Data-Semantics-Aware Recommendation of Diverse Pivot Tables

WHANHEE CHO, University of Utah, USA
ANNA FARIHA, University of Utah, USA

Data summarization is essential to discover insights from large datasets. In spreadsheets, *pivot tables* offer a convenient way to summarize tabular data by computing aggregates over some attributes, grouped by others. However, identifying attribute combinations that will result in *useful* pivot tables remains a challenge, especially for high-dimensional datasets. We formalize the problem of automatically recommending *insightful* and *interpretable* pivot tables, eliminating the tedious manual process. A crucial aspect of recommending a *set* of pivot tables is to *diversify* them. Traditional work inadequately address the table-diversification problem, which leads us to the problem of *pivot table diversification*.

We present SAGE, a data-semantics-aware system for recommending k-budgeted diverse pivot tables, overcoming the shortcomings of prior work for top-k recommendations that cause redundancy. SAGE ensures that each pivot table is *insightful*, *interpretable*, and *adaptive* to the user's actions and preferences, while also guaranteeing that the set of pivot tables are different from each other, offering a *diverse* recommendation. We make two key technical contributions: (1) a *data-semantics-aware model* to measure the utility of a single pivot table and the diversity of a set of pivot tables, and (2) a *scalable greedy algorithm* that can efficiently select a set of diverse pivot tables of high utility, by leveraging data semantics to significantly reduce the combinatorial search space. Our extensive experiments on four real-world datasets show that SAGE outperforms alternative approaches, and efficiently scales to accommodate high-dimensional datasets. Additionally, through multiple case studies, we demonstrate SAGE's qualitative superiority over existing tools, and through a user study, we validate its practical usefulness and alignment with user preferences.

CCS Concepts: • **Information systems** → **Database utilities and tools**; **Data analytics**; **Summarization**; **Recommender systems**; *Information retrieval diversity*.

Additional Key Words and Phrases: Pivot Tables, Data Semantics, Diversification.

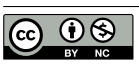
ACM Reference Format:
Whanhee Cho and Anna Fariha. 2026. Data-Semantics-Aware Recommendation of Diverse Pivot Tables. *Proc. ACM Manag. Data* 4, 1 (SIGMOD), Article 23 (February 2026), 28 pages. https://doi.org/10.1145/3786637

1 Introduction

Data is at the heart of data-driven decision making. We rely on trends observed in the data to obtain *insights* [28] that help us make informed decisions. However, due to limitations in human comprehensibility, data must be *summarized* [34, 40, 65] to enable humans discover insights from the summaries, either directly or via visualizations [70] over the summaries. One of the most common techniques to summarize data is *aggregation*. Simple aggregations involve functions (e.g, `SUM`) to aggregate all rows. More nuanced aggregations involve multiple groupings of the entities (e.g., `GROUP BY GENDER, EDUCATION`) and then aggregating each group separately.

While SQL provides functionalities for any custom aggregation query, it is not suitable for novices due to interface-related limitations. Thanks to the ubiquity of spreadsheet software—such

Authors' Contact Information: Whanhee Cho, University of Utah, Salt Lake City, Utah, USA, whanhee.cho@utah.edu; Anna Fariha, University of Utah, Salt Lake City, Utah, USA, afariha@cs.utah.edu.



ACM 2836-6573/2026/2-ART23
https://doi.org/10.1145/3786637

ID	Gender	Age	Experience	…	Degree	Department	Salary
1	Male	48	3	…	PhD	IT	$50,000
2	Female	32	1	…	MS	Sales	$20,000
3	Male	45	12	…	PhD	HR	$100,000

Table 1. A sample table from an employee compensation dataset.

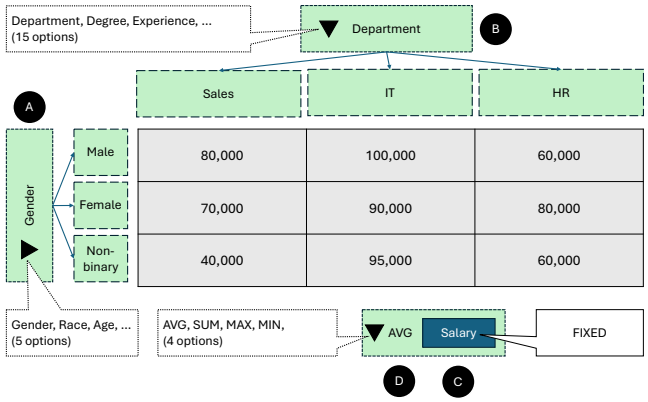


Fig. 2. A pivot table requires 4 parameters: Ⓐ row-groups, Ⓑ column-groups, Ⓒ aggregate attributes, and Ⓓ aggregate functions for each aggregate attribute. Users can choose multiple values for each parameter. In Example 1.1, Sasha has fixed Ⓒ, but needs to explore Ⓐ (5 options), Ⓑ (15 options), and Ⓓ (4 options).

as Microsoft Excel [48], Google Sheets [35], Apple Numbers [38], etc.—a substantial portion of businesses (around 60% [27]) and about 2 billion people [20] use spreadsheets for data management and analysis. These users rely on *pivot tables* [17], a summary of tabular data that computes aggregates over a few data attributes, grouped by other data attributes. Most commercial spreadsheets include a built-in and user-friendly mechanism to construct pivot tables. Spreadsheet pivot tables are particularly suitable for novices, where they can rearrange, group, and aggregate data using intuitive interfaces such as drag-and-drop. Unlike SQL aggregates, spreadsheet pivot tables offer dynamic user interactions, allowing interactive exploration such as drilling down, filtering, sorting, etc.

In an exploratory setting with the goal to discover interesting data trends, a key challenge in constructing insightful pivot tables lies in selecting the right *parameters*, i.e., determining which attributes to use for groupings and aggregations. This task becomes even harder for users who lack knowledge of how the data is represented (especially when the data contains missing or cryptic attribute names), lack domain knowledge, or work with high-dimensional data. In such cases, users must manually explore a vast space of parameter combinations through a tedious trial-and-error process. This involves experimenting with various combinations of (1) grouping attributes, (2) aggregation attributes, and (3) aggregation functions, then manually assessing the insightfulness of the resulting pivot tables. We illustrate this challenge with Example 1.1.

Example 1.1. *Sasha is investigating potential factors affecting salary across various groups in an employee compensation dataset over* 21 *attributes including* `ID`, `Gender`, `Age`, `Experience`, `Degree`, `Department`, `Salary`, *etc. (Table 1). With an aim to discover salary discrepancies across various group combinations, she starts with the pivot table shown in Fig. 2: she puts* `Gender` *in the row-groups* Ⓐ *and* `Department` *in the column-groups* Ⓑ*; and chooses* `Salary` *as an aggregate attribute* Ⓒ *and* `Average` *as the aggregate function* Ⓓ*.*

Sasha is interested in `Salary` *discrepancies, so her choice for* Ⓒ *is fixed. However, she still needs to explore various combinations for* Ⓐ*,* Ⓑ*, and* Ⓓ*. Sasha wishes to put demographic attributes (e.g.,*

Tool	Recommended Pivot Tables
Google Sheets [35] - Test conducted in February 2025	(1) Average of Age, Years of Experience, Annual Bonus, Overtime Hours, Sick Days, Training Hours, Satisfaction Score, #Projects, #Promotions by Gender (2) Average of Performance Rating for each Gender by Department (3) Average of Age, Years of Experience, Performance Rating, Salary, Annual Bonus, Overtime Hours, Sick Days, Training Hours, Satisfaction Score, #Projects, #Promotions by Gender
Microsoft Excel [48] - Microsoft Excel (Windows) version 2501 - Test conducted in February 2025	(1) Count of ID by Degree (2) Count of ID by Department (3) Sum of Employed Year by Department (4) Sum of #Promotions by Employed Year and Degree (5) Sum of Age, Children, Performance Rating by Degree (6) Sum of Children, Performance Rating, Salary by Degree (7) Sum of Children, Performance Rating, Salary by Department (8) Sum of Salary by Employed Year and Gender (9) Sum of Employed Year by Gender and Degree
ChatGPT [54] - Tested on ChatGPT (GPT-4o) in February 2025	(1) Average Salary by Years of Experience and Training Hours (2) Average Salary by Department and Children (3) Average Salary by Age and Satisfaction Score

Table 3. Google Sheets recommendations are redundant and convoluted; Microsoft Excel includes meaningless recommendations such as `SUM(Age)`; while ChatGPT recommendations look reasonable, they are data-content-unaware as the pivot table values are hallucinated.

`Gender`, `Race`, `Age`, `Marital Status`, *etc.) in the row-groups, as any discrepancy across different rows will indicate discrimination, and all other attributes in the column-groups.*

Sasha decides to explore 5 *demographic and* 15 *non-demographic attributes, as well as* 4 *aggregation functions:* `MAX`, `MIN`, `AVERAGE`, *and* `SUM`. *This leaves her* $5 \times 15 \times 4 = 300$ *possible combinations,*[1] *and she must carefully inspect each pivot table to identify salary discrepancies, by manually contrasting the pivot table cells. Assuming each pivot table has* 10 *cells on average and it takes about* 2 *minutes to examine each pivot table, Sasha needs* $300 \times 2 = 600$ *minutes (*10 *hours)!*

Recommending Pivot Tables. Example 1.1 highlights the need for a *recommendation* system that can automatically suggest the "best" pivot tables. While existing spreadsheet software, such as Microsoft Excel and Google Sheets, are equipped with features for automatic pivot table recommendation, they have several shortcomings, which we show next in Example 1.2.

Example 1.2. *Frustrated by manual exploration, Sasha tries the pivot table recommendations in Google Sheets (Table 3), obtaining three recommendations. However, the recommended tables often include too many aggregated attributes beyond her desired* `Salary`*, resulting in convoluted and large tables. Sasha also observes that most recommendations are redundant—they default to the groupings by* `Gender` *or* `Department`*—and lack diversity, causing her to miss out on insights involving other data attributes. While Microsoft Excel provides nine recommendations, it utilizes only 10 out of 21 possible attributes. Additionally, it suggests meaningless aggregations like* `Sum(Employed Year)` *and* `Sum(Age)`*, revealing its shortcoming in grasping the semantics. For both MS Excel and Google Sheets, Sasha failed to specify* `Salary` *as her intended aggregate attribute, restricting her ability to steer the recommendations towards her needs. Lastly, Sasha asks ChatGPT for three "insightful" and "diverse" pivot tables, focusing on average* `Salary`*. Apparently reasonable at first, she soon realizes that the values of the pivot tables are hallucinated, exposing ChatGPT's lack of access to the actual data and absence of result validation. Sasha concludes that LLMs are ill-suited for this task, as they do not explicitly enumerate and evaluate all possible options.*

Example 1.2 highlights several key limitations of existing tools for automatic recommendation of pivot tables. First, they do not cater to the user needs for a *focused* and *adaptive* recommendation

[1]Sasha chose only one option for each parameter. Multiple options (e.g., `Gender` and `Race` for row-groups) will further increase the search space of possible pivot tables.

of pivot tables. Second, they focus on top-k recommendations [26, 78, 79] and do not consider *diversification* [30], which may cause the users to miss certain data insights. Finally, existing approaches do not fully leverage the data and its semantics to ensure that the suggested pivot tables are *useful*, i.e., *insightful* and *interpretable*. We propose SAGE, a data-semantics-aware system for recommending k-budgeted diverse pivot tables, which overcomes the shortcomings of the existing approaches. We summarize the limitations of currently available tools and research work in Table 4 to contrast them against SAGE, and defer a detailed discussion to Section 8.

Problem. The problem we study in this paper is recommending a *diverse set* of pivot tables, under a *size* constraint, while ensuring that each recommended pivot table is *useful*, meaning it is *insightful* and *interpretable*. Furthermore, we want to achieve two usability goals during recommendation: (1) *adaptivity*, which takes into consideration already explored pivot tables by the users, and (2) *customizability*, which enables the users to guide the recommendation process by specifying certain data attributes to prioritize.

Challenges. We now highlight three key challenges that are associated with the problem:

Challenge 1: semantic modeling of pivot table utility. A useful pivot table must be *insightful*, to inform users of meaningful and non-obvious patterns, and *interpretable*, so that users can easily and quickly extract insights from it. Prior work [24, 28, 36, 70] ignore the *semantic aspect*; they use purely statistical measures to model insightfulness, without considering interpretability and semantics of the insight. For instance, the aggregate `SUM(Birth_Year)` is semantically meaningless, even if it indicates strong statistical insight. Furthermore, semantically modeling insightfulness and interpretability of a pivot table in a *multi-group setting* (e.g., group by `Gender`, `Department`, `Degree`) is non-trivial and is not addressed in prior work. In summary, how to model insightfulness and interpretability of a pivot table while remaining aware of the data semantics is a key challenge.

Challenge 2: modeling table diversity. Beyond recommending insightful and interpretable pivot tables, our goal is to also *diversify* the set of pivot tables. To the best of our knowledge, the notion of diversity in the context of pivot tables is not defined in prior work. Existing diversification approaches [12, 29–31] do not trivially extend for "table diversification", where the items under consideration are entire tables rather than individual tuples. Prior work for recommending insightful data summaries [78] or visualizations [70] do not consider diversity. For pivot table diversification, the key challenge is to develop an appropriate distance metric to model both the syntactic (e.g., attribute coverage) and semantic (e.g., provided insights) distances between a pair of pivot tables.

Challenge 3: developing an efficient system. Our goal is to recommend highly insightful and interpretable pivot tables, while ensuring diversity among them. Unlike insightfulness and interpretability, which can be measured for each pivot table in isolation, diversity requires considering a *set* of pivot tables. Note that the number of candidate pivot tables grows exponentially with the number of data attributes. Furthermore, finding the best fixed-sized set of pivot tables from these candidates is identical to the minimum set-cover problem, due to the search space growing exponentially with the number of candidates. Consequently, the problem is NP-hard in the number of candidates, which grows exponentially with the number of data attributes. Furthermore, evaluating insightfulness of a pivot table requires its materialization, which adds to the computational complexity. While greedy approaches with approximation guarantees [12, 15, 30, 51, 58] can alleviate the problem of combinatorial search, the requirement of materializing candidate pivot tables remain. Even with approximation algorithms [3, 21, 37, 73] for efficient materialization, without aggressive pruning before materialization, far too many candidates become the bottleneck. Therefore, a key challenge here is to develop mechanisms that can leverage semantic understanding of the data to prune unpromising pivot tables and avoid unnecessary materialization. Another challenge is to discover

Legends
● Always
◒ Partially
○ Not supported
⊗ Unknown

		Commercial software				Research work		LLMs			This work
		Microsoft Excel [48]	Google Sheets [35]	PowerBI [22, 28]	Tableau [62]	DAISY [78]	AutoSuggest [79]	ChatGPT [54]	Llama3-instruct [47]	TableGPT [68]	**SAGE**
Desirable Properties	Budgeted recommendations	○	○	○	○	○	○	●	●	●	●
	Guarantees syntactic validity	●	●	●	●	●	●	○	○	○	●
	Guarantees semantic validity	○	○	●	●	●	●	○	○	○	●
	Ensures interpretability	●	○	●	●	⊗	●	●	◒	◒	●
	Adaptive to user actions	○	○	○	○	○	●	●	●	●	●
	Allows user specifications	○	○	○	●	○	○	●	●	●	●
	Ensures diversity	○	○	○	○	○	○	●	●	●	●
	Attribute-name semantics aware	●	●	●	○	○	●	●	●	●	●
	Attribute-order insensitive	○	○	●	●	○	○	○	○	○	●
	Data-semantics aware	◒	○	●	○	●	○	◒	◒	◒	●
	No additional requirements	●	●	●	○	○	○	○	○	○	●
	Low-cost	●	●	●	●	●	●	○	●	●	●
	Open-source	○	○	○	○	○	○	○	◒	◒	●

Table 4. SAGE satisfies all desirable properties. While PowerBI, Tableau, DAISY, and AutoSuggest do not directly/always recommend pivot tables, we include them since they recommend summaries. Code for DAISY and AutoSuggest are unavailable, thus we rely on the papers, and mark some things as unknown. LLMs are not designed to directly recommend pivot tables but can be prompted to do so.

effective techniques that can "push down" [80] components of the diversity requirements to the search process to further prevent unnecessary pivot table materialization.

Contributions. Our main contribution is development of a novel system SAGE, for recommendation of a diverse set of useful pivot tables under a budget (size) constraint. Below, we provide the key contributions we make in this paper:

- We motivate and *formalize the problem* of budgeted recommendation of a diverse set of useful pivot tables, model it as a *constrained optimization problem*, and establish its *desiderata* (Section 2).
- We provide a formal model to measure the *utility* of a pivot table in terms of insightfulness and interpretability. Unlike prior work, our utility model leverages data semantics (Section 3).
- We establish the notion of *pivot-table diversification*, a key component of the problem we study in this paper. Our contribution lies in the formulation of a suitable *distance metric*—which considers both structural and semantic properties of pivot tables—and its application to diversifying a set of pivot tables (Section 4).
- To ensure SAGE's efficiency and practicality, we must tackle the NP-hardness of the problem. To reduce the search space, we introduce *aggressive semantic pruning*. To expedite the recommendation process, we leverage offline computation and "push down" diversity requirements to the search process (Section 5).
- Through an empirical analysis over 4 real-world datasets and case studies, we show that SAGE outperforms prior approaches while staying scalable and efficient (Section 6).
- We present a user study that validates our utility model and demonstrates SAGE's superiority over competing baselines based on human perception (Section 7).

(a)

Gender	Degree		
	BS	MS	PhD
Male	200K	300K	1000K
Female	100K	200K	300K

Avg Salary by Gender & Degree

(b)

Gender	Department	
	IT	Sales
Male	1000K	500K
Female	400K	200K

Avg Salary by Gender & Dept

(c)

Gender	Degree, Department					
	BS		MS		PhD	
	IT	Sales	IT	Sales	IT	Sales
Male	4	1	8	1	10	1
Female	2	8	2	3	1	2

Count Id by Gender, Degree, & Dept

(d)

Degree	Department	
	IT	Sales
BS	200K	100K
MS	300K	200K
PhD	900K	400K

Avg Salary by Degree & Dept

Fig. 5. Four pivot tables over the dataset of Table 1. While (a) and (b) indicate gender-based salary gap, (c) and (d) add additional context.

2 Recommending a Diverse set of Pivot Tables

In this section, we motivate the need for diversity and adaptivity during recommending pivot tables (Section 2.1). Then we develop the desiderata for the problem (Section 2.2) and formalize it (Section 2.3).

2.1 The Need for Diversity and Adaptivity

A key limitation of top-k recommendation is that it may provide redundant information, causing the users to miss out on relatively less useful, but complementary data insights. Such lack of *diversity* may even mislead the users to believe in partial insights that are "half-true". Another shortcoming of existing approaches is that they are not *adaptive* to user actions, i.e., when the user acknowledges a recommendation, it should be excluded from the subsequent iterations. However, commercial pivot table recommendation features are not adaptive (Table 4). We proceed to provide an example to highlight the need for *diverse* recommendation to help the users get a broader picture of the dataset, and *adaptive* [44, 64] recommendation to enable the user guide the recommendation process.

Example 2.1. *Recall from Example 1.1 that Sasha is interested in salary discrepancies. She initially finds two pivot tables (Fig. 5 (a) & (b)) suggesting gender-based pay gap. However, a deeper pattern emerges when she expands her analysis using other aggregate functions (e.g.,* COUNT*) and discovers the pivot tables shown in Fig. 5 (c) & (d), which provide her an additional context that the discrepancy stems from the hiring process: employee counts are uneven across degrees and departments. Sasha also notes that IT employees earn more than those in Sales, and PhDs earn more than others. This indicates degree- and department-based discrepancies, which are expected and acceptable. Sasha concludes that males earn more on average not due to gender bias, but largely because more male PhDs work at IT.*[2]

Furthermore, in an incremental setting where Sasha iteratively requests for recommendations of a few pivot tables at a time, she expects the system to adapt to her actions. For instance, after she accepts or rejects the suggestions of Fig. 5 (a) and (b), the system should avoid recommending redundant pivot tables that reiterate the same concept (gender-based salary gap) across other aspects (e.g., marital status).[3]

2.2 Desiderata

We now list key desiderata for a pivot-table recommendation system:

D1. Each recommended pivot table must provide *insightful* [24, 70] and *semantically interesting* information. For instance, a significant gap in average salary across genders provides insight into gender-based pay gap.

D2. Each pivot table must be *interpretable*, ensuring ease of comprehension by humans. For instance, a concise table with 10 cells is more interpretable than one with 1000 cells.

[2]This phenomenon is known as Simpson's paradox [9, 11]. While our focus is not to expose Simpson's paradox, we show this as a motivating use-case.

[3]This example does not dispute the established link between marital status and gender pay gaps.

D3. While insightfulness and interpretability model the goodness of a single pivot table, a desirable property for a set of pivot tables is *diversity*. Thus, the recommended set of pivot tables must minimize redundancy, covering various data aspects.

D4. The system for pivot table recommendation must allow (I) *customizability*—allowing users to specify the desired size of the recommendation set, degree of diversity, data scope, etc.—and (II) *adaptiveness* to user actions.

D5. Finally, the system must be *efficient* and *scalable*—to ensure handling large and high-dimensional data effectively—and *accessible*—in terms of cost and availability.

2.3 Problem Formulation

We now formalize our problem for a single-relation database instance (dataset) D.

Definition 2.1 (Pivot Table). Given a dataset D over a set of attributes $\mathbf{A}$ and the domain of aggregate functions $\mathcal{F} = \{\texttt{COUNT}, \texttt{SUM}, \texttt{AVG}, \texttt{MIN}, \texttt{MAX}\}$, a pivot table $T(\mathbf{F}(\mathbf{V}), \mathbf{G})$ takes the form: `SELECT` $\mathbf{F}(\mathbf{V})$ `FROM` D `GROUP BY` $\mathbf{G}$, where, $\mathbf{G} = \{G_1, G_2, \dots\} \subseteq \mathbf{A}$ is a subset of attributes for grouping; $\mathbf{V} = \{V_1, V_2, \dots\} \subseteq \mathbf{A}$ is a subset of attributes for computing aggregates over; $\mathbf{G} \cap \mathbf{V} = \emptyset$ ensures that no attribute is used for both grouping and aggregation; $\mathbf{F} = \{F_1, F_2, \dots\}$ is a set of aggregate functions where $F_i \in \mathcal{F}$ and $|\mathbf{F}| = |\mathbf{V}|$; and with slight abuse of notation, $\mathbf{F}(\mathbf{V})$ denotes $F_1(V_1), F_2(V_2), F_3(V_3) \dots$

Tabular representation of a pivot table. For $T(\mathbf{F}(\mathbf{V}), \mathbf{G})$, with $|\mathbf{G}| \geq 2$, we fix as row-groups and column-groups (Fig. 2) two non-empty sets $\mathbf{R}, \mathbf{C} \subset \mathbf{G}$, respectively, where $\mathbf{R} \cup \mathbf{C} = \mathbf{G}$, $\mathbf{R} \cap \mathbf{C} = \emptyset$.

Example 2.2. *Fig. 5(c) represents a possible tabular representation for the pivot table:* `SELECT COUNT(ID) FROM D GROUP BY Gender, Degree, Department`. *Here,* $\mathbf{R}$=`{Gender}` *and* $\mathbf{C}$=`{Degree, Dept.}`

Pivot table canonicalization. The above mechanism allows structurally different tabular representations of semantically equivalent tables. However, we only care about the semantics of a table, not its tabular orientation. Thus, we canonicalize a pivot table $T(\mathbf{F}(\mathbf{V}), \mathbf{G})$ by lexicographically sorting $\mathbf{F}(\mathbf{V})$ and $\mathbf{G}$ to obtain $\mathbf{F}(\mathbf{V})_{\leq}$ and $\mathbf{G}_{\leq}$, respectively, and derive the canonical pivot table $T(\mathbf{F}(\mathbf{V})_{\leq}, \mathbf{G}_{\leq})$. Furthermore, we obtain a *canonical tabular representation* of a pivot table $T(\mathbf{F}(\mathbf{V}), \mathbf{G})$ by assigning to the row-groups ($\mathbf{R}_{\leq}$) the first $\lceil \frac{|\mathbf{G}|}{2} \rceil$ elements of $\mathbf{G}_{\leq}$ and the remaining elements to the column-groups ($\mathbf{C}_{\leq}$). This process ensures that pivot tables remain *organization-invariant* (e.g., transpose-invariant), i.e., $T(\mathbf{F}(\mathbf{V}), \mathbf{G})$ and all its variants derived from different permutations of $\mathbf{F}(\mathbf{V})$ and $\mathbf{G}$ result in an identical canonical tabular representation.

Example 2.3. *The canonical pivot table for Fig. 5(c) is:* `SELECT COUNT(ID) FROM D GROUP BY Degree, Department, Gender`. *The canonical tabular representation is obtained by setting* $\mathbf{R}_{\leq}$ = ⟨`Degree, Dept.`⟩ *and* $\mathbf{C}_{\leq}$ = ⟨`Gender`⟩, *which is simply the transpose of Fig. 5(c).*

Based on the desiderata of Section 2.2, we set our goal to find a bounded sized (D4-I) set of pivot tables such that the overall utility (D1 & D2) of the pivot tables are maximized while the set of pivot tables meet the minimum diversity requirement (D3).

Problem 2.1 (Recommending a set of Pivot Tables). *Given (i) a set of possible pivot tables* $\mathcal{T}_{\mathbf{A}}$ *over a dataset* D *with attributes* $\mathbf{A}$, *(ii) a function* $Utility : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$ *that returns the utility of a pivot table* $T \in \mathcal{T}_{\mathbf{A}}$, *(iii) a function* $Diversity : 2^{\mathcal{T}_{\mathbf{A}}} \mapsto [0, 1]$ *that returns the diversity of a set of pivot tables* $\mathbf{T} \subseteq \mathcal{T}_{\mathbf{A}}$, *(iv) a budget* $k \in \mathbb{N}^+$, *and (v) a threshold* $\theta \in [0, 1]$, *find a set of pivot tables* $\mathbf{T}^* \subseteq \mathcal{T}_{\mathbf{A}}$ *s.t:*

$$
\begin{aligned}
&\textbf{\textit{(objective)}} && \mathbf{T}^* = \arg\max_{\mathbf{T} \subseteq \mathcal{T}_{\mathbf{A}}} \textstyle\sum_{T \in \mathbf{T}} Utility(T), \\
&\textbf{\textit{(size constraint)}} && |\mathbf{T}^*| \leq k, \textit{ and} \\
&\textbf{\textit{(diversity constraint)}} && Diversity(\mathbf{T}^*) \geq \theta
\end{aligned}
$$

Problem 2.1 balances utility and diversity by maximizing utility while putting a constraint on diversity. Other variants of this problem are possible such as maximizing a linear combination of the objective and the diversity constraint. More details are in Section 5.

Adaptive recommendation of a set of pivot tables. In the adaptive version (D4-II), we discard the already explored pivot tables by the user $\mathbf{T}_u$ from the set $\mathcal{T}_\mathbf{A}$ to obtain $\mathcal{T}_\mathbf{A} - \mathbf{T}_u$. When the user highlights a data scope (D4-I) by specifying a subset of attributes $\mathbf{A}_u \subseteq \mathbf{A}$ they want to focus on, we set the possible pivot tables to $\mathcal{T}_{\mathbf{A}_u}$.

Considerations. Multiple aggregates within a pivot table is essentially equivalent to concatenating the corresponding single-aggregate pivot tables, i.e., $T(\mathbf{F}(\mathbf{V}), \mathbf{G}) \equiv \bigcup_{F,V \in \mathbf{F},\mathbf{V}} T(F(V), \mathbf{G})$. Therefore, for simplicity and to promote interpretability, we limit each pivot table to have exactly one aggregate. We use F and V to denote the aggregation function and attribute, respectively. We summarize the notations used in the rest of this paper in Table 6.

A note on generalizability. While our work focuses on recommending pivot tables in spreadsheet environments, the techniques can be generalized to recommend aggregate queries in relational databases. Pivot tables can be represented by SQL aggregate queries involving `Group-by`, allowing SAGE's adaptation in RDBMS.

3 Utility of a Pivot Table

Based on the desiderata of Section 2.2, a pivot table has high utility if it offers *insights* (D1) while being easily *interpretable* by humans (D2). Thus, we use insightfulness (Section 3.1) and interpretability (Section 3.2) as the two building blocks to model utility of a pivot table.

3.1 Insightfulness

Intuitively, an insightful pivot table must involve attributes that are *significant*, i.e., inherently interesting and relevant (Section 3.1.1). Furthermore, it should satisfy at least one of the following criteria: (1) provide high *informativeness* (Section 3.1.2), (2) highlight meaningful *trends* (Section 3.1.3), or (3) reveal *surprising* [16, 39, 65] findings (Section 3.1.4). We build on prior work [16, 24, 28, 36, 39, 70] that model insightfulness based on only statistical properties, but significantly extend it by taking a *semantics-aware* approach, enabled by LLMs [47].

3.1.1 Attribute significance. Typically, not all data attributes are of interest by humans. E.g., grouping data by `Name` is typically much less insightful than by `Gender`. However, semantic understanding is required to figure out attribute significance. To this end, we consult an LLM [47] to determine the significance of an attribute A. When attribute name is missing or semantically meaningless (e.g., "Column 1"), we first query an LLM to suggest appropriate names for attributes by providing it with a small sample of the data. LLM's semantic-reasoning capability allows us to achieve this

Symbol	Description
$D, \mathbf{A}, D[\mathbf{A}]$	Database, attributes, possible unique value combinations
$\mathbf{G}, \mathbf{R}, \mathbf{C}$	Grouping attributes, row-groups, column-groups; $\mathbf{R} \cup \mathbf{C} = \mathbf{G}$
F, V	Aggregation function and attribute
$T(F(V), \mathbf{G})$ or T	A pivot table for the query `SELECT` $F(V)$ `FROM` D `GROUP BY` $\mathbf{G}$
T_{r_i}/T^{c_j}	The row/column of T with row/column header r_i/c_j
$T^{c_i}_{r_i}$	The pivot table cell with row header r_i and column header c_j
n, m	The cardinality of $D[\mathbf{R}]$, $D[\mathbf{C}]$

Table 6. Table of notations. We use bold letters to denote sets.

without any domain-specific pre-configuration. To avoid noise, we employ multiple paraphrased prompts while querying the LLM. In this work, we condition the LLM to return a simple binary answer (yes $\rightarrow$ 1/no$\rightarrow$ 0). However, this component can be replaced by a domain-aware model that can return the likelihood of an attribute being significant for a specific context. We compute attribute significance of a pivot table T, $S_{\text{sig}} : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$ as follows:

$$S_{\text{sig}}(T) = \prod_{A \in \{V\} \cup \mathbf{G}} \text{Significance}(A) \tag{1}$$

Here, Significance : $\mathbf{A} \mapsto [0, 1]$ denotes the probability that an attribute $A \in \mathbf{A}$ is a significant attribute w.r.t human interest.

Example 3.1. *For Fig. 5 (d),* `Degree`, `Department`, *and* `Salary`, *all are significant attributes. Thus,* $S_{sig}(T) = 1 \times 1 \times 1 = 1$.

3.1.2 Informativeness. A statistical way to measure informativeness within data is to measure spread of the values. Intuitively, if values in a pivot table deviate from each other significantly, the "entropy" is high, and so is the informativeness. In this work, we use *deviation* across different groups to model informativeness. Unlike prior work [24, 36, 70] that only consider a two-group setting (Male vs Female), we consider a multi-group setting.

Given a database D and a pivot table T with row-groups $\mathbf{R}$ and column-groups $\mathbf{C}$, let $D[\mathbf{R}]$ and $D[\mathbf{C}]$ be the set of row and column headers for T, respectively. E.g., for Fig. 5 (c), the row headers are {`Male`, `Female`} and the column headers are {(`BS`, `IT`), (`BS`, `Sales`), (`MS`, `IT`), (`MS`, `Sales`), (`PhD`, `IT`), (`PhD`, `Sales`)}. We use T_{r_i} (T^{c_i}) to denote the row (column) of T with row (column) header r_i (c_i). We use n and m to denote the number of rows $|D[\mathbf{R}]|$ and columns $|D[\mathbf{C}]|$ in T, respectively. We compute the row-wise and column-wise informativeness scores $S^{\text{row}}_{\text{inf}}$ and $S^{\text{col}}_{\text{inf}}$ as follows:

$$S^{\text{row}}_{\text{inf}}(T) = \frac{1}{\binom{n}{2}} \sum_{r_i, r_j \in D[\mathbf{R}] \text{ s.t } i<j} \frac{||T_{r_i}, T_{r_j}||_2}{\gamma \cdot m} \qquad S^{\text{col}}_{\text{inf}}(T) = \frac{1}{\binom{m}{2}} \sum_{c_i, c_j \in D[\mathbf{C}] \text{ s.t } i<j} \frac{||T^{c_i}, T^{c_j}||_2}{\gamma \cdot n}$$

Here, γ is a normalization parameter, set to $\max(T) - \min(T)$, ensuring that $S^{\text{row}}_{\text{inf}}$ and $S^{\text{col}}_{\text{inf}}$ are bounded between 0 and 1. Also note that while we use Euclidean distance (L_2 distance), any other distance function such as L_1 distance can be used here. We compute the informativeness score $S_{\text{inf}} : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$ by taking the maximum of the row-wise and column-wise informativeness scores:

$$S_{\text{inf}}(T) = \max(S^{\text{row}}_{\text{inf}}(T), S^{\text{col}}_{\text{inf}}(T)) \tag{2}$$

Example 3.2. *We first compute the pairwise distances along the rows of Fig. 5 (d):* $||T_{BS}, T_{MS}||_2 = 141.4K$, $||T_{BS}, T_{PhD}||_2 = 761.6K$, *and* $||T_{MS}, T_{PhD}||_2 = 632.5K$. *We normalize using* $\gamma = 900K - 100K = 800K$ *and* $m = 2$, *resulting in normalized distances of* $[0.088, 0.476, 0.395]$. *Taking an average gives us* $S^{row}_{inf}(T) = 0.32$. *We similarly compute* $S^{col}_{inf}(T) = 0.22$ *and obtain* $S_{inf}(T) = \max(0.32, 0.22) = 0.32$.

3.1.3 Trend. Trends observed in a pivot table provide insights. However, the degree of insightfulness hinges on two key factors: the *magnitude* of the trend metric and how *atypical* or rare it is. For instance, a positive correlation between income and years of service is generally expected—employees with longer tenures typically earn more. In contrast, a trend showing that new hires earn more on average than long-serving employees contradicts this expectation and thus is particularly insightful.

We use two metrics to quantify the magnitude of a trend: *correlation* and *ratio*. Furthermore, to assess the degree of a trend's rarity, we query an LLM, which is aware of a broader semantic context. Thus, our definition of the *trend score* for a pivot table combines (1) purely statistical insights, reflected in high correlation and consistent ratio across pivot table values and (2) semantic insights, captured through the LLM's assessment of the trend's atypicality.

Correlation. We use $\rho_{i,j}$ to denote the Pearson correlation coefficient between the rows T_{r_i} and T_{r_j}. Since consulting LLMs is costly, we only consider significant correlations and require the magnitude to be at least τ_ρ, a customizable threshold parameter, with a default value of 50%. The indicator function $[\![|\rho_{i,j}| \geq \tau_\rho]\!]$ denotes if the correlation between the rows T_{r_i} and T_{r_j} is significant. We compute the row-wise correlation-trend score $S^{\text{row}}_{\text{cor}} : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$ as follows:

$$S^{\text{row}}_{\text{cor}}(T) = \frac{1}{\binom{n}{2}} \sum_{r_i, r_j \in D[\mathbf{R}] \text{ s.t. } i<j} |\rho_{i,j}| \cdot [\![|\rho_{i,j}| \geq \tau_\rho]\!] \cdot \widetilde{Pr}_{\text{cor}}(r_i, r_j)$$

Here, $\widetilde{Pr}_{\text{cor}}(r_i, r_j)$ is the likelihood of *not* observing a high correlation between the groups r_i and r_j determined by an LLM. We prompt an LLM "In a five-point scale from *very likely* to *very unlikely*, how likely is it that the <$F(V)$> for <r_i> and <r_j> are <positively/negatively> correlated across <$D[\mathrm{C}]$>?", where each part within <> is replaced with actual values such as "Average Salary" for <$F(V)$>. We then map the LLM-provided likelihood to a numerical value using the mappings: *very likely* $\rightarrow$ 20%, *likely* $\rightarrow$ 40%, *neutral* $\rightarrow$ 60%, *unlikely* $\rightarrow$ 80%, and *very unlikely* $\rightarrow$ 100%. The intuition behind this *inverse* mapping is that the more unlikely it is to observe a trend, the more insightful it is. We compute $S^{\text{col}}_{\text{cor}}(T)$ similarly and set the correlation-trend score $S_{\text{cor}}(T) = \max(S^{\text{row}}_{\text{cor}}(T), S^{\text{col}}_{\text{cor}}(T))$.

Example 3.3. *In Fig. 5 (d), T_{BS} and T_{PhD} exhibit a positive correlation of 98%, which is likely (from LLM consultation), leading $\widetilde{Pr}_{cor}(BS, PhD)$ to be* 40%. *The correlation between T_{BS} and T_{MS} is 100% (very likely); and T_{MS} and T_{PhD} is 100% (likely). Since all correlations meet the threshold 50%, we compute $S^{row}_{cor}(T) = (0.98 \times 0.4 + 1.0 \times 0.2 + 1.0 \times 0.4)/3 = 0.33$. The column-wise correlation-trend score $S^{col}_{cor}(T) = (0.98 \times 0.4)/1 = 0.39$. Thus, $S_{cor}(T) = \max(0.33, 0.39) = 0.39$.*

Ratio. Since correlation fails to capture the relative *magnitude*, we use *ratio trends* based on *persistent* ratios between two groups, e.g., T_{PhD} earning at least 5× more than T_{BS} across all departments. Like correlation, LLMs inform us the rarity of ratio trends. We compute the row-wise ratio-trend score $S^{\text{row}}_{\text{ratio}} : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$ as follows:

$$S^{\text{row}}_{\text{ratio}}(T) = \frac{1}{\binom{n}{2}} \sum_{r_i, r_j \in D[\mathbf{R}]} (1 - \frac{1}{\pi_{i,j}}) \cdot [\![\pi_{i,j} \geq \tau_\pi]\!] \cdot \widetilde{Pr}_{\text{ratio}}(r_i, r_j)$$

Here, $\pi_{i,j}$ denotes the minimum element-wise ratio between T_{r_i} and T_{r_j}, i.e., the smallest factor by which any value in T_{r_i} exceeds its corresponding value in T_{r_j}. To reduce LLM consultation cost, we use a threshold τ_π and require $\pi_{i,j}$ to be at least τ_π before consulting an LLM. While we set $\tau_\pi = 2.0$, it is a customizable parameter (but must be ≥ 1). $\widetilde{Pr}_{\text{ratio}}(r_i, r_j)$ denotes the LLM-provided likelihood of *not* observing the ratio trend between T_{r_i} and T_{r_j}. Note that for any i and j, at most one of $\pi_{i,j}$ or $\pi_{j,i}$ can contribute to this score, hence we fix the scaling factor to $\binom{n}{2}$. We normalize the trend magnitude by subtracting the inverse of $\pi_{i,j}$ from 1, so that larger ratios yield higher scores. We compute the column-wise ratio-trend score $S^{\text{col}}_{\text{ratio}}(T)$ similarly and set the ratio-trend score $S_{\text{ratio}}(T) = \max(S^{\text{row}}_{\text{ratio}}(T), S^{\text{col}}_{\text{ratio}}(T))$.

Example 3.4. *In Fig. 5 (d), the minimum ratio between T_{MS} and T_{BS} is 1.5; between T_{PhD} and T_{BS} is 4.0; and between T_{PhD} and T_{MS} is 2.0. After applying the threshold $\tau_\pi = 2.0$, the ratio trends for (T_{PhD}, T_{MS}) and (T_{PhD}, T_{BS}) are retained. The LLM returns the likelihoods:* `[Very Unlikely, Unlikely]` $\rightarrow$ $[1.0, 0.8]$ *for these two trends. This gives us the row-wise ratio-trend score $S^{row}_{ratio}(T) = (3/4 \times 1.0 + 1/2 \times 0.8)/3 = 0.37$. For the column-wise ratio-trend score, no pair satisfies the threshold requirement and thus the score is* 0.0. *Therefore, the ratio-trend score $S_{ratio}(T) = \max(0.37, 0.0) = 0.37$.*

Finally, we compute the trend score by taking the maximum of the correlation-trend and ratio-trend scores:

$$S_{\text{trend}}(T) = \max\left(S_{\text{cor}}(T),\ S_{\text{ratio}}(T)\right) \tag{3}$$

Example 3.5. *For the pivot table of Fig. 5 (d), we computed the correlation-trend score as 0.39 in Example 3.3 and the ratio-trend score as 0.37 in Example 3.4 Thus,* $S_{trend}(T) = \max(0.39, 0.37) = 0.39$.

3.1.4 Surprise. Surprising values or *outliers* often indicate insights. E.g., in Fig. 5 (d), $T^{\text{IT}}_{\text{PhD}}$ is exceptionally high (900K) in its column. While such outliers can be insightful, not all are. Some, like this one, are expected: IT is high-paying, and PhDs earn more. In contrast, an unusually high $T^{\text{Sales}}_{\text{BS}}$ would be surprising, and thus insightful. Beyond simply identifying outliers, we incorporate the unexpectedness of observing outliers using LLM's semantic knowledge. We compute the row-wise surprise score $S^{\text{row}}_{\text{sur}} : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$ as follows:

$$S^{\text{row}}_{\text{sur}}(T) = \frac{1}{n}\sum_{r_i \in D[\mathbf{R}]} OutlierScore(T_{r_i})$$

$$\text{where, } OutlierScore(T_{r_i}) = \begin{cases} 1 - \frac{\sum_{c \in O_{r_i}} \widetilde{Pr}_{\text{outlier}}(r_i, c, T^c_{r_i})}{|O_{r_i}|+1}, & \text{if } |O_{r_i}| > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$O_{r_i} = \{c_j \in D[\mathbf{C}] \; s.t. \; |T^{c_j}_{r_i} - \mu(T_{r_i})| \geq \tau_O \cdot \sigma(T_{r_i})\}$$

O_{r_i} is the set of column headers for each outlier in T_{r_i}. E.g., $O_{\text{PhD}} = \{\text{IT}\}$, if $900K$ is an outlier for the row T_{PhD}. We ensure that the score increases with the number of outliers by subtracting the inverse of their count from 1. Since even a single outlier matters, we add 1 to the denominator to ensure that even one outlier results in a score multiplier of 0.5. The threshold τ_O is set to 4, since, assuming normal distribution, 99.99% of the population is expected to lie within 4 standard deviations from the mean [33][4]; and anything outside this range is an outlier. $\widetilde{Pr}_{\text{outlier}}(r_i, c, T^c_{r_i})$ denotes the LLM-obtained likelihood of $T^c_{r_i}$ *not* being an outlier w.r.t T_{r_i}. We compute $S^{\text{col}}_{\text{sur}}(T)$ similarly and compute the surprise score:

$$S_{\text{sur}}(T) = \max(S^{\text{row}}_{\text{sur}}(T), S^{\text{col}}_{\text{sur}}(T)) \tag{4}$$

Computing Insightfulness. A pivot table is insightful if it exhibits any of the characteristics: informativeness, trend, or surprise. Thus, we take the *maximum* of S_{inf}, S_{trend}, and S_{sur} (Equations 2, 3, and 4) to compute $Insightfulness : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$. To prioritize pivot tables involving significant attributes, we scale this score by S_{sig}.

$$Insightfulness(T) = S_{\text{sig}}(T) \cdot \max\left(S_{\text{inf}}(T),\ S_{\text{trend}}(T),\ S_{\text{sur}}(T)\right) \tag{5}$$

Example 3.6. *For the pivot table of Fig. 5 (d), the attribute significance score* $S_{sig}(T) = 1$ *(Example 3.1). The informativeness score* $S_{inf}(T) = 0.32$ *(Example 3.2), the trend score* $S_{trend}(T) = 0.32$ *(Example 3.5), and since there is no outlier in the pivot table, the surprise score* $S_{sur}(T) = 0.0$. *Thus* $Insightfulness(T) = 1 \times \max(0.32, 0.39, 0.0) = 0.39$.

Remark 1. Presence of outliers can inflate the normalization factor γ (Section 3.1.2), causing S_{inf} to shrink significantly due to the compression of the value range of non-outliers. However, such cases typically yield a high S_{sur}, complementing the low S_{inf}.

Remark 2. The Max operator used in Equation 5 may favor the component with the highest mean, without considering their distributions—a common challenge in multi-criteria decision making [53]. However, standardizing all components to the same mean can distort their relative importance. E.g., if all pivot tables have uniformly low informativeness (with low variance) but uniformly high trend (with low variance), normalization would artificially inflate informativeness and suppress trend, leading to incorrect selections. The Max operator retains the strongest signal without distorting component distributions through forced standardization. Nevertheless, SAGE is agnostic to the choice of the formula to compute *Insightfulness* and any linear formula would work.

[4]Pivot tables may not be normally distributed. This heuristic ensures simplicity, but can be replaced with other methods.

3.2 Interpretability

A key measure of a pivot table's utility is *interpretability*, which takes into account the cognitive constraints of humans. Consider the aggregate `SUM(AGE)`, row group `Degree` (3 values), and column groups `Employed_Year` (14 values) and `Department` (2 values), which results in an 84-cell pivot table. Its interpretability suffers due to three reasons: (1) High sparsity resulting from many value combinations yielding empty sets—e.g., no MS hire in 2011 for the IT department (Section 3.2.1), (2) Semantically invalid aggregate `SUM(AGE)` (Section 3.2.2), and (3) Excessive #columns ($14 \times 2 = 28$) from fine-grained yearly grouping, compromising conciseness (Section 3.2.3). We proceed to describe three desirable interpretability properties.

3.2.1 Density. Since each pivot table cell maps to a data subset under a specific value combination (e.g., MS hires in IT in 2011), empty subsets can occur. When aggregated, these empty subsets produce *null* values. However, excessive nulls hinder interpretability [1], as humans struggle to draw insights from sparse tables. This motivates a key interpretability criterion: high *density*. We compute the *density score* $S_{\text{den}} : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$ as follows:

$$S_{\text{den}}(T) = \frac{\sum_{(r_i, c_j) \in D[\mathbf{R}] \times D[\mathbf{C}]} [\![T_{r_i}^{c_j} \neq null]\!]}{n \cdot m} \tag{6}$$

Example 3.7. *The pivot table of Fig. 5 (d) has 3 rows and 2 columns (total 6 cells) and no null values. Hence,* $S_{den}(T) = \frac{6}{2 \times 3} = 1.0$.

3.2.2 Semantic validity. Row and column headers in a pivot table represent unique values of the grouping attributes in $\mathbf{G}$. For interpretability, these headers must be semantically meaningful [8]. E.g., `Degree` with values `{MS, BS, PhD}` is interpretable, while a functionally equivalent `Degree_ID` with values `{1, 2, 3}` is not, due to the lack of direct semantic meaning [23]. Similarly, the aggregate function F must be semantically valid w.r.t V: `AVG` is semantically valid for `AGE`, but `SUM` is not [43]. Though intuitive for humans, such judgments require domain knowledge. Thus, we leverage an LLM to mimic human reasoning and assess aggregation semantics. We define the *semantic validity score* of $T(F(V), \mathbf{G})$ based on two criteria: (1) whether the data types of attributes in $\mathbf{G}$ are textual, and (2) the extent to which F is semantically valid w.r.t V.

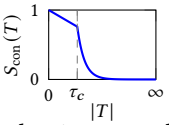
$$S_{\text{sem}}(T) = \frac{|\{A \in \mathbf{G} \text{ s.t } \texttt{DataType}(A) \text{ is Text}\}|}{|\mathbf{G}|} \cdot Pr_{\text{agg}}(F, V) \tag{7}$$

We compute $Pr_{\text{agg}}(F, V)$ from an LLM-generated ranking of $F \in \mathcal{F}$ based on its semantic validity w.r.t V. We score the best function 1.0, the next 0.8, and so on, which ensures that $S_{\text{sem}}(T) \in [0, 1]$.

Example 3.8. *The pivot table of Fig. 5 (d), contains only textual headers. LLM responds to our prompt "Rank the functions* `COUNT`, `AVG`, `SUM`, `MIN`, *and* `MAX`, *based on their appropriateness for analyzing* `Salary`*" with {*`AVG`, *...}. Thus,* $S_{sem}(T) = \frac{2}{2} \times 1.0 = 1.0$.

3.2.3 Conciseness. While multiple grouping attributes may enhance insightfulness, too many cells reduce comprehensibility [57], and, thus, interpretability. To model this, we define conciseness score $S_{\text{con}} : \mathcal{T}_{\mathbf{A}} \mapsto [0, 1]$ using a piecewise function [25]:

$$S_{\text{con}}(T) = \begin{cases} 1 - z|T|, & \text{if } |T| \leq \tau_c \\ (1 - z\tau_c) e^{-\lambda(|T| - \tau_c)}, & \text{if } |T| > \tau_c \end{cases} \tag{8}$$



Here, $|T|$ denotes the number of cells in T. This formula captures the intuition that interpretability declines gradually at first, but drops sharply once $|T|$ exceeds a threshold τ_c, set to 16. We apply a 3% linear decrease ($z = 0.03$) until $|T|$ exceeds τ_c, and an exponential decay at a rate of 50% ($\lambda = 0.5$)

beyond that. This is grounded in cognitive load theory [7, 50], which states that performance declines sharply when cognitive demand exceeds working-memory capacity.

Example 3.9. *The pivot table of Fig. 5 (d) has 6 cells. Since 6<16, we compute the linear part:* $1-0.03\times6=0.82$. *Thus,* $S_{con}(T)=0.82$.

Computing Interpretability. Unlike insightfulness, where a strong signal from *any* single type of insight is sufficient, interpretability demands that *all* criteria be met simultaneously. Therefore, we compute $Interpretability : \mathcal{T}_{\mathrm{A}} \mapsto [0,1]$ as the average of the three scores: S_{den}, S_{sem}, and S_{con} (Equations 6, 7, and 8):

$$Interpretability(T) = \frac{S_{\mathrm{den}}(T) + S_{\mathrm{sem}}(T) + S_{\mathrm{con}}(T)}{3} \tag{9}$$

Example 3.10. *For Fig. 5 (d), we obtained values for* S_{den} *(T),* S_{sem} *(T), and* S_{con} *(T) to be 1.0, 1.0, and 0.82, respectively, in the previous examples. Thus,* $Interpretability(T)$ $=(1.0+1.0+0.82)/3=0.94$.

3.3 Computing Utility

We now define $Utility : \mathcal{T}_{\mathrm{A}} \mapsto [0,1]$ of a pivot table T by combining *Insightfulness* (Eq. 5) and *Interpretability* (Eq. 9). To balance their contributions, we introduce a tunable parameter α, set to 0.5 by default to give equal weight to both components. However, α can be adjusted to reflect application-specific preferences.

$$Utility(T) = \alpha \cdot Insightfulness(T) + (1-\alpha) \cdot Interpretability(T)$$

Example 3.11. *For Fig. 5 (d), the* $Insightfulness$ *and* $Interpretability$ *scores are 0.39 and 0.94, respectively. Thus,* $Utility(T) = 0.5 \times 0.39 + 0.5 \times 0.94 = 0.67$.

Need for a new Utility Model. Our *Utility* model significantly extends prior work that address related problems [14, 24, 28, 36, 70, 76]. *First*, prior work ignore the *semantic validity* of the summary *structure*, which directly affects interpretability (e.g., `SUM(Birth_Year)` or `AVG(Zip_Code)` are invalid; `{Val_1, Val_2}` is less interpretable than `{Large, Small}`; and `Gender` is a more interesting grouping attribute than `Employee ID`). *Second*, prior work rely solely on statistical signals. For example, SeeDB [70] uses an EMD-based deviation metric, which is just one of the seven components of our *Utility* model, while others [28, 76] use value-proportion and statistical-significance tests. Moreover, they do not support multi-group reasoning, required in our setting involving pivot tables. But most importantly, they lack *semantic awareness* to validate an apparent data insight. For instance, "a significant deviation in average height between toddlers and adults" will incur a high score based on their metrics. However, such deviation is trivially expected. In contrast, even a moderate "deviation in average height across socio-economic groups" indicates an interesting insight. *Third*, these methods ignore quality factors, such as `NULL` values in some groups in the data summary (which yields empty bars for visualization). In general, most prior work do not incorporate various aspects of interpretability in their utility models. The above shortcomings of the prior work motivates a *semantics-aware* and *interpretability-focused* Utility model, like ours, that identifies insights that are meaningful in real-world contexts, not merely statistically distinct.

4 Diversity in a Set of Pivot Tables

While utility quantifies the goodness of a single pivot table in isolation, *diversity* captures how well a *set* of pivot tables, *collectively*, provide complementary and unique perspectives on the data (D3). High diversity in a set of pivot tables is achieved when the pivot tables are distant from each other with respect to data coverage and the insights they provide.

Diversity. Following Max-Min diversification [2], we define diversity of a set of pivot tables $\mathbf{T} = \{T_1, T_2, \dots\}$ by the smallest pairwise distance between $\mathbf{T}$'s elements. More formally, given a symmetric distance function $dist : \mathcal{T}_\mathbf{A} \times \mathcal{T}_\mathbf{A} \mapsto [0, 1]$, we define $diversity : 2^{\mathcal{T}_\mathbf{A}} \mapsto [0, 1]$ of a set of pivot tables $\mathbf{T} \subseteq \mathcal{T}_\mathbf{A}$ as follows:

$$Diversity(\mathbf{T}) = \min_{T_i, T_j \in \mathbf{T} \text{ s.t. } i<j} dist(T_i, T_j)$$

Distance between pivot tables. A simple heuristic to model $dist$ is the degree of disjointness between the attributes that define the pivot-table queries—if two pivot tables operate on the same set of attributes, the distance is 0; for completely disjoint set of attributes, the distance is 1. However, this heuristic fails to account for the *structural semantics* of the pivot-table queries and the *content semantics* of the data in the pivot tables. To this end, we employ a semantics-preserving embedding function $E : \mathcal{T}_\mathbf{A} \mapsto [-1, 1]^p$, which maps a pivot table $T \in \mathcal{T}_\mathbf{A}$ to a p-dimensional vector. Then, we compute distances between pivot tables in this embedding space:

$$dist(T_1, T_2) = \frac{1 - cosine\_similarity(E(T_1), E(T_2))}{2}$$

Here, $cosine\_similarity : \mathbb{R}^p \times \mathbb{R}^p \longrightarrow [-1, 1]$ is a widely used measure for comparing embeddings [5, 60]. We divide by 2 to achieve normalization s.t $dist(T_1, T_2) \in [0, 1]$.

Pivot-table embedding. Pivot-table embedding should capture both the syntactic and semantic characteristics of the pivot table. To this end, we combine the query embedding E_Q and the content embedding E_C through concatenation, i.e., $E(T) = [E_Q(T); E_C(T)]$. Concatenating embeddings is a widely used technique in machine learning, natural language processing, and multi-modal learning [84].

Query embedding. Queries define the structural intent of a pivot table. While they reference data attributes and are aware of the schema, they are agnostic to the pivot table's content. Thus, the same pivot-table query over different database instances with an identical schema should yield the same query embedding. Effective query embeddings must also capture the semantics of the attribute names and reflect the query semantics. E.g., `GROUP BY Income` and `GROUP BY Salary` are semantically similar and should therefore be close in the query-embedding space. To this end, we use T5 [59], a natural-language encoder fine-tuned over a Text-to-SQL dataset [82], to obtain the query embedding $E_Q : \mathcal{T}_\mathbf{A} \mapsto [-1, 1]^{1024}$.

Content embedding. The content embedding must capture both the statistical and distributional properties of the pivot-table data and structural relationships among its attributes and tuples. In this work, we leverage TAPEX [45], a pre-trained encoder trained on sentence-table pairs, which is designed to understand both the structure and content of tabular data. This results in a content embedding $E_C : \mathcal{T}_\mathbf{A} \mapsto [-1, 1]^{1024}$.

5 The SAGE Algorithm

We now present a solution to Problem 2.1, which is to recommend a k-budgeted set of pivot tables with maximum utility under a diversity constraint. This is an instance of the NP-hard Maximum Weight Independent Set (MWIS) problem [61] with complexity $O(|\mathcal{T}_\mathbf{A}|^k)$ in the size of the search space ($\mathcal{T}_\mathbf{A}$ in our case). Furthermore, $\mathcal{T}_\mathbf{A}$ grows combinatorially with $|\mathbf{A}|$ (the number of data attributes), leading to an exponential growth. Approximation techniques exist for special cases of MWIS and related problems [13, 42, 52, 61] and alternative formulations are possible such as incorporating diversity into the objective or solving the dual version that maximizes diversity [2] under a utility constraint. However, in SAGE, we adopt a simple greedy approach (Section 5.2) for

two key reasons: (1) Interactive response time is desirable for our problem settings and a greedy approach achieves linear time complexity of $O(|\mathcal{T}_{\mathbf{A}}|)$. (2) As we demonstrate in Section 6, our greedy approach works remarkably well in practice over real-world datasets, almost always matching the exact solution.

5.1 Optimizations

The linear-time complexity of the greedy approach is still prohibitive for practical use for two reasons: (1) Computing the *Utility* of candidate pivot tables requires their materialization, which is computationally expensive—especially since $\mathcal{T}_{\mathbf{A}}$ grows exponentially with $|\mathbf{A}|$. (2) Some components of *Utility* relies on the LLM, whose inference latency is a bottleneck [69, 71]. To address these efficiency challenges (D5), we introduce two offline optimizations: (1) *Candidate pruning*: based on the query structure of the pivot tables, we eliminate potentially low-utility ones before materialization, thereby significantly reducing the search space and avoiding many materializations (Section 5.1.1). (2) A lightweight *proxy model*: tailored to the dataset, it approximates LLM inferences efficiently (Section 5.1.2).

5.1.1 Pruning. To prune pivot tables likely to yield low utility without materialization, we leverage three components of *Utility* (Section 3.3): C-I: *attribute significance* (Section 3.1.1) used in *Insightfulness* (Eq 5), C-II: *semantic validity* (Section 3.2.2), and C-III: *conciseness* (Section 3.2.3) used in *Interpretability* (Eq 9). These choices are motivated by computational efficiency, as these components require knowledge of only the pivot-table query, which defines the structure of the pivot table—such as which attributes are used, number of cells, etc.—and do not require a full materialization over the dataset.

While *Insightfulness* selects only the maximum among multiple components, *Interpretability* averages out three components—two of which are C-II and C-III above. This enables effective pruning of pivot tables that already show low scores for C-II and/or C-III. Furthermore, since C-I interacts in a multiplicative way with other components of *Insightfulness* (Eq 5), a low value will inevitably result in low *Insightfulness*, making it an ideal choice. Our pruning algorithm works as follows: we evaluate scores for the above three components and over-approximate *Utility* (Section 3.3) by assigning maximum possible values for all other components whose values are unknown. Through this conservative estimate, we discard candidates with a score below a threshold (a system parameter set to 0.5).

5.1.2 LLM-proxy. Recall that the computation of pivot-table utility requires LLM inferences (Section 3), and querying LLMs is time-consuming in practice. Even if we cache the LLM responses, any change in the underlying data or user-specified parameters (D4) would require LLM re-consultation. To expedite this process, we train a cheap but significantly faster decision-tree classifier to mimic LLM behavior, serving as an "LLM-proxy" during the online phase of SAGE (Algorithm 1). We train the classifier over 10,000 LLM prompt-response pairs, where we generate potential LLM-queries based on the dataset. The proxy model is a simple predictive model to answer the fixed-template questions discussed in Section 3. No re-training of this proxy model is required as long as the data distribution and overall trends in the dataset remain unchanged. While training with additional prompt-response pairs or employing more complex classifiers can further improve the accuracy of this proxy model, we show empirically in Section 6.1 that 10,000 prompt-response pairs already provide reasonable accuracy for practical applications.

5.2 SAGE: Greedy Algorithm

Algorithm 1 shows the SAGE workflow. Line 1 denotes the offline pruning step and lines 2–4 show the steps for LLM-proxy training. The online phase is shown in lines 5–14. The pivot tables that survive the pruning step are materialized in line 6, we then compute their *Utility* scores (line 7) and sort the pruned candidate set of pivot tables by descending order of utility (line 8). The

Algorithm 1: SAGE algorithm

Input : Database D,
Unmaterialized candidate pivot table set $\mathcal{T}_{\mathbf{A}}$,
Diversity threshold θ,
The desired number of pivot tables k

Output : A high-utility set $\mathbf{T} \subseteq \mathcal{T}_{\mathbf{A}}$ s.t. $|\mathbf{T}| \leq k$ and $Diversity(\mathbf{T}) \geq \theta$

/* **Offline phase**: executed once for each dataset. Re-executed if the schema changes or major change happens in the data distribution. */

/* Prune candidate set based on pivot-table structure (Section 5.1.1) */

1 $\mathcal{T}_{\mathbf{A}}^{Pr} \leftarrow Prune(\mathcal{T}_{\mathbf{A}})$

/* Building LLM-Proxy (Section 5.1.2) */

2 $Q \leftarrow$ list of questions about D /* Generate prompts for the LLM */

3 $R \leftarrow$ LLM-Response(Q) /* Get responses from the LLM */

4 $LLM\text{-}Proxy \leftarrow Train(Q, R)$ /* Train a proxy prediction model */

/* **Online phase**: executed whenever the data changes. */

/* Materialize and compute utility based on pivot-table contents */

5 **foreach** $T \in \mathcal{T}_{\mathbf{A}}^{Pr}$ **do**

6 Materialize T over D

7 Compute $Utility(T)$ /* Section 3.3 */

8 $\mathcal{T}_{\mathbf{A}}^{Pr} = sorted(\mathcal{T}_{\mathbf{A}}^{Pr})$ /* Sort by the descending order of Utility */

/* Greedy selection (Section 5.2) */

9 $\mathbf{T} \leftarrow \emptyset$ /* Initialize an empty set */

10 **while** $|\mathbf{T}| \leq k$ **do**

11 **foreach** $T \in \mathcal{T}_{\mathbf{A}}^{Pr}$ **do**

/* No pivot table in $\mathbf{T}$ is within θ distance away from T */

12 **if** $\nexists T' \in \mathbf{T}$ *s.t.* $dist(T, T') < \theta$ **then**

13 $\mathbf{T} \leftarrow \mathbf{T} \cup \{T\}$

14 **return** $\mathbf{T}$

greedy selection phase (lines 9–14) selects pivot tables greedily while ensuring that they satisfy the diversity constraint w.r.t the already selected ones in $\mathbf{T}$. If a candidate pivot table is at least θ away from all the previously selected tables, we include it to $\mathbf{T}$ (lines 12–13). The algorithm terminates when we have selected k pivot tables or no more candidates satisfy the diversity constraint.

5.3 SAGE$^+$: A Practical Variant

Even after the two optimizations, SAGE may not ensure interactive speed especially when the dataset has millions of tuples or hundreds of attributes, hurting it's practical adoption. To this end, we propose SAGE$^+$, a variant of SAGE that incorporates sampling and approximation techniques to ensure practically acceptable runtimes in an interactive setting, which is essential for spreadsheet environments. Specifically, to handle high-cardinality and high-dimensionality, it (i) uses 20% of the data samples when the dataset is large (e.g., contains 1M+ tuples or 100+ attributes) for candidate pivot-table generation, (2) uses 10% subsampling for approximating *Trend* (Section 3.1.3) and *Surprise* (Section 3.1.4) scores, and (iii) applies dimensionality reduction using Johnson-Lindenstrauss lemma [41], which preserves pairwise distances up to a distortion factor ratio $\epsilon = 0.2$, for approximating the *Informativeness* score (Section 3.1.2). Furthermore, SAGE$^+$ uses multiple cores (12 in our experiments) for parallelizing computations. Note that SAGE$^+$ does not improve the theoretical runtime complexity compared to SAGE; instead, it provides practical enhancements that ensure better suitability for practical use.

6 Experimental Results

We now present experimental results to demonstrate the efficacy of SAGE in practical settings to address the following research questions:

- (Q1) How do SAGE's runtime and recommendation quality compare quantitatively with those of existing methods? (Section 6.3)
- (Q2) What is the effect of the optimization techniques—pruning and LLM-proxy—on SAGE's runtime performance? (Section 6.4)
- (Q3) How well does SAGE scale with data growth? (Section 6.5)
- (Q4) How do key parameters (budget k and diversity threshold θ) influence the quality of SAGE recommendations? (Section 6.6)
- (Q5) How do SAGE recommendations qualitatively compare against commercial software and LLMs over real datasets and how does SAGE adapt based on user feedback? (Section 6.7)

6.1 Experimental Setup, Datasets, and Baselines

6.1.1 Setup. All experiments were run on machines with 256 GB RAM running Ubuntu 22.04 LTS with CPU 12 cores and GPU NVIDIA H100 96GB with CUDA 12.8. We implemented our solutions (available publicly [19]) with Python 3.10.3. For embeddings, we utilized TAPEX-large [49] and T5 trained by Spider [4]. We employed Llama-3-7B [46] as the LLM for semantic consultation.

6.1.2 Datasets. We used four real-world datasets with varying domains and sizes.

– **Marketing** [6] contains 2,240 tuples and 28 attributes (19 categorical and 9 numerical), capturing demographic and behavioral information about customers like marital status & purchase history.
– **Video** [67] comprises video game sales from various countries, platforms, and release years, containing 16,600 tuples across 11 attributes (7 categorical and 4 numerical).
– **House** [56] contains information about property sales, comprising 1,460 tuples and 81 attributes (58 categorical and 23 numerical).
– **CoverType** [10] is a large dataset with 581,000 tuples and 110 attributes, containing tree observations across a National Forest.

6.1.3 Baselines. Since no open-source tool or academic work directly addresses our problem, we include commercially available software as baselines, despite not being open-sourced.

Brute-Force considers all possible k-sized pivot table sets and follows an exhaustive approach to solve Problem 2.1. In the absence of ground truth, its results can be treated as the optimal solution.

Top-k ranks candidate pivot tables in descending order of their utility scores and selects the top-k, without accounting for diversity. For fair comparison, we applied our optimizations here.

LLM refers to Llama-3-8B-Instruct by Meta [46], a transformer-based large language model. We prompted it with a data sample and asked for interesting and diverse pivot tables in natural language.

DAISY [78] is a query recommendation system trained on crowdsourced "interesting" queries. Due to the original model and data not being available, we did our own implementation. We generated insightful pivot tables from Auto-Suggest [79], created negative training data by replacing attributes in the positive tables with random ones, and trained a binary classifier to distinguish them.

Microsoft Excel [48] is a widely used commercial spreadsheet software that offers built-in pivot table recommendations. We used the Windows version 2501.

PowerBI [22, 28], by Microsoft, is a business intelligence software that offers "quick insights" in various forms. For comparison, we only considered ones that align with our format of pivot table.

Google Sheets [35] is an online spreadsheet software known for easy collaboration, which recommends pivot tables. We used the browser version during February, 2025.

Dataset	Correlation	Ratio	Surprise
Marketing	89%	88%	65%
Video	92%	91%	61%
House	88%	87%	63%
CoverType	90%	90%	66%
Average	90%	89%	64%

Table 7. Likelihood prediction accuracy of the LLM-proxy model across datasets.

6.2 Offline Phase

As discussed in Section 5.1, SAGE relies on two offline optimizations: pruning (Section 5.1.1) and developing an LLM-Proxy model (Section 5.1.2). The LLM-Proxy model is composed of a decision tree whose parameters are learned from a training dataset of prompts and their corresponding answers retrieved from Llama-3-7B [46]. We generated 10,000 prompts—for each of correlation (Section 3.1.3), ratio (Section 3.1.3), and surprise (Section 3.1.4)—splitting the resulting datasets 80/20 into training and test sets. We configured the decision tree to have a maximum depth of 15 and used the Gini function as the splitting criterion. On average, across the four datasets, the LLM-Proxy model achieved an accuracy of 90%, 89%, and 64% for correlation, ratio, and surprise, respectively. Table 7 reports the accuracy of our proxy models across the three components for each dataset. The results are consistent across datasets, with surprise showing moderate accuracy, while both correlation and ratio achieve relatively high accuracy.

6.2.1 LLM-proxy training time. The training time for the LLM-proxy model was 2,742 seconds for surprise and 3,171 seconds for trend (correlation and ratio) for the marketing dataset; 1,404 seconds for surprise and 3,129 seconds for trend for the video dataset; 3,599 seconds for surprise and 3,087 seconds for trend for the house dataset; and 1,473 seconds for surprise and 3,569 seconds for trend for the CoverType dataset.

6.2.2 Pruning time. The offline pruning step takes 1.57 seconds for the house dataset (780,840 combinations), 0.04 seconds for the video dataset (2,145 combinations), 2.73 seconds for the marketing dataset (43,848 combinations), and 16.51 seconds for the CoverType dataset (2,785,120 combinations). Although the house dataset contains far more combinations than the marketing dataset, its pruning time is smaller. This is because the pruning process first computes attribute significance, and only if the attribute is considered interesting, it proceeds to compute interpretability. For the house dataset, only a small number of attributes are considered significant, resulting in far fewer computations for interpretability, and, therefore, a shorter pruning time. On average, the offline phase took about 92 minutes per dataset, which includes both pruning and LLM-Proxy training.

6.3 Contrasting against Baselines

Table 8 contrasts SAGE against the baselines across three datasets. Our primary metrics for comparison are *Utility* (Util) and *Diversity* (m-dist). However, since *Insightfulness* and *Interpretability* constitute *Utility* (Section 3), we report them for additional context. The results show that SAGE effectively balances utility and diversity, outperforming approaches like Top-k, which inherently lack diversity. For Marketing, SAGE achieves a strong overall performance, ranking either best or second-best in both metrics. Notably, when SAGE marginally loses in terms of diversity, that is usually complemented by significantly higher utility. For example, in Marketing, $k = 10$, LLM's diversity (0.20) is more than SAGE's diversity (0.11). However, LLM's utility (3.69) is about half of SAGE's utility (7.44). A similar situation is seen for $k = 3$ where GSheets yields only 30% of SAGE's utility. Recall that SAGE's goal is to just satisfy the diversity constraint, not maximize it.

		Marketing							Video							House						
		#PT	T(s)	Ins	Int	Util	Div m-dist	Div heatmap	#PT	T(s)	Ins	Int	Util	Div m-dist	Div heatmap	#PT	T(s)	Ins	Int	Util	Div m-dist	Div heatmap
k=3, θ=0.30 or 0.20	Top-k	3	147	2.86	2.23	**2.55**	0.39		3	64	1.99	1.27	**1.63**	0.06		3	2	2.26	2.13	**2.20**	0.09	
	DAISY	3	23	0.54	1.48	1.01	0.16		3	21	1.63	1.00	1.32	0.04		3	455	0.23	1.17	0.70	**0.44**	
	LLMs	3	15	0.16	1.75	0.96	0.16		3	25	0.01	0.58	0.29	0.32		3	30	0.07	0.55	0.31	0.06	
	PowerBI	3	9	0.91	0.94	0.93	0.41		3	6	0.91	0.94	0.93	**0.36**		3	15	0.50	1.76	1.13	0.36	
	GSheets	2	1	0.00	1.67	0.83	**0.56**		3	1	0.04	0.59	0.31	0.32		–	–	–	–	–	–	–
	Excel	3	1	0.29	2.42	1.35	0.14		3	1	0.14	2.16	1.15	0.21		3	1	0.00	1.87	0.93	0.31	
	SAGE	3	161	2.86	2.23	**2.55**	0.45		3	69	1.99	1.27	**1.63**	0.30		3	2	2.21	2.09	2.15	0.21	
	SAGE$^+$	3	23	2.76	2.23	2.50	0.32		3	17	1.47	1.63	1.55	0.30		3	2	2.09	2.06	2.07	0.22	
k=5, θ=0.30 or 0.20	Top-k	5	148	4.76	3.72	**4.24**	0.11		5	64	3.31	2.03	**2.67**	0.05		5	2	3.77	3.56	**3.66**	0.05	
	DAISY	5	23	0.54	2.51	1.52	0.16		5	21	2.57	1.67	2.21	0.04		5	455	0.23	1.84	1.04	0.29	
	LLMs	5	22	0.93	2.94	1.94	0.31		5	20	0.18	0.86	0.52	0.03		5	44	0.07	0.75	0.41	0.13	
	PowerBI	5	9	0.20	4.23	2.21	0.25		-	-	-	-	-	-	-	5	15	0.50	2.99	1.74	**0.36**	
	Excel	5	1	0.29	3.98	2.13	0.08		5	1	0.31	3.28	1.80	0.04		5	1	0.00	3.05	1.52	0.08	
	SAGE	5	161	4.76	3.70	4.23	**0.33**		5	69	3.31	2.03	**2.67**	**0.30**		5	2	3.42	3.52	3.47	0.20	
	SAGE$^+$	5	23	4.67	3.67	4.17	0.30		5	17	1.80	2.92	2.36	0.23		5	2	2.65	3.46	3.06	0.22	
k=10, θ=0.10	Top-k	10	147	9.53	7.44	**8.49**	0.02		10	64	4.06	5.50	**4.78**	0.05		10	2	7.57	7.02	**7.30**	0.05	
	DAISY	10	23	0.54	4.22	2.38	0.12		10	21	4.09	3.33	3.71	0.02		10	455	0.23	3.28	1.75	0.11	
	LLMs	10	32	1.70	5.67	3.69	**0.20**		10	28	0.74	2.24	1.49	0.04		10	43	0.26	1.72	0.99	0.02	
	PowerBI	9	9	0.20	4.88	2.54	0.09		-	-	-	-	-	-	-	10	15	0.50	5.99	3.24	**0.14**	
	Excel	7	1	0.29	5.30	2.79	0.08		9	1	0.31	4.71	2.51	0.01		7	1	0.00	3.97	1.98	0.08	
	SAGE	10	161	5.58	9.53	7.44	0.11		10	69	4.06	5.50	**4.78**	**0.11**		10	2	7.55	6.91	7.23	0.10	
	SAGE$^+$	10	23	9.49	7.44	8.47	0.10		10	17	0.54	5.56	3.05	0.10		10	2	6.13	6.92	6.52	0.10	

Table 8. Comparison with baselines across three datasets and three values of k. For $k = 3$ and $k = 5$, we used $\theta = 0.30$ on Marketing and Video, and $\theta = 0.20$ on House. For $k = 10$, $\theta = 0.10$ was used across all datasets. Columns show the number of pivot tables recommended (#PT), elapsed time (T, in seconds), *Insightfulness* (Ins), *Interpretability* (Int), *Utility* (Util), and *Diversity* (Div) in terms of minimum pairwise distance (m-dist) and a distance matrix visualization (heatmap). In the distance matrix, the diagonal is black, denoting 0 self distance from a pivot table to itself. Lighter colors denote less similar pairs, offering diversity. For Excel, PowerBI, and Google Sheets, k cannot be controlled. Thus, we consider the first-k items when more than k are returned. Google Sheets fails to produce more than 3 recommendations, thus, we exclude it when $k > 3$. The best values in Util and m-dist are marked as bold and the second best underlined.

For Video, SAGE outperforms all baselines across all cases in utility and two cases in diversity. SAGE and Top-k's similar performance can be attributed to the dataset attributes, such as North-America Sales and Europe-Sales, which possess similar value ranges, leading to comparable utility. For House, Top-k marginally outperforms SAGE in utility, but at the cost of very low diversity. SAGE diversity is relatively poor here, because the pruning phase retained only 11 of 81 attributes, where others used all attributes.

While Top-k achieves high utility by design, it fails to diversify. LLM performs well on Marketing but struggles on Video and House, because Marketing has many categorical values, which LLMs are adept at interpreting and utilizing for diverse recommendations. DAISY shows good diversity because it predicts most insightful tables, benefitting diversity. While PowerBI demonstrates good diversity across the board, it typically yields poor utility. Google Sheets demonstrates good diversity on Marketing and Video, however, its recommendations are limited to a small set of tables, leading to low utility. It also fails for House due to high dimensionality (81 attributes), indicating its limitation in handling complex, high-dimensional datasets. Excel consistently shows reasonable utility, but with low diversity, due to allowing significant column overlaps.

	Approach	BruteForce	No PR/PX	No PR	No PX	SAGE
	#Pivot Tables (PTs)	130	130	130	20	20
K = 2	#PT combinations	8,385	N/A	N/A	N/A	N/A
	Runtime (s)	5,817	4,971	76	492	23
	Utility (%)	100	100	97	100	97
	– Insightfulness (%)	100	100	94	100	94
	– Interpretability (%)	100	100	100	100	100
K = 5	#PT combinations	286M	N/A	N/A	N/A	N/A
	Runtime (s)	19,604	4,971	76	492	24
	Utility (%)	100	100	97	100	97
	– Insightfulness (%)	100	100	92	100	92
	– Interpretability (%)	100	100	100	100	100

PR = Pruning optimization
PX = LLM-Proxy optimization
No PR/PX = Plain greedy, no optimization
SAGE = Greedy + PR + PX

Table 9. Comparison among variants for $\theta = 0.1$ on the Marketing dataset over 5 attributes. We report the runtimes for the online phase for SAGE and other variants when optimizations are applied.

In most cases, the practical variant SAGE$^+$ shows slight reduction in the utility score. However, the runtime gain is significant compared to SAGE, e.g., from 161s to 23s for Marketing. Since SAGE$^+$ uses sampling, its scores can be either an over- or under-approximation. However, empirically we found its scores to be comparable to SAGE's scores (typically within 10%).

6.4 Effect of Optimizations

Table 9 demonstrates how the optimizations—pruning (PR) and LLM-proxy (PX)—significantly improve SAGE's runtime performance while costing minimal utility loss. For this experiment, we used a vertical slice of the Marketing dataset over 5 attributes to allow Brute Force to finish within a reasonable time. We performed an ablation study where we turned off all the optimizations (No PR/PX = plain greedy), no pruning (No PR), no LLM-proxy (No PX), and SAGE with both optimizations. Notably, we find that greedy achieves exact results in par with Brute Force, validating our choice.

BruteForce shows an infeasible runtime of over $19K$ seconds (over 5 hours) even for a moderate $k = 5$, as the number of candidate pivot table combinations was $\binom{130}{5} \approx 286$M, for only 5 data attributes. LLM-proxy causes slight decrease (3%) in utility due to less accurate proxy classifier, however, boosts performance significantly. Pruning does not hurt utility, due to its conservative filtering of unpromising candidates. Fig. 10 (left) shows runtime comparison over three datasets (limited to 5 attributes), where both optimizations offer significant performance boost across the board. Pruning is significantly beneficial for high-dimensional datasets, due to the exponential growth of the candidate space w.r.t attributes.

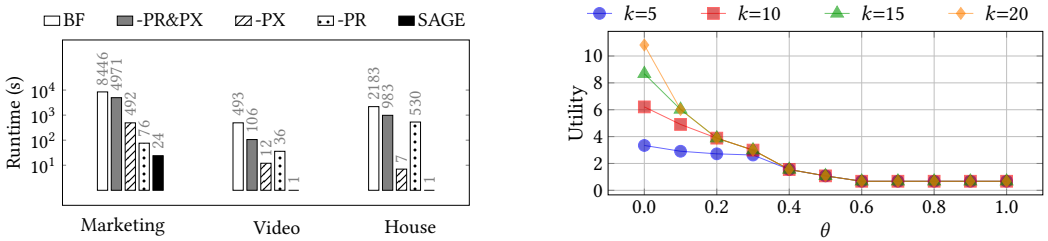


Fig. 10. (Left) Effect of optimization techniques. We used $k = 4$ and $\theta = 0.1$. (Right) Effect of θ on *Utility* on the Video dataset.

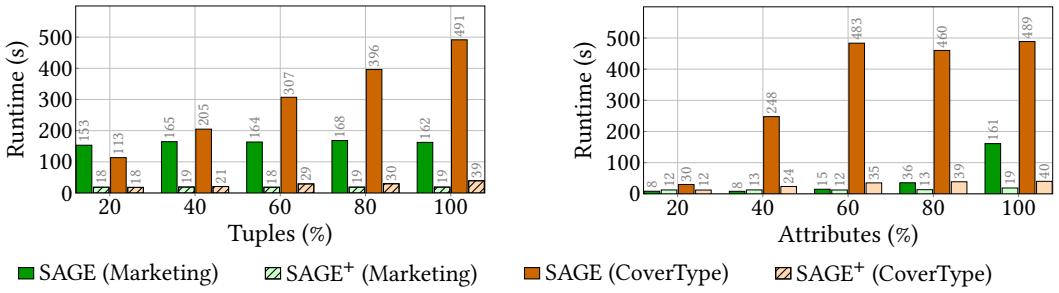


Fig. 11. SAGE and SAGE$^+$ runtime (s) w.r.t (Left) #tuples, (Right) #attributes. We used $k = 5$ and $\theta = 0.10$.

6.5 Scalability

Fig. 11 shows the runtimes of SAGE and SAGE$^+$ w.r.t #tuples (left) and #attributes (right), averaged over three executions. For both SAGE and SAGE$^+$, runtime increases linearly with data cardinality and quadratically with dimensionality, because increasing dimensions significantly expands the combinatorial search space. SAGE$^+$ shows only small fluctuations as #tuples or #attributes increases, thanks to its sampling and approximation methods. Remarkably, SAGE$^+$ processed the CoverType dataset (110 attributes, 581K+ tuples) in under 50 seconds—demonstrating practical scalability. Despite the substantial runtime reduction, SAGE$^+$ maintained comparable utility and diversity scores to SAGE (within a 10% variation).

6.6 Parameter Sensitivity

Fig. 10 (right) depicts the impact of varying the diversity threshold (θ) on the utility score over the Video dataset. We observe that as θ increases, the utility score drops for all k. This is expected because a higher θ constrains the selection of candidates more stringently, leading to fewer eligible items. To satisfy this stricter criterion, the algorithm may be compelled to select items with lower utility to maintain diversity, thus decreasing the total utility. The utility drop becomes particularly significant for high values of k, as satisfying a high diversity threshold for a larger result set is more challenging.

6.7 Case Studies

We now present findings from case studies in which we manually examined the pivot tables for qualitative insights regarding recommendations made by SAGE vs. commercial software and LLMs.

6.7.1 Diverse and meaningful aggregates. We found SAGE to consistently recommend diverse and semantically meaningful aggregates. For Video, SAGE's includes `COUNT, MAX, MIN, SUM`, and `MEAN`. In contrast, Excel, Google Sheets, and PowerBI typically involve only `SUM` or `COUNT`. While LLMs too suggest a variety of aggregates, they often involve hallucinated attributes. E.g., `LotLocation` is not an attribute of House, which LLM recommends. Despite requesting to use the specified aggregates, LLM still selects out-of-scope aggregates such as median or standard deviation. Excel once suggested `SUM(Birth_Year)`, a completely meaningless aggregate.

6.7.2 Diverse and meaningful attributes. We found SAGE to select diverse and semantically meaningful attributes for aggregation and grouping. In Video, SAGE utilizes `Publisher, Sales,` and `Year` for various aggregations, providing comprehensive coverage of the attribute space. This contrasts sharply with Excel and Google Sheets, which make repetitive `GROUP BY` choices, while LLM consistently focuses only on `Sales` for aggregation. In House, SAGE identifies attributes such as `KitchenAbvGr`, `HalfBath`, and `FullBath` for aggregations that other baselines ignore. While only SAGE and LLM appropriately use `SalesPrice` for aggregation, LLM limits itself to aggregate

only over this attribute. In contrast, SAGE diversifies selection of attributes, avoiding meaningless choices made by other tools, such as PowerBI's poor recommendation to `GROUP BY ID`.

6.7.3 Avoiding unsurprising summaries. Since SAGE can assess the degree of surprisingness in observed trends, it avoids recommending trivial queries with low utility. Unlike other tools that consistently recommend common aggregations such as `SUM(INCOME)`, regardless of context, SAGE avoids recommendations that provide obvious insights (e.g., people with low GDP spend less). For the Video dataset, existing baselines typically focused on aggregations such as `SUM(SALES)`. In contrast, SAGE included diverse types of aggregations such as `MEAN(NA_Sales)` and `MAX(JP_Sales)`. Additionally, it discovered more surprising patterns, such as `COUNT(YEAR) GROUP BY GENRE`, which were not considered by others.

6.7.4 Adaptive recommendation. We evaluated SAGE's adaptability to user needs (desiderata D4) through a case study on the Video dataset. Initially, the user indicates `Genre` as a grouping attribute they are interested in, and requests three recommendations ($k = 3$). In the first round, SAGE suggests pivot tables showing `COUNT(Publishers)` and `AVERAGE(Global Sales)` by `Genre` and `Region`. After the user indicates disinterest in `Publishers`, SAGE adapts by recommending `COUNT(Years)` and `AVERAGE(EU Sales)` by `Genre`, which the user accepts. For the third recommendation, SAGE presents `SUM` and `MEAN` of `EU Sales` and `Global Sales` by `Genre`, further diversifying the aggregates while maintaining focus on the user-specified grouping on `Genre`. Unlike existing pivot table recommenders that disallow customizability and lack mechanism for user feedback, SAGE dynamically adapts to user preferences.

7 User Study

To validate whether our utility (Section 3) and diversity (Section 4) models align with real-world user perception, we conducted a comprehensive user study with 36 participants, 53% of whom use spreadsheets daily or several times per week. The study comprised five sections, each containing 1–4 objective questions followed by an open-ended prompt for participants to explain their choices. Below, we summarize each section's setup and key findings based on the responses.

Validating insightfulness. We presented four pairs of pivot tables; within each pair, one had a high score and the other a low score for one of the *Insightful* components: (a) Attribute Significance (Section 3.1.1), (b) Informativeness (Section 3.1.2), (c) Trend (Section 3.1.3), and (d) Surprise (Section 3.1.4). For each pair, participants had to select the more insightful table or indicate a tie. The percentages of participants who chose the high-scoring pivot table (declared a tie) were 88.9% (8.3%), 91.7% (8.3%), 91.7% (8.3%), and 75.0% (16.7%), respectively. These results show that, on average, 86.8% of participants agreed with our notion of pivot-table insightfulness, demonstrating strong alignment between our model and human perception.

Validating interpretability. Setup of this section was identical to the previous one, but to validate three *Interpretability* components: (a) Density (Section 3.2.1), (b) Semantic validity (Section 3.2.2), and (c) Conciseness (Section 3.2.3). The percentages of users who identified the high-scoring pivot table as more interpretable (declared a tie) were 77.8% (8.3%), 91.7% (5.6%), and 91.7% (8.3%), respectively. These results show that, on average, 87.1% of participants agreed with our notion of pivot-table interpretability, demonstrating strong alignment between our model and human perception.

Validating utility via ablation. This section included two questions, each presenting a pair of pivot tables for participants to choose the one that is overall more useful—i.e., shows strong insights while being easy to interpret—or declare a tie. We found that 86.1% (8.3%) of participants preferred

(tied with) the balanced summary over a highly interpretable but low-insight summary, and 69.4% (13.9%) preferred (tied with) the balanced summary over a highly insightful but less interpretable one. These results establish that both components of our utility model are essential.

Validating Diversity. We asked participants to compare two sets of pivot tables, one more diverse than the other. A striking 91.7% preferred the diverse set, confirming alignment between our model and user perception.

Contrasting with other baselines. We asked two questions to compare recommendations by SAGE with four other baselines: Microsoft Excel, Google Sheets, an LLM-based method, and Top-K, with names of the tools anonymized to avoid any bias. Among the participants, 52.8% preferred SAGE over the other tools and 13.9% noted that all recommendations are equally good. This result confirms that SAGE consistently provides more user-aligned recommendations compared to existing baselines.

The free-text justifications helped us understand participants' reasoning and we found several interesting responses. For instance, one wrote, "Tool A (SAGE) is [...] the best option, because each table provides useful insights. Tool B has the odd sum of birth year attribute, while Tool C essentially has the same table twice [...]." We provide additional details of the user study in our full technical report [18].

8 Related Work

Below, we discuss three major areas that share the general problem of data summarization and understanding. Ours is a particular instance of this problem, which is recommending interesting aggregates or views to help users understand data.

OLAP cube exploration. Prior work in OLAP cube exploration [55, 63–66] aim to identify "surprising" data regions to help users uncover previously unseen and interesting patterns. They define surprise based on user familiarity, and interestingness based on purely statistical methods. However, they lack awareness of the underlying data *semantics*, often resulting in recommendations that are practically uninteresting and hard to interpret. Moreover, they typically suggest a single item or a top-k list, without considering *bundle* suggestions that offer a *set* of diversified items.

Insight generation and visualization. A number of prior work define informative summaries as tables that exhibit significant statistical disparities among groups [14, 24, 28, 36, 70, 76]. Some measure the significance of insights using null-hypothesis testing [14, 28] while others exploit LLMs [76]. While insight generation and visualization recommendation [14, 76, 77, 81] are related to our work, most prior works in this space do not incorporate the three key dimensions together: insightfulness, interpretability, and diversity. Notably, recommending a k-budgeted *set* of summaries is a significantly harder problem than simple top-k recommendation.

View or exploration-step recommendation. View recommendation systems [75, 77, 83] explore a related direction. Voyager [77] ranks visualizations based on perceptual effectiveness to improve interpretability whereas ViewSeeker [83] compares multiple visualizations using various similarity measures. However, these systems are not semantics-aware and do not consider diversity. Smart Drill-Down [40] enables users to discover and summarize interesting groups of tuples described by rules. However, their unit is a rule, while ours is a pivot table. Auto-Suggest [79] and DAISY [78] leverage large-scale user logs, SQL queries, or crowdsourcing to recommend pivot tables based on historical usage patterns. However, they cannot ensure pivot-table informativeness, as they do not validate the content of the generated pivot tables.

In summary, prior work fall short in one of the following aspects. *First*, they do not consider the semantic validity and interpretability aspects of the summary structure. E.g., (a) the aggregates `SUM(Birth_Year)` and `AVG(Zip_Code)` are semantically invalid, and (b) an attribute with values {`Val_1, Val_2`} is not interpretable whereas {`Large, Small`} is. *Second*, they lack *semantic awareness* to *validate* an apparent data insight. For instance, they typically determine the degree of insightfulness purely based on statistical properties, such as whether aggregated values between two groups significantly deviate from each other, without factoring in the degree of expectedness of that deviation based on common knowledge (e.g., a significant deviation in average height between toddlers and adults is expected). In contrast, SAGE consults with an LLM—which can mimic a human domain expert who can provide the likelihood of observing a statistically interesting phenomenon—and effectively prunes statistically insightful, but practically mundane insights. *Third*, beyond simple top-k, they do not focus on the *bundle recommendation problem*—which involves suggesting a *set* of k results—with *diversity* requirements, which is our focus. *Fourth*, while a few works consider diversity in summarization [32, 74, 75, 81], they ignore the semantic aspect of diversity. DAISY [78] models diversity during query collection, not recommendation, while QAGView [75] diversifies at a tuple-level granularity. In contrast, we use a semantics-aware model for the distance metric—using query and contend embedding—to ensure *semantic* diversity across the recommended pivot tables.

Diversification algorithms. Query result diversification [72] methods define diversity across three aspects: (1) content, (2) novelty, and (3) coverage [30, 31]. Max-Sum diversification [15, 30, 58] maximizes the linear combination of diversity and utility scores, while Max-Min diversification maximizes the minimum diversity between selected items. However, existing diversification methods target tuples or documents, not entire tables. Thus, they do not trivially extend to table recommendations, as in our case.

9 Conclusions and Future Work

We presented SAGE to recommend diverse set of pivot tables while balancing insightfulness and interpretability. We introduced a utility model for a single pivot table, a diversity metric for pivot-table sets, and a simple greedy algorithm built on two optimization techniques for efficient recommendation. We empirically showed that SAGE outperforms baselines in diversity while maintaining high utility. Our case studies illustrated SAGE's ability to avoid generic patterns and provide data-semantics-aware suggestions. To the best of our knowledge, this is the first work to combine diversity and data-semantics for data summarization.

SAGE currently does not incorporate contextual information, such as the user's workflow (e.g., their end goals) or broader ecosystem (e.g., tools in their software stack). Integrating such context could significantly enhance the quality of recommendations. Another promising direction is to support alternative forms of data summaries—such as textual descriptions that highlight key trends—in addition to structured formats like pivot tables. Finally, another important direction of future work is optimizing the performance of the offline phase, such as using an adaptive approach to dynamically adapt when the spreadsheet distribution changes.

Acknowledgments. We thank Aditya Bhaskara and Zafeiria Moumoulidou for insightful discussions on the theoretical complexity of our algorithms; Jeff M. Phillips, El Kindi Rezig, and the UtahDB students for their insightful feedback throughout this work; and Aritra Mazumder and Shiyi He for assistance with the design of our user study. We also thank the students of the Advanced Database Management course at the University of Utah (Fall 2025) for their participation in the user study. This work was supported in part by the NSF under the Grant CNS-2346555, a Stena Center for Financial Technology Seed Grant, and a One Utah Data Science Hub Seed Grant.

References

[1] 2023. Microsoft Community Forum: Issues with null values in pivot tables. https://answers.microsoft.com/en-us/msoffice/forum/all/how-to-get-rid-of-blank-appearing-in-pivot-table/449f785a-f993-4c40-922b-1b52f02571ce. Accessed: 2025-06-02.

[2] Raghavendra Addanki, Andrew McGregor, Alexandra Meliou, and Zafeiria Moumoulidou. 2022. Improved Approximation and Scalability for Fair Max-Min Diversification. In *25th International Conference on Database Theory, ICDT 2022, March 29 to April 1, 2022, Edinburgh, UK (Virtual Conference) (LIPIcs, Vol. 220)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 7:1–7:21.

[3] Sameet Agarwal, Rakesh Agrawal, Prasad Deshpande, Ashish Gupta, Jeffrey F. Naughton, Raghu Ramakrishnan, and Sunita Sarawagi. 1996. On the Computation of Multidimensional Aggregates. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3-6, 1996, Mumbai (Bombay), India*. Morgan Kaufmann, 506–521.

[4] Gauss Algorithmic. 2025. T5-LM-Large-text2sql-spider Model Card. https://huggingface.co/gaussalgo/T5-LM-Large-text2sql-spider. Accessed 16 July 2025.

[5] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[6] Sahil Bajaj. 2025. Marketing Campaigns Data Set. https://www.kaggle.com/datasets/sahilnbajaj/marketing-campaigns-data-set. Accessed 16 July 2025.

[7] Pierre Barrouillet, Sophie Bernardin, Sophie Portrat, Evie Vergauwe, and Valérie Camos. 2007. Time and cognitive load in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 33, 3 (2007), 570.

[8] Leilani Battle and Jeffrey Heer. 2019. Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau. *Comput. Graph. Forum* 38, 3 (2019), 145–159.

[9] P. J. Bickel, E. A. Hammel, and J. W. O'Connell. 1975. Sex Bias in Graduate Admissions: Data from Berkeley. *Science* 187, 4175 (1975), 398–404.

[10] Jock A. Blackard. 1998. Covertype — Forest Cover Types Dataset. https://archive.ics.uci.edu/ml/datasets/Covertype. Accessed: 2025-10-07.

[11] Colin R. Blyth. 1972. On Simpson's Paradox and the Sure-Thing Principle. *J. Amer. Statist. Assoc.* 67, 338 (1972), 364–366.

[12] Allan Borodin, Hyun Chul Lee, and Yuli Ye. 2012. Max-Sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*. ACM, 155–166.

[13] Matteo Brucato, Juan Felipe Beltran, Azza Abouzied, and Alexandra Meliou. 2016. Scalable Package Queries in Relational Database Systems. *Proc. VLDB Endow.* 9, 7 (2016), 576–587.

[14] Yu-Rong Cao, Xiao-Han Li, Jia-Yu Pan, and Wen-Chieh Lin. 2022. VisGuide: User-Oriented Recommendations for Data Event Extraction. In *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022*. ACM, 412:1–412:13.

[15] Jaime G. Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*. ACM, 335–336.

[16] Minmin Chen, Yuyan Wang, Can Xu, Ya Le, Mohit Sharma, Lee Richardson, Su-Lin Wu, and Ed H. Chi. 2021. Values of User Exploration in Recommender Systems. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*. ACM, 85–95.

[17] Whanhee Cho and Anna Fariha. 2024. UTOPIA: Automatic Pivot Table Assistant. *Proc. VLDB Endow.* 17, 12 (2024), 4305–4308.

[18] Whanhee Cho and Anna Fariha. 2025. Data-Semantics-Aware Recommendation of Diverse Pivot Tables. arXiv:2507.06171 [cs.DB]

[19] Whanhee Cho and Anna Fariha. 2025. SAGE: Data-Semantics-Aware Recommendation of Diverse Pivot Tables. https://github.com/whnhch/sage.

[20] Deepti Chopra. 2022. The Importance of Excel in Business. https://www.adaface.com/blog/importance-of-excel-in-business. Accessed: 2025-11-27.

[21] Graham Cormode and S. Muthukrishnan. 2005. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms* 55, 1 (2005), 58–75.

[22] Microsoft Corporation. 2025. Microsoft Power BI. https://www.microsoft.com/en-us/power-platform/products/power-bi. Accessed: 2025-02-16.

[23] Anna DeCastellarnau. 2018. A classification of response scale characteristics that affect data quality: a literature review. *Quality & Quantity* 52, 4 (2018), 1523–1559.

[24] Çagatay Demiralp, Peter J. Haas, Srinivasan Parthasarathy, and Tejaswini Pedapati. 2017. Foresight: Recommending Visual Insights. *Proc. VLDB Endow.* 10, 12 (2017), 1937–1940.

[25] Dazhen Deng, Aoyu Wu, Huamin Qu, and Yingcai Wu. 2023. DashBot: Insight-Driven Dashboard Generation Based on Deep Reinforcement Learning. *IEEE Trans. Vis. Comput. Graph.* 29, 1 (2023), 690–700.

[26] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 143–177.

[27] Diginomica. 2025. *Shadow IT Never Dies: Why Spreadsheets Are Still Running Your Business.* Accessed: 2025-01-21.

[28] Rui Ding, Shi Han, Yong Xu, Haidong Zhang, and Dongmei Zhang. 2019. QuickInsights: Quick and Automatic Discovery of Insights from Multi-Dimensional Data. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*. ACM, 317–332.

[29] Marina Drosou, H. V. Jagadish, Evaggelia Pitoura, and Julia Stoyanovich. 2017. Diversity in Big Data: A Review. *Big Data* 5, 2 (2017), 73–84.

[30] Marina Drosou and Evaggelia Pitoura. 2010. Search result diversification. *SIGMOD Rec.* 39, 1 (2010), 41–47.

[31] Marina Drosou and Evaggelia Pitoura. 2012. DisC diversity: result diversification based on dissimilarity and coverage. *Proc. VLDB Endow.* 6, 1 (2012), 13–24.

[32] Ori Bar El, Tova Milo, and Amit Somech. 2020. Automatically Generating Data Exploration Sessions Using Deep Reinforcement Learning. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*. ACM, 1527–1537.

[33] Anna Fariha, Ashish Tiwari, Arjun Radhakrishna, Sumit Gulwani, and Alexandra Meliou. 2021. Conformance Constraint Discovery: Measuring Trust in Data-Driven Systems. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. ACM, 499–512.

[34] Kareem El Gebaly, Parag Agrawal, Lukasz Golab, Flip Korn, and Divesh Srivastava. 2014. Interpretable and Informative Explanations of Outcomes. *PVLDB* 8, 1 (2014), 61–72.

[35] Google. 2025. Google Sheets - Online Spreadsheet Editor. https://sheets.google.com. Accessed: 2025-01-21.

[36] Camille Harris, Ryan A. Rossi, Sana Malik, Jane Hoffswell, Fan Du, Tak Yeon Lee, Eunyee Koh, and Handong Zhao. 2023. SpotLight: Visual Insight Recommendation. In *Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*. ACM, 19–23.

[37] Wassily Hoeffding. 1994. *Probability Inequalities for sums of Bounded Random Variables*. Springer New York, New York, NY, 409–426.

[38] Apple Inc. 2025. Numbers - Apple (IN). https://www.apple.com/in/numbers. Accessed: 2025-01-21.

[39] Mahmood Jasim, Christopher Collins, Ali Sarvghad, and Narges Mahyar. 2022. Supporting Serendipitous Discovery and Balanced Analysis of Online Product Reviews with Interaction-Driven Metrics and Bias-Mitigating Suggestions. In *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022*. ACM, 9:1–9:24.

[40] Manas Joglekar, Hector Garcia-Molina, and Aditya G. Parameswaran. 2019. Interactive Data Exploration with Smart Drill-Down. *IEEE Trans. Knowl. Data Eng.* 31, 1 (2019), 46–60.

[41] William B Johnson, Joram Lindenstrauss, et al. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics* 26, 189-206 (1984), 1.

[42] Changhee Joo, Xiaojun Lin, Jiho Ryu, and Ness B Shroff. 2015. Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels. *IEEE/ACM Transactions on Networking* 24, 3 (2015), 1476–1488.

[43] Ralph Kimball. 2013. *The data warehouse toolkit : the definitive guide to dimensional modeling* (third edition. ed.). John Wiley & Sons, Incorporated.

[44] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, Ricardo Baeza-Yates, and Hongyuan Zha. 2017. Exploring Query Auto-Completion and Click Logs for Contextual-Aware Web Search and Query Suggestion. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 539–548.

[45] Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table Pre-training via Learning a Neural SQL Executor. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

[46] Meta Llama. 2025. Llama-3.1-8B-Instruct Model Card. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct. Accessed 16 July 2025.

[47] Meta. 2025. Llama3-Intruct. https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct. Accessed: 2025-02-25.

[48] Microsoft. 2025. Microsoft Excel - Spreadsheet Software. https://www.microsoft.com/en-us/microsoft-365/excel. Accessed: 2025-01-21.

[49] Microsoft. 2025. Tapex-large Model Card. https://huggingface.co/microsoft/tapex-large. Accessed 16 July 2025.

[50] George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2 (1956), 81.

[51] Zafeiria Moumoulidou, Andrew McGregor, and Alexandra Meliou. 2021. Diverse Data Selection under Fairness Constraints. In *24th International Conference on Database Theory, ICDT 2021, March 23-26, 2021, Nicosia, Cyprus (LIPIcs, Vol. 186)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 13:1–13:25.

[52] Sk Md Abu Nayeem and Madhumangal Pal. 2007. Genetic algorithmic approach to find the maximum weight independent set of a graph. *Journal of Applied Mathematics and Computing* 25 (2007), 217–229.

[53] Godwin Odu. 2019. Weighting methods for multi-criteria decision making technique. *Journal of Applied Sciences and Environmental Management* 23 (09 2019), 1449.

[54] OpenAI. 2025. ChatGPT. https://chat.openai.com. Accessed: 2025-02-14.

[55] Carlos Ordonez and Zhibo Chen. 2009. Exploration and visualization of OLAP cubes with statistical tests. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration, Paris, France, June 28, 2009*. ACM, 46–55.

[56] Animesh Parikshya. 2025. House Sale Dataset. https://www.kaggle.com/datasets/animeshparikshya/house-sale-data-81-column. Accessed 16 July 2025.

[57] Zening Qu and Jessica Hullman. 2018. Keeping Multiple Views Consistent: Constraints, Validations, and Exceptions in Visualization Authoring. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (2018), 468–477.

[58] Filip Radlinski and Susan T. Dumais. 2006. Improving personalized web search using result diversification. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*. ACM, 691–692.

[59] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[60] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 3980–3990.

[61] Shuichi Sakai, Mitsunori Togasaki, and Koichi Yamazaki. 2003. A note on greedy algorithms for the maximum weighted independent set problem. *Discrete applied mathematics* 126, 2-3 (2003), 313–322.

[62] Salesforce. 2024. Tableau. https://www.tableau.com. Accessed: 2025-10-14.

[63] Sunita Sarawagi. 1999. Explaining Differences in Multidimensional Aggregates. In *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*. Morgan Kaufmann, 42–53.

[64] Sunita Sarawagi. 2000. User-Adaptive Exploration of Multidimensional Data. In *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*. Morgan Kaufmann, 307–316.

[65] Sunita Sarawagi. 2001. User-cognizant multidimensional analysis. *VLDB J.* 10, 2-3 (2001), 224–239.

[66] Sunita Sarawagi, Rakesh Agrawal, and Nimrod Megiddo. 1998. Discovery-Driven Exploration of OLAP Data Cubes. In *Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology, Valencia, Spain, March 23-27, 1998, Proceedings (Lecture Notes in Computer Science, Vol. 1377)*. Springer, 168–182.

[67] Gregory Smith. 2025. Video Game Sales Dataset. https://www.kaggle.com/datasets/gregorut/videogamesales. Accessed 16 July 2025.

[68] Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, Haoze Li, Haoxuan Lan, Jiaming Tian, Jing Yuan, Junbo Zhao, Junlin Zhou, Kaizhe Shou, Liangyu Zha, Lin Long, Liyao Li, Pengzuo Wu, Qi Zhang, Qingyi Huang, Saisai Yang, Tao Zhang, Wentao Ye, Wufang Zhu, Xiaomeng Hu, Xijun Gu, Xinjie Sun, Xiang Li, Yuhang Yang, and Zhiqing Xiao. 2024. TableGPT2: A Large Multimodal Model with Tabular Data Integration. arXiv:2411.02059 [cs.LG]

[69] Aaquib Syed, Phillip Huang Guo, and Vijaykaarti Sundarapandiyan. 2023. Prune and Tune: Improving Efficient Pruning Techniques for Massive Language Models. In *The First Tiny Papers Track at ICLR 2023, Tiny Papers @ ICLR 2023, Kigali, Rwanda, May 5, 2023*. OpenReview.net.

[70] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya G. Parameswaran, and Neoklis Polyzotis. 2015. SEEDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics. *Proc. VLDB Endow.* 8, 13 (2015), 2182–2193.

[71] Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. Efficient Large Language Models: A Survey. *Transactions on Machine Learning Research* (2024). Survey Certification.

[72] Yue Wang, Alexandra Meliou, and Gerome Miklau. 2018. RC-Index: Diversifying Answers to Range Queries. *Proc. VLDB Endow.* 11, 7 (2018), 773–786.

[73] Zhewei Wei and Ke Yi. 2011. Beyond simple aggregates: indexing for summary queries. In *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece.*

PODS, 117–128.
[74] Yuhao Wen, Xiaodan Zhu, Sudeepa Roy, and Jun Yang. 2018. Interactive Summarization and Exploration of Top Aggregate Query Answers. *Proc. VLDB Endow.* 11, 13 (2018), 2196–2208.
[75] Yuhao Wen, Xiaodan Zhu, Sudeepa Roy, and Jun Yang. 2018. QAGView: Interactively Summarizing High-Valued Aggregate Query Answers. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018.* ACM, 1709–1712.
[76] Luoxuan Weng, Xingbo Wang, Junyu Lu, Yingchaojie Feng, Yihan Liu, Haozhe Feng, Danqing Huang, and Wei Chen. 2025. InsightLens: Augmenting LLM-Powered Data Analysis With Interactive Insight Management and Navigation. *IEEE Trans. Vis. Comput. Graph.* 31, 6 (2025), 3719–3732.
[77] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock D. Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Trans. Vis. Comput. Graph.* 22, 1 (2016), 649–658.
[78] Junjie Xing, Xinyu Wang, and H. V. Jagadish. 2024. Data-Driven Insight Synthesis for Multi-Dimensional Data. *Proc. VLDB Endow.* 17, 5 (2024), 1007–1019.
[79] Cong Yan and Yeye He. 2020. Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020.* ACM, 1539–1554.
[80] Cong Yan, Yin Lin, and Yeye He. 2023. Predicate Pushdown for Data Science Pipelines. *Proc. ACM Manag. Data* 1, 2 (2023), 136:1–136:28.
[81] Brit Youngmann, Sihem Amer-Yahia, and Aurélien Personnaz. 2022. Guided Exploration of Data Summaries. *Proc. VLDB Endow.* 15, 9 (2022), 1798–1807.
[82] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018.* Association for Computational Linguistics, 3911–3921.
[83] Xiaozhong Zhang, Xiaoyu Ge, Panos K. Chrysanthis, and Mohamed A. Sharaf. 2021. ViewSeeker: An Interactive View Recommendation Framework. *Big Data Res.* 25 (2021), 100238.
[84] Ye Zhang, Stephen Roller, and Byron C. Wallace. 2016. MGNC-CNN: A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016.* The Association for Computational Linguistics, 1522–1527.

Received July 2025; revised October 2025; accepted November 2025