring τ_0 and all its incident edges, G_s is the complement graph of G —There is still a vertex bijection $f : V \rightarrow V_s$, but an edge exists between two vertices in G_s if and only if there is no edge between the corresponding vertices in G . Formally, the construction of G_s from G is as follows:

- $\forall \tau, \tau' \in V_s \setminus \{\tau_0\}$, $\gamma(\tau, \tau') \in E_s$ if and only if $\nexists e(v, v') \in E$, where $v = f^{-1}(\tau)$ and $v' = f^{-1}(\tau')$.
- $\forall \tau \in V_s \setminus \{\tau_0\}$, $\gamma(\tau, \tau_0) \in E_s$.

$\text{Clique}(G, k)$ is thus reduced to $\text{DiversePreview}(G_s, k, k, 2, 0)$ by the above construction of G_s . \square

Fig. 4 can help understand the reduction from $\text{Clique}(G, k)$ to $\text{DiversePreview}(G_s, k, k, 2, 0)$ in the above proof. The figure shows an example with G (left) and the constructed schema graph G_s (right), where the gray vertex in G_s is τ_0 . Consider an arbitrary pair of vertices (v, v') in G and their corresponding vertices (τ, τ') in G_s . On the one hand, if v and v' are not directly connected in G (e.g., v_1 and v_6), an edge between τ and τ' (i.e., τ_1 and τ_6) is included into G_s . When finding a diverse preview where pairwise table distance must be at least 2, τ and τ' will never be chosen together as the key attributes of two tables in the preview. Correspondingly, this means a clique must not include both v and v' . On the other hand, if v and v' are directly connected in G (e.g., v_1 and v_2), there must not be a direct edge between τ and τ' (i.e., τ_1 and τ_2) in G_s . The distance between τ and τ' is exactly 2, since they are only indirectly connected through τ_0 . They will thus be considered in choosing the key attributes of two tables in

a diverse preview where pairwise table distance must be at least 2. Correspondingly, the directly connected v and v' are considered together in forming a clique.

5. ALGORITHMS

In this section we discuss algorithms for solving the optimal preview discovery problem. As given in Eq. 3, the problem is to find a preview with the highest score among candidate previews, where the space of candidates can be concise previews ($\mathbb{P}_{k,n}$), tight previews ($\mathbb{P}_{k,n,\leq d}$) or diverse previews ($\mathbb{P}_{k,n,\geq d}$). Recall that we use $S(\tau)$ to denote the score of a candidate key attribute τ for a preview table T and $S^\tau(\gamma)$ to denote the score of a candidate non-key attribute $\gamma(\tau, \tau')$ (or $\gamma(\tau', \tau)$) for T whose key attribute is τ .

Our effort focuses on reducing the cost in finding optimal previews. Both the schema graph and the scoring measures defined in Sec. 3 are computed before optimal preview discovery. This is a realistic assumption, since the schema graph and scoring measures do not change by the size and distance constraints k, n, d . Furthermore, they can be incrementally updated when the underlying entity graph is updated (detailed discussion omitted). On the other hand, the optimal previews cannot be incrementally updated.

Before we present the algorithms, consider the space of all possible previews. Every entity type τ can be the key attribute of a preview table T . Let Γ^τ denote the set of all edges (i.e., relationship types) incident on τ in schema graph G_s . Any $\gamma \in \Gamma^\tau$ can be a candidate for the non-key attributes of T . By the scoring function in Eq. 2 and the problem formulation in Eq. 3, the non-key attributes of T must have the highest scores among the candidates in Γ^τ . This property, stated in Theorem 3, is important to our algorithms.

Theorem 3. Suppose an optimal (concise, tight or diverse) preview \mathcal{P}_{opt} contains a preview table $T \in \mathbb{T}$ with key attribute τ . If T has m non-key attributes, they must be the top- m non-key attributes by scores, i.e., $\forall \gamma, \gamma' \in \Gamma^\tau$, if $\gamma \in T.nonkey$ and $\gamma' \notin T.nonkey$, then $S^\tau(\gamma) \geq S^\tau(\gamma')$.

5.1 A Brute-Force Algorithm

Alg. 1 is a brute-force algorithm for the optimal preview discovery problem. It enumerates all possible k -subsets of entity types, as the k entity types in each subset form the key attributes of k preview tables in a preview \mathcal{P} (Line 4). For a candidate key attribute τ , the elements in the set of its candidate non-key attributes Γ^τ are ordered by their scores. We denote these candidates in descending order of scores by $\gamma_1^\tau, \gamma_2^\tau$, and so on (Line 2). Suppose preview table T uses τ as its key attribute. Each table must contain at least one non-key attribute, according to Definition 1. Hence, γ_1^τ (i.e., the candidate non-key attribute with the highest score) must be included into $T.nonkey$ (Line 8), by Theorem 3. Further, among the remaining candidate non-key attributes for the k entity types, the top- $(n-k)$ candidates by scores must be included into \mathcal{P} (Lines 11–14), by Theorem 3. Note that, since the sorted list of candidate non-key attributes for each τ is already created (Line 2), it is unnecessary to do a full sorting in order to determine the top- $(n-k)$ candidates Γ . Instead, a simple merge operation on the k sorted lists will get Γ .

The algorithm has an exponential complexity $O(KN \log N + \binom{K}{k}(k+n))$, where $K = |V_s|$ is the number of candidate key attributes, $N = 2|E_s|$ is the number of candidate non-key attributes for all candidate key attributes, $\binom{K}{k}$ is the number of k -subsets, and $KN \log N$ is for sorting individual lists of candidates (Line 2), in which each list contains at most N elements.

Alg. 1 is for finding one of the optimal previews. To find all optimal previews, it needs simple extension to deal with ties in

Algorithm 1: Brute-Force Algorithm for Optimal Preview Discovery

Input : schema graph G_s , size constraint (k, n)
Output: an optimal preview \mathcal{P}_{opt}

```

1 foreach  $\tau \in V_s$  do
2    $\langle \gamma_1^\tau, \gamma_2^\tau, \dots \rangle \leftarrow$  sort the candidate non-key attributes  $\gamma_j^\tau \in \Gamma^\tau$ 
   by their scores  $S^\tau(\gamma_j^\tau)$ ;
3  $max\_score \leftarrow 0$ ;  $\mathcal{P}_{opt} \leftarrow \emptyset$ ;
4 foreach  $k$ -subset of  $V_s$  (denoted  $V$ ) do
5    $score \leftarrow 0$ ;  $\mathcal{P} \leftarrow \emptyset$ ;  $i \leftarrow 1$ ;
6   foreach  $\tau \in V$  do
7      $\mathcal{P}[i].key = \tau$ ;
8      $\mathcal{P}[i].nonkey = \{\gamma_1^\tau\}$ ;
9      $score = score + S(\tau) \times S^\tau(\gamma_1^\tau)$ ;
10     $i \leftarrow i + 1$ ;
11    $\Gamma \leftarrow$  top- $(n-k)$  candidate non-key attributes from all  $\tau \in V$  in
   descending order of  $S(\tau) \times S^\tau(\gamma_j^\tau)$ ;
12   foreach  $\gamma_j^\tau \in \Gamma$ , where  $\tau = \mathcal{P}[x].key$  do
13      $score \leftarrow score + S(\tau) \times S^\tau(\gamma_j^\tau)$ ;
14      $\mathcal{P}[x].nonkey \leftarrow \mathcal{P}[x].nonkey \cup \{\gamma_j^\tau\}$ ;
15   if  $score > max\_score$  then
16      $max\_score \leftarrow score$ ;
17      $\mathcal{P}_{opt} \leftarrow \mathcal{P}$ ;
18 return  $\mathcal{P}_{opt}$ ;

```

scores, which we will not further discuss.

The same brute-force algorithm is applicable for optimal preview discovery in all three types of spaces—concise, tight and diverse previews. The pseudo code in Alg. 1 is for concise previews and does not enforce distance constraint, for simplicity of presentation. Enforcing distance constraint for tight/diverse previews is straightforward, by performing distance check on every pair of preview tables in each k -subset of entity types.

5.2 A Dynamic-Programming Algorithm for Concise Preview Discovery Problem

As the combinatorial number of k -subsets grows exponentially, the performance of the above brute-force algorithm becomes unacceptable for finding an optimal preview under modest size constraints. We thus developed a dynamic-programming algorithm to discover optimal concise previews more efficiently.

Consider an arbitrary order on all K entity types— τ_1, \dots, τ_K . We use $\mathcal{P}_{opt}(k, n, x)$ to denote an optimal concise preview among the first x entity types τ_1, \dots, τ_x . Thus the optimal concise preview discovery problem is to find $\mathcal{P}_{opt}(k, n, K)$. $\mathcal{P}_{opt}(k, n, x)$ can be constructed from the solutions to smaller problems, in two ways: (1) It can be equal to $\mathcal{P}_{opt}(k, n, x-1)$, i.e., its k tables and n non-key attributes are from the first $x-1$ entity types and the x -th entity type τ_x does not contribute anything; (2) It can also be the union of $\mathcal{P}_{opt}(k-1, n-m, x-1)$ and a table T_x^m , where $\mathcal{P}_{opt}(k-1, n-m, x-1)$ is an optimal preview with $k-1$ tables and $n-m$ non-key attributes among the first $x-1$ entity types, and T_x^m is the table whose key attribute is τ_x and whose non-key attributes are the top- m elements in Γ^{τ_x} —the sorted list of candidate non-key attributes for τ_x . The number m is between 1 and $n-(k-1)$ (or less if there are less than $n-(k-1)$ elements in Γ^{τ_x}), since each of the $k-1$ tables in $\mathcal{P}_{opt}(k-1, n-m, x-1)$ must contribute at least one non-key attribute. The optimal substructure of the problem is as follows. (We do not discuss boundary cases ($k=1$ or $x=1$ or $n=k$) due to space limitations.)

$$\mathcal{P}_{opt}(k, n, x) = \arg \max_{\mathcal{P} \in \mathbb{P}(k, n, x)} S(\mathcal{P})$$

Algorithm 2: Dynamic-Programming Algorithm for Optimal Concise Preview Discovery

Input : schema graph G_s , size constraint (k, n)
Output: an optimal concise preview \mathcal{P}_{opt}

```

1 foreach  $x \leftarrow 1$  to  $K$  do
2    $\langle \gamma_1^{\tau_x}, \gamma_2^{\tau_x}, \dots \rangle \leftarrow$  sort the candidate non-key attributes
    $\gamma_j^{\tau_x} \in \Gamma^{\tau_x}$  by their scores  $S^{\tau_x}(\gamma_j^{\tau_x})$ ;
3 for  $x \leftarrow 1$  to  $K$  do
4   for  $i \leftarrow 1$  to  $\min(k, x)$  do
5     for  $j \leftarrow i$  to  $n$  do
6        $\mathcal{P}_{opt}(i, j, x) \leftarrow \mathcal{P}_{opt}(i, j, x-1)$ ;
7       for  $m \leftarrow 1$  to  $\min(j-i+1, |\Gamma^{\tau_x}|)$  do
8          $T_x^m.key \leftarrow \tau_x$ ;
9          $T_x^m.nonkey \leftarrow$  top- $m$  candidate non-key
           attributes in  $\Gamma^{\tau_x}$ ;
10         $\mathcal{P} \leftarrow \mathcal{P}_{opt}(i-1, j-m, x-1) \cup \{T_x^m\}$ ;
11        if  $S(\mathcal{P}) > S(\mathcal{P}_{opt}(i, j, x))$  then
12           $\mathcal{P}_{opt}(i, j, x) \leftarrow \mathcal{P}$ ;
13  $\mathcal{P}_{opt} \leftarrow \mathcal{P}_{opt}(k, n, K)$ ;
14 return  $\mathcal{P}_{opt}$ ;

```

$$\mathbb{P}(k, n, x) = \left\{ \begin{array}{l} \mathcal{P}_{opt}(k, n, x-1), \\ \mathcal{P}_{opt}(k-1, n-1, x-1) \cup \{T_x^1\}, \\ \mathcal{P}_{opt}(k-1, n-2, x-1) \cup \{T_x^2\}, \\ \dots \\ \mathcal{P}_{opt}(k-1, k-1, x-1) \cup \{T_x^{n-(k-1)}\} \end{array} \right\},$$

where $T_x^m.key = \tau_x$ and $T_x^m.nonkey =$ top- m candidate non-key attributes in Γ^{τ_x} . Note that the optimal substructure is inapplicable when previews must satisfy distance constraint in addition to size constraint (details omitted). Therefore the dynamic-programming algorithm is for concise previews but not tight/diverse previews.

The pseudo code of the dynamic-programming algorithm is shown in Alg. 2. Its complexity is $O(KN \log N + Kkn^2)$. Similar to Alg. 1, Alg. 2 is for finding one optimal preview. Finding all optimal previews requires simple extension to deal with ties in scores, which we will not further discuss.

Both Alg. 1 and 2 assume that, given any k entity types (key attributes), they always together have at least n non-key attributes. That may not be true in reality. In fact, for two previews with the same number of tables, the preview with less non-key attributes may have the higher score than the other preview. Note that, in Eq. 3, the optimal preview is not required to have exactly n non-key attributes. It is simple to extend Alg. 1 and 2 to fully comply with the definition. Given any entity type τ , if it has less than n candidate non-key attributes, we can simply pad the sorted list Γ^τ by pseudo non-key attributes with zero scores.

5.3 An Apriori-style Algorithm for Tight / Diverse Preview Discovery Problem

Since the dynamic-programming algorithm is inapplicable when previews must satisfy distance constraint, we propose an efficient algorithm for optimal tight/diverse preview discovery, shown in Alg. 3. It consists of two steps: (1) finding k -subsets of entity types (i.e., vertices in G_s) satisfying the distance constraint (Lines 1–14); (2) for each qualifying k -subset of entity types, forming a preview under the size constraint, computing its score and choosing a preview with the highest score (Lines 15–20).

The first step is essentially finding k -cliques in a graph converted from the schema graph G_s , in which vertices are considered adjacent if they are within distance d (for tight previews) or apart by at least distance d (for diverse previews). The k -clique problem is well-studied and many efficient algorithms have been designed

Domain	# of vertices	# of edges
books	6M / 91	15M / 201
film	2M / 63	18M / 136
music	27M / 69	187M / 176
people	3M / 45	17M / 78
tv	2M / 59	17M / 177
basketball	19K / 6	557K / 21
architecture	133K / 23	432K / 48

Table 2: Sizes of Entity/Schema Graphs.

Domain	Coverage	Entropy
books	0.8	0.786
film	0.2	0.25
music	0.528	0.589
people	0.708	0.606
tv	0.622	0.379

Table 3: MRR of Non-Key Attribute Scoring.

Domain	Key Attribute			Non-Key Attribute	
	YPS09	Coverage	Random Walk	Coverage	Entropy
books	0.4	0.55	0.43	0.43	0.43
film	-0.01	0.48	0.25	0.35	0.35
music	0.37	0.33	0.46	0.42	0.41
people	0.36	0.31	0.29	0.43	0.43
tv	0.37	0.69	0.65	0.47	0.47

Table 4: PCC of Key and Non-Key Attribute Scoring.

Algorithm 3: Apriori-style Algorithm for Optimal Tight/Diverse Preview Discovery

Input : schema graph G_s , size constraint (k, n) , distance constraint d
Output: an optimal tight/diverse preview \mathcal{P}_{opt}

```

1  $\mathcal{L}_2 \leftarrow \emptyset$ ;
2 foreach  $i \leftarrow 1$  to  $K$  do
3   foreach  $j \leftarrow i + 1$  to  $K$  do
4     if  $\text{dist}(\tau_i, \tau_j) \leq d$  then /*  $\geq d$  for diverse preview */
5        $\mathcal{L}_2 \leftarrow \mathcal{L}_2 \cup \{\{i, j\}\}$ ;
6  $i \leftarrow 3$ ;
7 while  $i \leq k$  and  $\mathcal{L}_{i-1} \neq \emptyset$  do
8    $\mathcal{L}_i \leftarrow \emptyset$ ;
9   foreach  $A, B \in \mathcal{L}_{i-1}$  s.t.  $(\forall j < i - 1 : A[j] = B[j])$  and
     $(A[i - 1] < B[i - 1])$  do
10     /*  $\geq d$  for diverse preview */
11     if  $\text{dist}(\tau_{A[i-1]}, \tau_{B[i-1]}) \leq d$  then
12        $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \{A[1] \dots A[i - 1] B[i - 1]\}$ ;
13  $i \leftarrow i + 1$ ;
14 if  $\mathcal{L}_k = \emptyset$  then
15   return  $\emptyset$ ;
16  $\text{max\_score} \leftarrow 0$ ;
17 foreach  $A \in \mathcal{L}_k$  do
18    $\mathcal{P} \leftarrow \text{ComputePreview}(A)$ ;
19   if  $\text{score}(\mathcal{P}) > \text{max\_score}$  then
20      $\text{max\_score} \leftarrow \text{score}(\mathcal{P})$ ;
21    $\mathcal{P}_{opt} \leftarrow \mathcal{P}$ ;
22 return  $\mathcal{P}_{opt}$ ;

```

in the past. Our method is inspired by the well-known Apriori algorithm [1] for frequent itemset mining. In [11], an algorithm was proposed for finding k -cliques (where edges correspond to metabolite correlations) by similar ideas, although the connection to Apriori was not made. Their experimental results demonstrated superior efficiency in comparison with the more well-known Bron-Kerbosch algorithm [6]. Nevertheless, the two broad steps of our optimal tight/diverse preview discovery algorithm are independent from each other, and thus any more efficient or even approximate algorithm for finding k -cliques can be plugged into it to further improve its execution efficiency.

In more details, the first step of Alg. 3 iteratively generates a k -subset of entity types by merging two $(k-1)$ -subsets. Entity types are arbitrarily ordered as τ_1, \dots, τ_K . In the i -th iteration of the algorithm, if two $(i-1)$ -subsets A and B only differ by their last entity types $\tau_{A[i-1]}$ and $\tau_{B[i-1]}$, and the distance between their last entity types satisfies the distance constraint, a candidate i -subset is generated by appending $\tau_{B[i-1]}$ to the end of A .

In the second step, for each candidate k -subset of entity types, a preview is computed ($\text{ComputePreview}(A)$ in Line 17 of Alg. 3). The details of function ComputePreview are omitted. It follows Theorem 3 and is essentially the same as Lines 5–14 in Alg. 1. The score of each preview is computed (the same as in Lines 5–14 of Alg. 1) and a preview with the highest score is returned.

The worst-case complexity of Alg. 3 is the same as that of Alg. 1.

However, as Sec. 6 shows, in practice it significantly outperforms the brute-force algorithm, since Line 10 could filter out many combinations that do not satisfy the distance constraint.

6. EVALUATION

We conducted experiments to evaluate the preview scoring measures’ accuracy (Sec. 6.1), the preview discovery algorithms’ efficiency (Sec. 6.2), and the overall quality of discovered previews (Sec. A). All experiments were run on a Dell T100 server running Ubuntu 8.10. The server has a Dual Core Xeon E3120 processor, 6MB cache, 4GB RAM, and two 250GB RAID1 SATA hard drives. All algorithms are implemented in C++ and compiled with ‘-O2’ optimization in GCC-4.3.2.

The entity graph used in our experiment is a dump of Freebase at September 28, 2012.⁷ The dataset is imported into an MySQL database. In Freebase, the entire entity graph is partitioned into many domains. Our experiments were conducted on eight domains. The sizes of the entity and schema graphs in these domains are shown in Table 2. Our work currently is limited to named entities, thus all numeric attribute values from the data dump have been removed. Note that a schema graph may be disconnected. To ensure the convergence of random walk in such a graph, we added a small transition probability 10^{-5} to every pair of entity types.

6.1 Accuracy of Preview Scoring Measures

We conducted two experiments to evaluate the accuracy of the scoring measures for both key and non-key attributes presented in Sec. 3. One experiment compares the ranking orders of candidate key (non-key) attributes by the scoring measures with gold standard ranking orders. The other experiment calculates the correlation between two pairwise ordering results on candidate key (non-key) attributes—one by the scoring measures and the other collected through crowdsourcing. In both experiments, we used both measures proposed in this paper and an adaptation of the approach in [19].

6.1.1 Adaptation of [19]

Yang et al. [19] proposed an algorithm to summarize relational databases, specifically the tables in TPC-E benchmark.⁸ Their approach works in three steps. First they define an importance value for each table considering both information content of the tables and join relationships between the tables. Second, they measure the similarity/distance between tables. Finally, they use a weighted k -center clustering algorithm to place the tables into k clusters. The k cluster centers are the summary of the database. We implemented their algorithm. We compared the results on TPC-E tables with those reported in [19] and validated our own implementation.

We adapted [19] for the entity graphs in the aforementioned Freebase domains. Since [19] was designed to summarize relational databases only, we converted each entity graph into a relational database, as follows. For each entity type τ , we created a relational table, of which the first column takes entities belonging to τ as its

⁷ <https://developers.google.com/freebase/data>

⁸ <http://www.tpc.org/tpce/>

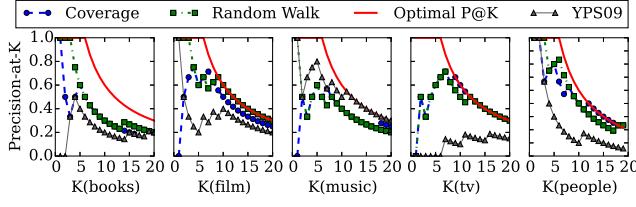


Figure 5: Precision-at- K of Key Attribute Scoring.

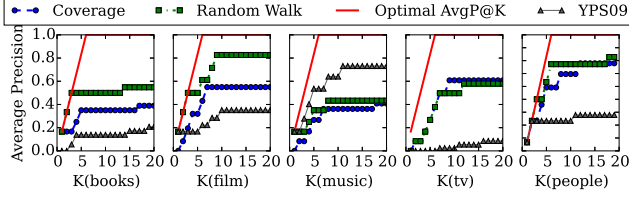


Figure 6: Average Precision of Key Attribute Scoring.

values. Furthermore, a column is created for each relationship type incident on τ in the scheme graph. The values in such a column are the entities adjacent to the entities in the first column through the corresponding relationship type. For each entity belonging to τ , a number of tuples are inserted into the table, which are essentially a Cartesian product of distinct values on all these columns.

6.1.2 Comparison with Gold Standard

Key attributes:

We collected gold standard data for 5 largest entity domains in Freebase—“books”, “film”, “music”, “people” and “tv”. For each domain, Freebase offers an entrance page showing 6 major entity types in that domain. A user can choose to browse entities in any of the 6 types.⁹ As such entrance pages were manually created by Freebase, our conjecture is that they are of high quality and reflect the most popular entity types. We thus treated the 6 entity types listed in the entrance page of a domain as the gold standard for top-6 key attributes in that domain. The schema of the tables in the gold standard can be found in Table 12 in the Appendix.

For both the coverage-based and the random-walk based scoring measures in Sec. 3.2, we ranked all candidate key attributes by their scores. We calculated the accuracy of a scoring measure by several widely-used IR evaluation measures, including Precision-at- K ($P@K$), Average Precision (AvgP) and Normalized Discounted Cumulative Gain (nDCG) [13]. An approach that ranks accurate results higher is expected to receive better values under these measures. For a scoring measure for key attributes, $P@K$ is the percentage of its top- K results that belong to the aforementioned gold standard top-6 key attributes. For the adaptation of [19], we use the ranked list by their table importance scoring. The results are in Fig. 5. The topmost curves (“Optimal $P@K$ ”) represent the best possible $P@K$ that can be archived by any method. For instance, $P@10$ can be at most 0.6, since there are only 6 gold standard key attributes in each domain, as mentioned above. Fig. 5 shows that both the coverage-based and the random-walk based scoring measures had $P@10$ close to 0.6 in 4 out of the 5 domains. They both had significantly higher $P@K$ values than [19] (denoted “YSP09”) in 4 out of the 5 domains and similar values in the remaining domain.

⁹ The entrance pages were all under “Featured Data” on Freebase.com. For instance, <http://www.freebase.com/view/film> was the entrance page for domain “film”. We collected these pages shortly after September 28, 2012, which is the timestamp of the Freebase entity graph dump used in our experiments. These pages have become unavailable lately.

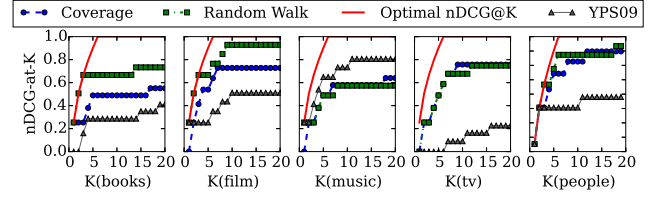


Figure 7: nDCG of Key Attribute Scoring.

We also used AvgP and nDCG to gauge the accuracy of the scoring measures for key attributes. The results are as follows:

- **Average Precision (AvgP):** The average precision of the top- k results is given by $\text{AvgP} = \frac{\sum_{i=1}^k P@i \times rel_i}{\text{size of ground truth}}$, where rel_i equals 1 if the result at rank i is in the ground truth and 0 otherwise. Fig. 6 shows significantly higher AvgP for both the coverage-based and the random-walk based scoring measures, compared to [19], in 4 out of 5 domains.
- **Normalized Discounted Cumulative Gain (nDCG):** The cumulative gain of the top- k results is $\text{DCG}_k = rel_1 + \sum_{i=2}^k \frac{rel_i}{\log_2(i)}$. It penalizes the results if a ground truth result is ranked low. DCG_k is normalized by IDCG_k , the cumulative gain for an ideal ranking of the top- k results. Thus $\text{nDCG}_k = \frac{\text{DCG}_k}{\text{IDCG}_k}$. It is shown in Fig. 7 that both the coverage-based and the random-walk based scoring measures had clearly higher nDCG, in comparison with [19], in 4 out of the 5 domains.

Non-key attributes:

For each entity type, Freebase offers a table for users to browse and query the entities belonging to that type.¹⁰ The table always has 3 common columns for recording names, types and article contents of entities. It also has 3 or less type-dependent non-key attributes manually selected by Freebase editors. Although Freebase allows users to add more attributes into this table, we believe the original 3 type-dependent attributes in general bear higher quality. We thus treated these attributes as the gold standard for top non-key attributes for that entity type.

For both the coverage-based and the entropy-based scoring measures in Sec. 3.3, we ranked all candidate non-key attributes by their scores. There is no comparison with [19] regarding non-key attributes, since it does not have a component that can be adapted for discovering non-key attributes. We calculated the accuracy of a scoring measure by Mean Reciprocal Rank (MRR) [13] instead of $P@K$ as there are only 3 or less gold standard answers for top non-key attributes in each entity type. For a scoring measure for non-key attributes, the reciprocal rank is the multiplicative inverse of the rank of the first gold standard non-key attribute among its ranking results. MRR is the average reciprocal rank across all entity types with at least 5 candidate non-key attributes. (If an entity type has only less than 5 candidates, the gold standard answers are ranked deceptively high. Thus we exclude such entity types, to obtain more accurate evaluation.) The results are shown in Table 3. In every domain except “film” and for both the coverage-based and the entropy-based measures, MRR is above 0.5. This means in average a gold standard non-key attribute appeared in the top-2 ranked results. The lower MRR for “film” domain is from only one entity type and thus is not truly indicative, since only that entity type has at least 5 candidate non-key attributes.

6.1.3 Correlation with Ranking Based on Crowdsourcing

¹⁰ <http://www.freebase.com/music/artist?instances=>, for instance, would display a table for type ARTIST in “music” domain.

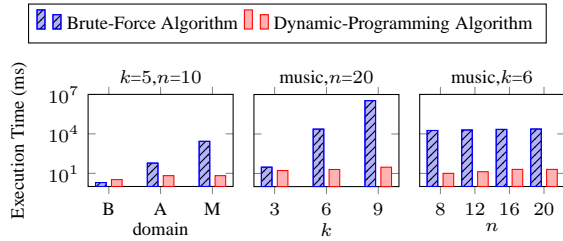


Figure 8: Execution Time of Optimal Concise Preview Discovery Algorithms.

We conducted an extensive study in Amazon Mechanical Turk (AMT)—a popular crowdsourcing service—and measured the correlation between our scoring measures and users’ opinions with regard to key and non-key attributes ranking. We explain the procedure for evaluating key attribute ranking in one domain, since the procedure is repeated for all 5 gold standard domains and is the same for both key and non-key attribute ranking.

Given a domain, we randomly generated 50 pairs of entity types, i.e., candidate key attributes. Each pair was presented to 20 AMT workers. The workers were asked which of the 2 entity types in the pair is more important. To help them understand the tasks, we provided a few examples to explain what are considered more important in common sense. The workers were also asked to answer a few screening questions that test their common knowledge. They must answer the screening questions correctly, otherwise their responses are not considered.

We collected 1,000 opinions (50 pairs \times 20 workers per pair) in total. We then constructed two lists— X and Y , each of which contains 50 values corresponding to the 50 pairs. A value in X represents the difference in the ranking positions (by our scoring measures, or by the table importance measure in [19]) of the two entity types in the corresponding pair. A value in Y represents the difference in the numbers of AMT workers favoring the two entity types. The correlation between X and Y is measured by Pearson Correlation Coefficient (PCC) [7] as follows.

$$PCC = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - (E(X))^2} \sqrt{E(Y^2) - (E(Y))^2}} \quad (4)$$

The PCC value ranging from -1 to 1 indicates the degree of correlation between the pairwise ranking orders produced by our scoring methods and the pairwise preferences given by AMT workers. A PCC value in the ranges of $[0.5, 1.0]$, $[0.3, 0.5]$ and $[0.1, 0.3]$ indicates a strong, medium and small positive correlation, respectively. PCC values for the 5 gold standard domains are in Table 4. For all 5 domains, the results show at least a medium positive correlation between our scoring measures and AMT workers. For 4 out of the 5 domains, the coverage-based and/or the entropy-based measures had significantly higher PCC values than the adaptation of [19] (“YPS09”), which even demonstrated slightly negative correlation in the “film” domain.

6.2 Efficiency of Algorithms

This section presents results on the efficiency of the optimal preview discovery algorithms in Sec. 5. On optimal concise preview discovery, we compared the Brute-Force Alg. 1 and the Dynamic-Programming Alg. 2. Specifically, we compared their execution times by varying: (1) size of schema graph (i.e., number of candidate key attributes (K) and number of candidate non-key attributes (N)); (2) number of preview tables (i.e., key attributes) in a preview (k); and (3) maximum number of non-key attributes in a preview (n). For (1), we fixed $k=5$, $n=10$ and experimented with 3 domains—“basketball” (B), “architecture” (A), and “music” (M). They differ greatly in the sizes of their schema graphs (B: $K=6$,

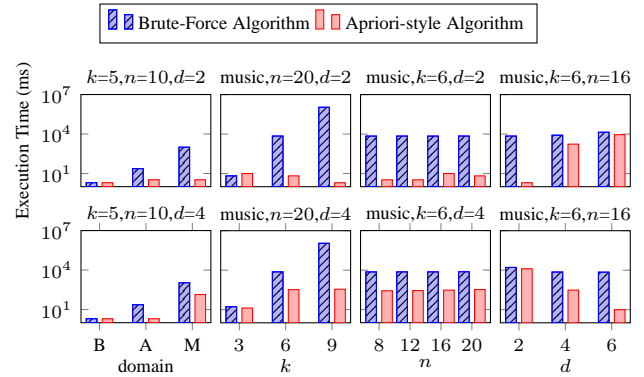


Figure 9: Execution Time of Optimal Tight (Upper) and Diverse (Lower) Preview Discovery Algorithms.

$N=21$; A: $K=23$, $N=48$; M: $K=46$, $N=133$). For (2), we varied k from 3 to 9, fixed $n=20$ and used “music” domain. For (3), we varied n from 8 to 20, fixed $k=6$ and used “music” domain.

On optimal tight/diverse preview discovery, we compared the Brute-Force Alg. 1 and the Apriori-style Alg. 3, by varying not only the aforementioned 3 parameters but also the distance constraint on d . When we varied other parameters, d is fixed at 2 and 4 for tight and diverse previews, respectively. When we fixed other parameters, d was varied from 2 to 6.

The results are in Figs. 8 and 9. In all results, the execution time is averaged across 3 runs, and execution time less than 1 millisecond is rounded to 1 millisecond. The results show that both the Dynamic-Programming and the Apriori-style algorithms outperformed the Brute-Force algorithm by orders of magnitude in most cases. The exceptions are the smallest domain “basketball” and when the number of requested preview tables is small ($k=3$). In these cases, the overheads of complex data structures and calculations in the advanced algorithms outweighed their benefits.

Fig. 9 shows that the Apriori-style algorithm did not perform well for $d=6$ in tight preview discovery and $d=2$ in diverse preview discovery. It is due to the excessive number of candidate k -subsets that satisfy the distance constraint in such cases. For instance, the diameter of a schema graph typically is not large. In the schema graph of “film” domain, the longest path length is 7 and the average path length is around 3–4. Setting distance constraint $d=6$ in finding tight previews will make most previews “tight”. It is unnecessary to enforce such a distance constraint.

6.3 User Study

We conducted an extensive user study to compare seven different approaches, including concise previews, tight previews, diverse previews, hand-crafted previews by experts, Freebase gold standard (cf. Section 6.1.2, and Table 12 in Appendix), schema summarization based on “YPS09” [19], and directly using schema graphs. For each approach, we created a website for presenting schema information using the approach and collecting participants’ responses, for each of the five domains—“books”, “film”, “music”, “people” and “tv”.

To produce hand-crafted previews, we used a group of 10 experts (PhD students in the database area in the author’s institution). Each expert participant was rewarded a \$20 gift card. For each domain, we set the expected numbers of key attributes (K) and non-key attributes (N) to be the same as the values in the Freebase gold standard (cf. Table 12 in Appendix). Each expert was requested to produce preview tables under the size constraints given by K and N . During the process, the experts had access to the Freebase website, to help them understand the data. After the experts worked on

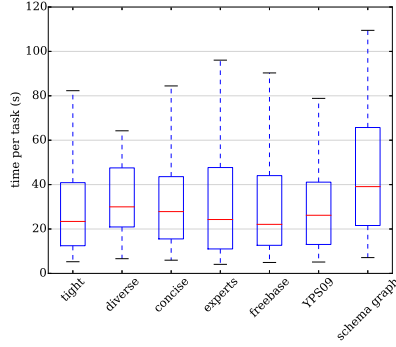


Figure 10: User Study: Time Taken on Existence Tests, Domain=“music”

all the five domains, they were asked to discuss among themselves and submit one consolidated preview for each domain. We use the consolidated previews as the hand-crafted previews in the ensuing user study. On average the experts spent around 10 minutes on the simplest domain “people” and more than 30 minutes on the most complex domain “film”. After that, the experts spent about 2 hours to discuss. The significant amount of time spent by the individual experts and by them together in discussion suggests that it is a challenging and time-consuming process to generate preview tables. This motivates the need for an automatic approach.

The participants of the user study include 84 computer science graduate students in the authors’ institution. They all have taken database courses. None of them was affiliated with the authors’ research group or exposed to the research project. Each participant was rewarded a \$15 gift card.

Each participant was randomly assigned to use one of the aforementioned seven approaches (websites). Each approach received 10-13 participants. Before a participant started their session, they were given a 20-minute introduction on the approach of presenting schema information that they are using. For each domain they were requested to answer 4 existence test questions about the existence/nonexistence of some specific information in the schema and 4 user experience questions. An example existence test question is “Based on this schema summary, I know the dataset provides the awards of a musician.” The participants were requested to provide a Boolean yes/no answer.

6.3.1 Existence Test Questions

The existence test questions were designed to measure how helpful the various approaches are in assisting the participants to acquire a good understanding of the data. We first verify if the approaches are convenient to use, in terms of how much time the participants must spend to answer the existence test questions. We recorded the time spent by every participant on every existence test question. Figure 10 for domain “music” suggests that all approaches except “schema graph” required similar amount of time on average, while “schema graph” required significantly more time. This is not difficult to understand, given the complexity of a schema graph. On the other hand, the other six table-based approaches enjoyed their simple structure of presentation. Figure 10 also suggests larger variance for “Experts” and “schema graph” and smaller variance for “Diverse” (diverse previews).

Based on the participants’ responses to the existence test questions, we measured the effectiveness of the seven approaches using *conversion rate*, which is the percentage of questions correctly answered by the participants. Based on the conversion rates, we conducted A/B testing to compare the seven approaches in a pairwise fashion. Table 5 summarizes the results for various pairs in

System	Sample Size	Conversion rate (<i>c</i>)	z-value	p-value
Tight	48	CT=0.979	-1.71	0.0436
Experts	48	CE=0.895		
Tight	48	CT=0.979	-1.09	0.1379
Freebase	52	CF=0.931		
Tight	48	CT=0.979	-1.32	0.0918
YPS09	52	CY=0.923		
Tight	48	CT=0.979	-1.85	0.0322
Schema Graph	40	CS=0.875		
Concise	52	CC=0.796	-1.64	0.0505
Tight	48	CT=0.8125		
Concise	52	CC=0.796	-2.34	0.0096
Diverse	52	CD=0.730		
Tight	48	CT=0.979	-3.83	0.0001
Diverse	52	CD=0.730		

Table 5: User Study: Accuracy on Existence Tests in Different Systems, Domain=“music”

System	Q1	Q2	Q3	Q4
tight	3.6667	3.8333	4.0833	3.75
diverse	3.75	3.75	3.9167	3
concise	3.8462	3.8462	4.1538	3.5833
experts	4.1667	4.1667	4.5	4.3333
freebase	3.8182	4.2727	4.4545	3.5455
YPS09	4.3077	4.5385	4.4615	3.8333
schema graph	3.6	4.6	4.5	3.9

Table 7: User Experience Based on Responses to Questionnaire, Domain=“music”

the “music” domain. (Due to space limitations, it does not list all 21 pairs given the seven approaches.) Since we are interested in comparing certain pairs, the conversion rates of approaches such as “Tight” (tight previews) are repeated several times.

We observe that “Tight” has a better conversion rate than all other approaches, even including the hand-crafted previews and Freebase gold standard. It means that on average the participants of tight previews had a higher chance of answering existence test questions correctly. We performed a two sample Z-test with significance level $\alpha=0.1$. We assume that answering the existence test questions follows a Bernoulli trial. For example, the probabilities of successfully answering the questions using “Tight” and “YPS09” are $p_T=CT$ and $p_Y=CY$, respectively. For comparing “Tight” with “YPS09”, our hypothesis, H_{A1} , is that “Tight” has a better conversion rate than “YPS09”: $H_{A1}: p_T > p_Y$. The null hypothesis is given by $H_{01}: p_T \leq p_Y$. For the aforementioned conversion rates of “Tight” and “YPS09”, and the respective sample sizes in Table 5, $z=-1.32$. This results in a p-value of 0.0918. Since $p\text{-value} < \alpha$, the null hypothesis will be rejected and the data significantly supports our claim that “Tight” users have a higher chance in answering the existence test questions correctly. By similar analysis, we can reach the conclusion that participants using “Tight” have a higher chance of answering the questions correctly, in comparison with other approaches. The only exception is the comparison of “Tight” and “Freebase”, in which $p\text{-value} > \alpha$ and thus the null hypothesis cannot be rejected.

6.3.2 User Experience Questions

For each domain, we asked a participant four user experience questions, after the four existence test questions. The four questions: Q1, Q2, Q3 and Q4, are listed in Table 6. Each question had five options, specifying the level of agreement a user could have with the particular aspect of the approach measured by the question. We assign a score for every option in each question based on the Likert scale shown in Table 6. The least favourable

Likert Scale Score	Q1: How easy was it to read the schema summary of this domain?	Q2: How much understanding of the data in this domain can you gain from the schema summary?	Q3: How helpful was the schema summary in assisting you to understand the data of this domain?	Q4: Is the schema summary missing important information about data in this domain?
1	Very hard	Very little	Not helpful at all	It provides very little important information.
2	Hard	A Little	Did not help much	It provides some important information.
3	Neutral	Neutral	Neutral	Neutral
4	Easy	Some	Somewhat helpful	It provides most of the important information.
5	Very easy	Very much	Very helpful	It provides all important information.

Table 6: User Experience Questionnaire.

experience with respect to each question is assigned a score of 1, and the most favoured experience is assigned a score of 5.

For a certain approach, the overall user experience score for each question is measured by averaging the scores obtained for that question from all the participants using that approach. Table 7 shows the results for the “music” domain. The scores on Q1 suggest that the participants’ perception on the easiness of the approaches they use is consistent with the amount of time they spent (cf. Figure 10). What is interesting is that the participants favored “YPS09” the most, in terms of easiness of use, although they answered the existence test questions less accurately using “YPS09” than using approaches such as “Tight”.

Another interesting observation is that the participants believed they acquired more understanding of the data when they used “schema graph” (Q2 and Q3), although the results from existence test questions (cf. Table 5) suggests that they answered questions more accurately when they use approaches such as “Tight”. Finally, the results on Q4 seems to indicate that the previews prepared by experts gave the participants the impression of having more complete information.

7. RELATED WORK

There have been several studies on schema summarization for relational databases [19, 20, 21], XML [21] and general graph data [17, 22]. [21] produces schema summarization for relational databases and XML data. Its summary is in the form of a condensed schema tree where a node may correspond to multiple nodes or a trunk in the original schema tree. The notion of summary in [19, 20] refers to clustering the tables in a database by their semantic roles and similarities as well as identifying direct join relationships and indirect join paths between the tables. The graph summarization in [17, 22] groups graph nodes based on their attribute similarity and allows users to browse the summary from different grouping granularities. As explained in Sec. 1, these methods are inapplicable or ineffective for producing preview tables from entity graphs, due to differences in input/output data models and goals.

There are many works on graph clustering [15]. They are not effective for generating preview tables, since clustering focuses on partitioning but does not present a concise structure. On the contrary, a preview only selects a small number of key attributes (vertices) and non-key attributes (edges) from a schema graph.

[14] proposed the concept of queried units (“qunits”) for representing desired query results on a database. For automatic derivation of qunits, [14] discussed several ideas. One idea is to utilize the concept of queriability [10] which measures the importance of a schema entity based on its schema connectedness and its data cardinality. The measure is thus similar to the key attribute scoring measures in Section 3.2. ObjectRank [3] applies authority-based ranking to keyword search in databases. Part of its ranking formula is extended from PageRank. The table importance measure in [19] and our random-walk based scoring measure in Section 3.2 bear similar ideas.

[9] studied how to generate query result snippets in XML search.

Similar to [21], they focus on semi-structured data. Differently, they produce snippets of query results while [21] summarizes schema. In [9], the problem of generating snippets is formulated as maximizing information under an upper bound on snippet size. At high level, this is similar to our problem of finding optimal previews under size constraint, although its detailed problem formulation, solution, and data model are different.

8. CONCLUSION

This paper studies how to generate preview tables for entity graphs. The problem is challenging due to the scale and complexity of such graphs. We proposed effective scoring measures for preview tables. We proved that the optimal preview discovery problem under distance constraint is **NP-hard**. We designed efficient algorithms for discovering optimal previews. Our experiments on Freebase data verified the accuracy and efficiency of our methods.

For future work, one direction is to study how to select representative entity tuples for preview tables. Using random tuples may not be adequate since a user may not be familiar with them. Another direction is to investigate how to incorporate numeric attributes into preview tables. It is not straightforward to compare the importance of numeric values and entities. Additionally, we would like to conduct further research on suggesting optimal N and K to the user. Recommending the optimum distance constraint for tight and diverse methods is also an interesting challenge that requires further study.

9. REFERENCES

- [1] R. Agarwal and R. Srikant. Fast algorithms for mining association rules. In *VLDB*, pages 487–499, 1994.
- [2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, , and Z. Ives. DBpedia: A nucleus for a Web of open data. In *ISWC*, 2007.
- [3] A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, pages 564–575, 2004.
- [4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.
- [5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW*, pages 107–117, 1998.
- [6] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *CACM*, 16(9):575–577, Sept. 1973.
- [7] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. 1988.
- [8] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmman, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610, 2014.

- [9] Y. Huang, Z. Liu, and Y. Chen. Query biased snippet generation in xml search. In *SIGMOD*, pages 315–326, 2008.
- [10] M. Jayapandian and H. V. Jagadish. Automated creation of a forms-based database query interface. *PVLDB*, 1(1):695–709, Aug. 2008.
- [11] F. Kose, W. Weckwerth, T. Linke, and O. Fiehn. Visualizing plant metabolomic correlation networks using clique-metabolite matrices. *Bioinformatics*, 17(12):1198–1208, Dec. 2001.
- [12] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, Mar. 2009.
- [13] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [14] A. Nandi and H. V. Jagadish. Qunits: queried units in database search. In *CIDR*, 2009.
- [15] S. E. Schaeffer. Survey: Graph clustering. *Comput. Sci. Rev.*, 1(1):27–64, Aug. 2007.
- [16] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge unifying WordNet and Wikipedia. In *WWW*, pages 697–706, 2007.
- [17] Y. Tian, R. A. Hankins, and J. M. Patel. Efficient aggregation for graph summarization. In *SIGMOD*, pages 567–580, 2008.
- [18] W. Wu, H. Li, H. Wang, and K. Q. Zhu. Probase: a probabilistic taxonomy for text understanding. In *SIGMOD*, pages 481–492, 2012.
- [19] X. Yang, C. M. Procopiuc, and D. Srivastava. Summarizing relational databases. *PVLDB*, 2(1):634–645, 2009.
- [20] X. Yang, C. M. Procopiuc, and D. Srivastava. Summary graphs for relational database schemas. *PVLDB*, pages 899–910, 2011.
- [21] C. Yu and H. V. Jagadish. Schema summarization. In *VLDB*, pages 319–330, 2006.
- [22] N. Zhang, Y. Tian, and J. M. Patel. Discovery-driven graph summarization. In *ICDE*, pages 880–891, 2010.

APPENDIX

A. SAMPLE OPTIMAL PREVIEWS

To demonstrate the combined effectiveness of both scoring measures and preview discovery algorithms, Table 8 presents the optimal concise previews in 3 selected domains by 3 different combinations of key attribute scoring (KS) and non-key attribute scoring (NKS) measures. The size constraint is set as $k=5$ and $n=10$. All result previews show that the selected key and non-key attributes have covered important entity types and their important relationship types. Further, Table 9 shows the optimal tight ($d=2$) and diverse ($d=4$) previews in “film” domain by one particular choice of key and non-key attribute scoring measures. We see that, in the tight preview result, the chosen key attributes are all highly related to one entity type *FILM*. In the diverse preview result, the chosen key attributes are far less related to each other. Both verify the effectiveness of the concepts of tight/diverse previews.

Note that in the generated previews, certain non-key attributes represent relationship types involving more than two entity types. An example in Table 8 is *Portrayed in films*, which is a non-key attribute of entity type *FILM CHARACTER*. Different from other non-key attribute such as *Films directed*, it represents a 3-way relationship among *FILM CHARACTER*, *FILM* and *FILM ACTOR*. For instance, Agent J is a *FILM CHARACTER* played by *FILM ACTOR* Will Smith in *FILM* Men in Black. To present the values of such a *multi-way* non-key attribute in a preview table, we employ a simple approach of presenting values for all participating entity types in this relationship. It is arguable

Key attributes	Non-key attributes (Target entity types)
Domain=“film”, KS=Coverage, NKS=Coverage, $k=5$, $n=10$	
FILM CHARACTER FILM ACTOR FILM	<i>Portrayed in films</i> (FILM, FILM ACTOR) <i>Film performances</i> (FILM, FILM CHARACTER) <i>Performances</i> (FILM ACTOR, FILM CHARACTER), <i>Genres</i> (FILM GENRE), <i>Runtime</i> (FILM CUT), <i>Country of origin</i> (COUNTRY), <i>Directed by</i> (FILM DIRECTOR), <i>Languages</i> (HUMAN LANGUAGE) <i>Films directed</i> (FILM) <i>Films crewed</i> (FILM, FILM CREW ROLE)
FILM DIRECTOR FILM CREWMEMBER	
Domain=“music”, KS=Random Walk, NKS=Coverage, $k=5$, $n=10$	
MUSICAL RECORDING	<i>Releases</i> (MUSICAL RELEASE), <i>Tracks</i> (RELEASE TRACK), <i>Recorded by</i> (MUSICAL ARTIST) <i>Tracks</i> (MUSICAL RECORDING), <i>Track list</i> (RELEASE TRACK)
MUSICAL RELEASE	<i>Release</i> (MUSICAL RELEASE), <i>Recording</i> (MUSICAL RECORDING)
RELEASE TRACK	<i>Tracks recorded</i> (MUSICAL RECORDING)
MUSICAL ARTIST MUSICAL ALBUM	<i>Releases</i> (MUSICAL RELEASE), <i>Release type</i> (MUSICAL ALBUM TYPE)
Domain=“tv”, KS=Random Walk, NKS=Entropy, $k=5$, $n=10$	
TV EPISODE	<i>Previous episode</i> (TV EPISODE), <i>Next episode</i> (TV EPISODE), <i>Performances</i> (TV ACTOR, TV CHARACTER), <i>Season</i> (TV SEASON), <i>Series</i> (TV PROGRAM) , <i>Personal appearances</i> (PERSON, PERSONAL APPEARANCE ROLE) <i>Regular acting performances</i> (TV ACTOR, TV CHARACTER, TV SEASON)
TV PROGRAM	<i>Episodes</i> (TV EPISODE)
TV SEASON TV ACTOR	<i>TV episode performances</i> (TV EPISODE, TV CHARACTER)
TV DIRECTOR	<i>TV episodes directed</i> (TV EPISODE)

Table 8: Sample Optimal Concise Previews.

that this approach widens the preview table, which to some extent violates a given size constraint. An alternative solution is to use separate preview tables for all multi-way relationships. These pose interesting directions for our future work.

B. ADDITIONAL RESULTS

The schema of the tables in the gold standard can be found in Table 12 in the Appendix. There are 6 key attributed and at most 3 non-key attributes for each key-attribute in the gold standard.

Table 10 shows the precision-at-k for Freebase gold standard using the experts-generated preview tables as ground truth.

Table 11 shows the precision-at-k for experts’ hand-crafted preview tables using Freebase gold standard as ground truth.

Key attributes	Non-key attributes (Target entity types)
Domain="film", KS=Coverage, NKS=Coverage, k=5, n=10, d=2	
FILM	<i>Performances</i> (FILM CHARACTER, FILM ACTOR), <i>Genres</i> (FILM GENRE), <i>Runtime</i> (FILM CUT), <i>Country of origin</i> (COUNTRY), <i>Directed by</i> (FILM DIRECTOR), <i>Languages</i> (HUMAN LANGUAGE)
FILM DIRECTOR	<i>Films directed</i> (FILM)
FILM PRODUCER	<i>Films produced</i> (FILM)
FILM WRITER	<i>Film writing credits</i> (FILM)
FILM EDITOR	<i>Films edited</i> (FILM)
Domain="film", KS=Coverage, NKS=Coverage, k=5, n=10, d=4	
FILM CHARACTER	<i>Portrayed in films</i> (FILM, FILM ACTOR), <i>Portrayed in films (dubbed)</i> (FILM, FILM ACTOR)
FILM CREWMEMBER	<i>Films crewed</i> (FILM, FILM CREW ROLE)
PERSON OR ENTITY	<i>Films appeared in</i> (FILM, TYPE OF APPEARANCE)
APPEARING IN FILM	
FILM FESTIVAL	<i>Individual festivals</i> (FILM FESTIVAL EVENT), <i>Location</i> (LOCATION), <i>Focus</i> (FILM FESTIVAL FOCUS), <i>Sponsoring organization</i> (SPONSER)
FILM COMPANY	<i>Films</i> (FILM)

Table 9: Sample Optimal Tight (Upper) and Diverse Previews (Lower).

K	film	books	music	TV	people
1	1	1	1	1	1
2	0.5	0.5	1	1	1
3	0.334	0.334	1	1	0.664
4	0.5	0.25	1	0.75	0.5
5	0.6	0.2	1	0.6	0.6
6	0.5	0.333	0.833	0.5	0.5

Table 10: precision-at-k of key attribute scoring, using gold standard as ground truth.

K	film	books	music	TV	people
1	1	1	1	1	1
2	0.5	1	1	1	0.5
3	0.667	0.667	1	0.667	0.667
4	0.75	0.5	1	0.75	0.75
5	0.6	0.4	0.8	0.6	0.6
6	0.5	0.333	0.833	0.5	0.5

Table 11: precision-at-k of key attribute scoring, using Experts as ground truth.

Key attributes	Non-key attributes
Domain="film", KS=Coverage, NKS=Coverage, k=6, n=9	
FILM	<i>Directed By</i> <i>Tagline</i> <i>Initial Release Date</i>
FILM ACTOR	<i>Film performances</i>
FILM GENRE	<i>Films of this genre</i>
FILM DIRECTOR	<i>Films directed</i>
FILM PRODUCER	<i>Films Executive Produced</i> <i>Films Produced</i>
FILM WRITER	<i>Film Writing Credits</i>
Domain="book", k=6, n=15	
BOOK	<i>Characters</i> <i>Genre</i> <i>Editions</i> <i>Publication Date</i> <i>Publisher</i> <i>Credited To</i>
BOOK EDITION	<i>Genre</i> <i>Characters</i>
SHORT STORY	<i>Characters</i> <i>Meter</i> <i>Verse Form</i>
POEM	<i>Mode Of Writing</i> <i>Verse Form</i>
SHORT NON-FICTION	<i>Series Written (Or Contributed To)</i> <i>Works Edited</i> <i>Works Written</i>
AUTHOR	
Domain="music", k=6, n=18	
COMPOSITION	<i>Includes</i> <i>Lyricist</i> <i>Composer</i>
CONCERT	<i>Venue</i> <i>Start Date</i> <i>Concert Tour</i>
MUSIC VIDEO	<i>Song</i> <i>Initial release date</i>
MUSICAL ALBUM	<i>Artist</i> <i>Release Type</i> <i>Initial Release Date</i>
MUSICAL ARTIST	<i>Artist</i> <i>Albums</i> <i>Place Musical Career Began</i> <i>Musical Genres</i>
MUSICAL RECORDING	<i>Length</i> <i>Featured artists</i> <i>Recorded by</i>
Domain="tv", k=6, n=9	
TV PROGRAM	<i>Program Creator</i> <i>Air Date Of First Episode</i> <i>Air Date Of Final Episode</i> <i>Starring TV Roles</i> <i>Programs In Which This Was A Regular Character</i>
TV ACTOR	<i>TV Programs (Recurring Writer)</i>
TV CHARACTER	<i>TV Programs Produced</i>
TV WRITER	<i>TV Episodes Directed</i>
TV PRODUCER	<i>TV Segments Directed</i>
TV DIRECTOR	
Domain="people", k=6, n=16	
PERSON	<i>Profession</i> <i>Country Of Nationality</i> <i>Date Of Birth</i> <i>Cause Of Death</i> <i>Place Of Death</i> <i>Date Of Death</i>
DECEASED PERSON	<i>People Who Died This Way</i> <i>Includes Causes Of Death</i> <i>Parent Cause Of Death</i>
CAUSE OF DEATH	<i>Geographic Distribution</i> <i>Includes Group(S)</i> <i>Included In Group(S)</i>
ETHNICITY	<i>Specializations</i> <i>Specialization Of</i> <i>People With This Profession</i>
PROFESSION	<i>Professions In This Field</i>
PROFESSIONAL FIELD	

Table 12: Gold Standard.