Scalable Goal-Oriented Source Selection

Ambarish Singh
Purdue University
West Lafayette, IN, USA
sing1104@purdue.edu

Romila Pradhan
Purdue University
West Lafayette, IN, USA
rpradhan@purdue.edu

ABSTRACT

Given a query table, existing data lake discovery techniques seek to identify tables related to the query table through schema similarity and joinability. The discovered tables, however, are not always useful for downstream analyses; while some improve task generalization, some have the potential of introducing harmful noise or distribution shifts. We formalize *source selection* as the problem of finding a subset of the discovered tables (or, *sources*) that maximize downstream task utility. We propose Splice, a system based on the idea of gene splicing to efficiently solve the source selection problem by identifying sources that provide additional signals that a single source consolidated over all sources misses. On classification tasks over two real-world benchmarks derived from SOTA table union search tools, Splice shows $3-5\%$ improvement in F1 scores compared to baselines. Beyond data discovery benchmarks, compared to the case when all discovered tables are considered, Splice improves task utility on two real-world classification tasks by $2-5\%$ with up to 95% reduction in model evaluation costs.

VLDB Workshop Reference Format:
Ambarish Singh and Romila Pradhan. Scalable Goal-Oriented Source Selection. VLDB 2026 Workshop: 15th International Workshop on Quality in Databases (QDB'26).

1 INTRODUCTION

Data lakes have emerged as a powerful storage paradigm to store heterogeneous data extracted from several data sources without enforcing a unified schema. Given a target schema to train a machine learning model, preparing the training dataset from a data lake primarily consists of two distinct steps: *discovering* which data lake tables are related to the target schema, and *selecting* which of the discovered tables are useful to the downstream task utility. Existing tools such as Santos[10], Josie[21], and Aurum[4] have made the discovery step tractable at the scale of data lakes through techniques such as measuring schema overlap and computing the joinability of tables. Despite these efforts, a key gap remains: a table that is joinable with the target table is not necessarily *useful* for model training. In other words, a joinable table is not guaranteed to improve the downstream task utility. While data discovery systems often accurately identify tables that are suitable for training the model, they lack the knowledge of whether the label distribution, feature coverage, or domain shift of an identified table will help or harm the model.

Proceedings of the VLDB Endowment. ISSN 2150-8097.

Example 1.1. Consider the Santos Large benchmark [10] consisting of 11,090 tables. Given a query table with 9 attributes, table union search system Santos [10] returns the top 20 joinable tables: all monthly spending records from the UK Intellectual Property Office sharing a related schema. Using the query table as the test set and the remaining 19 tables as candidate data sources, we consider the binary classification task of predicting whether a transaction is high-spend (target variable `Amount` exceeding the median amount in the query table) or low spend given its `ExpenseType` and `ExpenseArea`. Subsets of these tables exhibit varying utilities: a model trained on all sources has an F1 score of 0.715 while one trained only on August 2010 data has zero F1 score. The latter, while schema compatible, has solely low-spend transactions; hence, a classifier trained on it wrongly predicts every transaction of query month as low-spend. In contrast, a carefully selected subset of 7 tables has F1 score= 0.745, exceeding the bound of 0.726 obtained when the query table is used for both training and testing.

While data integration solutions usually consolidate the discovered sources into a single dataset for downstream tasks, note that considering all the sources is not the most optimal choice. More data is often expected to result in a more generalizable model but is vulnerable to differences in data distributions of the query data and the source data which primarily arise from three kinds of failures: (a) *label distribution mismatch*, when the outcome rate of query data differs from that of the sources (e.g., 36% fewer positive outcomes in August 2010 data compared to query data); (b) *feature coverage gap*, when sources from different organizational units or time periods adopt categories not present in query data, thus causing vocabulary mismatch in feature space encoding; (c) *source cancellations*, when the effect of a source harmful to the task dominates model behavior and nullifies the effect of useful sources (e.g., data from June 2010 achieves an F1 score of 0.736, but combining 19 sources degrades it to 0.715 possibly caused by harmful sources such as August 2010).

We leverage the aforementioned insights to propose Splice, an algorithm inspired from gene-based splicing [1] to efficiently select the subset of data sources best-suited for a downstream machine learning task. Figure 1 shows an overview of Splice. Starting from an initial *active set* of data sources drawn from the pool of discovered sources, Splice iteratively refines the selection by swapping out *low-value* sources (that do not improve the task utility) and replacing them with *high-value* sources (that improve the task utility) from the inactive set comprising the remaining discovered data sources. The algorithm converges when the active set no longer changes, and is chosen as the final selected subset. Determining whether a selection of sources is of high value or low value entails training a model with the selected sources and computing its utility — a computationally expensive step. Instead of model training, we efficiently estimate the improvement in task utility by using a meta

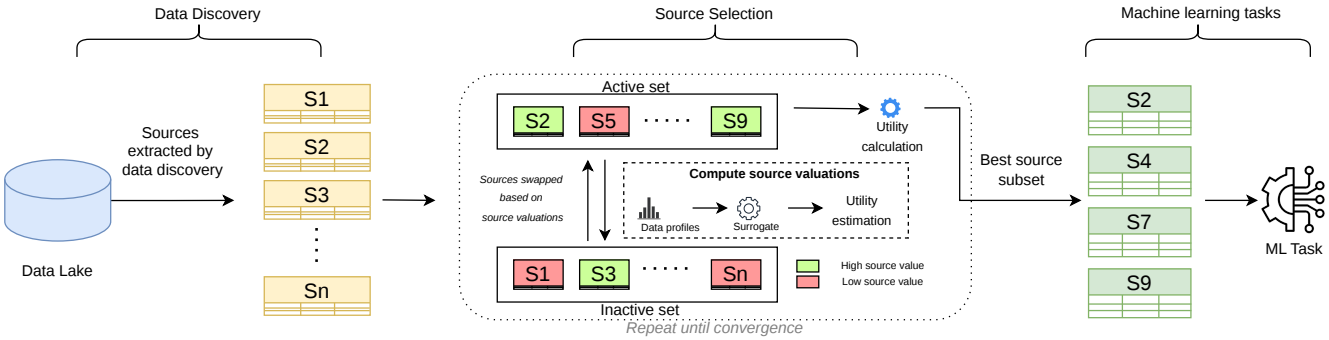


Figure 1: An overview of SPLICE: a gene-splicing-based source selection mechanism that actively adds and removes sources from an active set to output the subset of sources having the best downstream performance.

learning approach [20] over the computed *data profiles* [5] of the resulting dataset. Details on the approach can be found in Section 3. Our main contributions can be summarized as follows:

- We formalize the source selection problem as determining the subset of data sources that results in the best utility over a downstream machine learning task. (Section 2).
- We propose SPLICE, an efficient algorithm based on gene splicing that finds subsets with high utility without exhaustive enumeration. We further introduce SPLICE-DP, a surrogate variant that uses data profiles to estimate marginal utility, substantially reducing the number of model trainings required. (Section 3).
- We empirically demonstrate that SPLICE and SPLICE-DP consistently outperform several baselines on two SANTOS benchmarks and two real-world ML tasks, with SPLICE-DP returning the best subsets in interactive response times. (Section 4).

2 PRELIMINARIES

We consider a labeled target dataset $\mathcal{T} \in \text{Dom}(\mathbf{X}) \times \text{Dom}(Y)$ where $\mathbf{X}$ denotes a set of features and $\text{Dom}(Y)$ denotes the set of labels.

Data sources. Let $\mathcal{S} = \{S_1, \ldots, S_n\}$ denote a set of candidate data sources returned by a data discovery tool. Therefore, each source $S_i = (\mathbf{X}_i, \mathbf{Y}_i)$ has a sufficient schema overlap with $\mathcal{T}$. Given a subset of sources $\mathbf{S} \subseteq \mathcal{S}$, let $\mathbf{D}_S$ denote the dataset formed by combining data points in each source $S \in \mathbf{S}$.

Machine learning task. We consider the problem of binary classification ($\text{Dom}(Y) = \{0, 1\}$) with the objective of training a classifier $\mathcal{M} : \text{Dom}(\mathbf{X}) \rightarrow \hat{Y}$ on training dataset $\mathbf{D}$ such that each data point $\mathbf{x} \in \mathbf{X}$ has an associated predicted label $\hat{y} = \mathcal{M}(\mathbf{x}) \in \{0, 1\}$.

Profit score. We define the *profit* score of a subset $\mathbf{S}$ over model $\mathcal{M}$, denoted by $\mathcal{P}_{\mathcal{T}}(\mathbf{S}) \in \mathbb{R}$, as a real value that quantifies the quality of analysis performed by the model when trained on subset $\mathbf{S}$ and evaluated on target dataset $\mathcal{T}$. We map profit score to diverse performance metrics such as model accuracy, mean squared error, fairness, F1 score etc. Profit score can also be modeled as a combination of performance metrics. For example, model accuracy and true positive rate can be combined to model the profit score $\mathcal{P}_{\mathcal{T}}(\mathbf{S}) = \text{Accuracy}(\mathcal{M}(\mathbf{S})) + \lambda \cdot \text{TruePositiveRate}(\mathcal{M}(\mathbf{S}))$ where $\lambda \geq 0$ controls the weight of model fairness relative to accuracy.

Given a target dataset, a pool of data sources and a profit score, we define the source selection problem as follows:

PROBLEM 1 (SOURCE SELECTION PROBLEM). *Given a set of candidate data sources $\mathcal{S} = \{S_1, ..., S_n\}$, target dataset $\mathcal{T}$ and a profit score $\mathcal{P}$, determine the subset $\mathbf{S}^* \subseteq \mathcal{S}$ that maximizes $\mathcal{P}_{\mathcal{T}}(\mathbf{S})$:*

$$\mathbf{S}^* = \arg\max_{\mathbf{S} \subseteq \mathcal{S}} \quad \mathcal{P}_{\mathcal{T}}(\mathbf{S}) \tag{1}$$

Equation (1) is NP-hard as the search space has $2^{|\mathcal{S}|}$ candidate subsets; practical solutions must, therefore, find high-quality subsets without exhaustively enumerating all possible subsets of sources.

3 SPLICE: GENE-SPLICING-BASED SOURCE SELECTION

We describe our approach for selecting the best subset of data sources that has a high profit score. We first describe the core of our approach that uses the concept of gene splicing [1]. We then expedite our solution by using meta learning [20] to estimate profit scores without model training. Our solution, SPLICE, draws on the idea of gene splicing which is a concept originally used for the synthesis of alternate proteins and has been hugely successful in domains such as healthcare, medicine, and agriculture [12, 14, 15]. Adapting it to our problem setting, SPLICE starts with an active and an inactive subset of sources and simultaneously swaps out low-value sources for high-value ones. Rather than building a subset by greedily adding one source at a time or through random perturbations, SPLICE allows discarding locally optimal subsets.

SPLICE starts with an active set of sources $A \subseteq \mathcal{S}$ seeded with the top-i sources ranked by individual profit (where $i = \{1, \ldots, S_{max}\}$ and S_{max} is the maximum subset size). Given the number of sources that can be swapped in each iteration (denoted by k), the splicing step scores each source $s \in A$ for removal from the active set and each source $u \in \mathcal{S} \setminus A$ for addition to the active set using their *marginal* profit contributions:

$$\Delta^-(s) = \mathcal{P}(A) - \mathcal{P}(A \setminus \{s\}) \tag{2}$$

$$\Delta^+(u) = \mathcal{P}(A \cup \{u\}) - \mathcal{P}(A) \tag{3}$$

If $\Delta^- < 0$, then removing the source from the active set is beneficial; a higher magnitude of Δ^- indicates a better candidate for removal. In contrast, $\Delta^+ > 0$ indicates that adding the source to the active set is beneficial with higher Δ^+ being better. The k sources with the lowest Δ^- are removed from the active set and the k sources with the highest Δ^+ are added to the active set. The resulting active set is

Algorithm 1: SPLICE for Source Selection

Input: Set of sources $\mathcal{S}$, max. iterations S_{max}, max. swaps k_{max}
Output: Optimal subset $\mathbb{S}^*$, maximum profit P^*

$\mathbb{S}^* \leftarrow \emptyset, P^* \leftarrow -\infty$;
for $i \leftarrow 1$ **to** S_{max} **do**
  $A_{init} \leftarrow$ Top-i sources in $\mathcal{S}$ ranked by `get_profit`$(\{s\})$;
  $A_{res}, P_{res} \leftarrow$ `FixedSupport`$(A_{init}, \mathcal{S})$;
  if $P_{res} > P^*$ **then** $P^* \leftarrow P_{res}, \mathbb{S}^* \leftarrow A_{res}$;
`FixedSupport`$(A, \mathcal{S})$;
  $A_{curr} \leftarrow A, A_{prev} \leftarrow \emptyset$;
  while $A_{curr} \neq A_{prev}$ **do**
    $A_{prev} \leftarrow A_{curr}, k \leftarrow \min(k_{max}, |A_{prev}|)$;
    A_{curr}, profit $\leftarrow$ `Splicing`$(A_{prev}, \mathcal{S}, k)$;
  return A_{curr}, profit;
`Splicing`$(A, \mathcal{S}, k_{max})$;
  $P_{max} \leftarrow$ `get_profit`$(A), A_{best} \leftarrow A, \mathcal{I} \leftarrow \mathcal{S} \setminus A$;
  for $k \leftarrow 1$ **to** k_{max} **do**
    for $s \in A$ **do** $\Delta^-(s) \leftarrow -$`EstGain`$(A, A \setminus \{s\})$;
    $L^- \leftarrow$ `SortAsc` $(A,$ key $= \Delta^-)$, $A_{tmp} \leftarrow A \setminus L^-[1:k]$;
    for $u \in \mathcal{I}$ **do** $\Delta^+(u) \leftarrow$ `EstGain`$(A_{tmp}, A_{tmp} \cup \{u\})$;
    $L^+ \leftarrow$ `SortDsc` $(\mathcal{I},$ key $= \Delta^+)$, $A_{new} \leftarrow A_{tmp} \cup L^+[1:k]$;
    $P_{curr} \leftarrow$ `get_profit`(A_{new});
    if $P_{curr} > P_{max}$ **then** $P_{max} \leftarrow P_{curr}, A_{best} \leftarrow A_{new}$;
  return A_{best}, P_{max};

evaluated and accepted if it improves the current best profit score. This process is performed for all swap sizes $k = 1 \ldots k_{max}$ and the process is repeated until convergence. The entire procedure is then run for initialization sizes $i = 1, \ldots, s_{max}$, seeding the active set with the top-i individually ranked sources at each run. Initializing with different swapping sizes covers both sparse and dense subsets and avoids reliance on a single starting point. Finally, the subset with the best overall profit score is returned as output. The pseudocode for SPLICE can be seen in Algorithm 1.

Note that evaluating a subset of sources using $\mathcal{P}(A)$ requires training and testing an ML model, which is practically infeasible with a large number of sources. We expedite this computation by estimating the marginal profit gains without full model evaluation.

Gain estimation using data profiles We rely upon the concept of *meta learning* using data profiles [6] to develop the SPLICE-DP variant that leverages characteristics of data points in a subset of sources to accurately estimate its marginal profit gain. SPLICE-DP trains a lightweight linear surrogate regression model on the difference between the data profiles of a candidate subset and the current active set to predict the marginal profit gain of the former. The model is trained offline on a small set of subset evaluations and is used online to estimate marginal profit gains of candidate source subsets. This adaptation improves model evaluations by $5-15$x while preserving profit improvement.

4 EXPERIMENTAL EVALUATION

We evaluate the effectiveness and efficiency of SPLICE through experiments on two real-world datasets and two data discovery benchmarks comparing it with competing methods for the task of selecting the best subset of sources for downstream analyses.

4.1 Experimental Setup

Datasets. We evaluated our approach on the following datasets and benchmarks: ACSIncome [3] dataset contains demographic and employment information of 1.6M individuals and is used to predict whether one earns more than 50k annually. The sensitive attribute is sex and true positive rate (TPR) parity is enforced through $\lambda = 50$ in the profit score. We partition the dataset by US states and select a subset of 15 states of varying sizes as sources in our experiments with the state of Puerto Rico serving as the test source. Amazon [9] dataset contains 270k product reviews for 27 product categories. We use 24 categories (spanning digital goods, physical products, and consumables) as data sources and 3 categories to create the target. Given a review, the task predicts its sentiment (rating ≥ 4: positive; else, negative) and reports the F1 score. Data discovery benchmarks [10] returned by SANTOS. The YDNPA benchmark covers 20 monthly files from Yorkshire Dales National Park Authority whereas the IPO benchmark covers 20 monthly files from the UK Intellectual Property Office. The query table in each group is designated as the target and the remaining 19 form the pool of data sources. The task predicts whether a transaction exceeds the target table's median spend using `Expense Type` and `Expense Area` and reports the binary F1 score.

Competing Methods. We consider the following methods for creating the best subset of sources to train a model:
SINGLESOURCE: Each data source is evaluated independently and the source that results in the highest profit is selected.
ALLSOURCES: All data sources are combined to form a single dataset.
RANDOM samples random subset of sources; averaged over 10 runs.
GREEDY: Iteratively adds the source with the highest profit score; stops when no further improvement found.
GRASP: Greedy randomized adaptive search procedure [18] that alternates between randomized greedy construction and local search.
SPLICE and SPLICE-DP: Our algorithms proposed in Section 3.

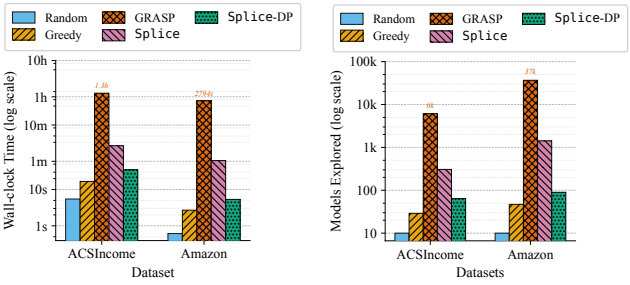
4.2 Performance of SPLICE on ML tasks

We report profit scores of the methods on two ML tasks in Table 1 and observe that SPLICE has an overall superior performance. On ACSIncome, both SPLICE and GRASP exhibit similar performance; however, as seen in Figure 2, GRASP requires over 6k model evaluations (Figure 2b) and more than an hour of wall-clock time (Figure 2a) compared to 307 model evaluations and less than three minutes required by SPLICE. On Amazon, SPLICE has a slightly better profit score than GRASP at an order of magnitude lower evaluation cost. We also observe that combining all sources has significant performance gap compared to SPLICE, reflecting the effect of harmful sources (as in Example 1.1). Moreover, our gain estimation method SPLICE-DP exhibits performance comparable to SPLICE albeit at a $5-15$x reduction in model evaluations, staging it as a practical choice in the face of expensive model evaluation.

Figure 3 shows anytime performance on the two tasks i.e., the best profit found as a function of models explored. We observe a stark contrast between the methods. On Amazon, SPLICE-DP reaches a high F1 score (=0.755) within the first 50 evaluations

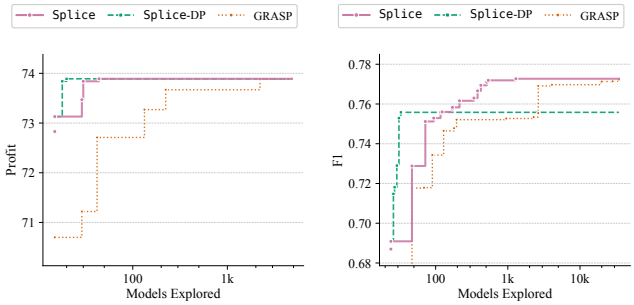
Task	Dataset	SingleSource	AllSources	Random	Greedy	Grasp	Splice	Splice-DP
Accuracy+Fairness	ACSIncome	73.135	72.289	71.623	73.877	**73.892**	**73.892**	**73.892**
Sentiment analysis	Amazon	0.6908	0.7274	0.6946	0.7528	0.7723	**0.7726**	0.7558

Table 1: Results by task and dataset. Best results are shown in bold and second-best results are underlined.



(a) Time taken to select sources. **(b) Number of models explored.**

Figure 2: Efficiency of different methods.



(a) ACSIncome. **(b) Amazon.**

Figure 3: Effect of number of models explored on profit.

and plateaus, trading a small amount of final quality for dramatically faster convergence. Splice starts lower, overtakes Splice-DP around 200 evaluations, and continues improving to its final value of 0.773 by 500 evaluations. In contrast, Grasp makes slow, irregular stepwise progress through tens of thousands of evaluations before matching Splice. A similar behavior is observed even in the case of ACSIncome. Together, these curves illustrate the core efficiency argument: Splice-DP is the right choice when evaluation budgets are tight, Splice is preferable when the budget allows a few hundred evaluations and maximum quality is needed, and Grasp should be avoided when evaluation cost is non-negotiable.

4.3 Integration of Splice with data discovery

To demonstrate Splice's compatibility with data discovery tools, we report the performance when different source selection methods are applied on tables discovered by Santos: + Single selects the single best among the discovered sources, + All selects all discovered sources, and + Splice selects sources using our approach.

Dataset	Santos + Single	Santos + All	Santos + Splice
YDNPA (Spending)	0.737	0.722	**0.768**
IPO (Payments)	0.736	0.715	**0.745**

Table 2: F1 scores on tables discovered by Santos.

We observe that selecting all sources discovered by Santos rather than selecting the single best source hurts the F1 scores. Furthermore, Splice selects 6 of 19 sources for YDNPA and 7 of 19 for IPO, and exceeds the oracle self-training upper bound obtained by training and testing on the target table itself. This behavior is not a contradiction: the source months selected by Splice have spending patterns that closely mirror the target month, providing labeled examples augmenting the target table. Our most striking observation was the IPO worst-source result (F1 = 0.000 for August 2010 table). Even though this source is joinable per Santos' criteria (same schema, organization, and reporting format), a classifier trained on it predicts every target transaction as low-spend. Careful selection of sources is one of the early steps in the data pipeline with the potential to catch such faulty behaviors.

5 RELATED WORK

The present work is related to research in data discovery and data valuation. While we studied these areas extensively, our approach of valuation-based source selection for optimized ML tasks is novel.

Data discovery using table union search. There is a rich body of literature on table union search that aims to discover joinable or semantically similar tables in data lakes [2, 13, 17, 19]; the more recent works include Josie [21], Santos [10], and Aurum [4]. Splice treats these tools as black-box discovery oracles, takes the tables discovered by these tools as input sources and determines the sources best suited for a downstream task.

Data valuation and selection. Data valuation has recently emerged as a powerful tool to explain the output of black-box ML models where techniques such as Data Shapley [7] and influence functions [11] quantify the value of each data point toward model performance. Splice operates at the granularity of sources (a natural unit in data lake management since provenance, licensing, and access control are primarily tracked per table instead of data points [8, 16]) and evaluates sources for task utility, thus avoiding the problem of scaling that comes with a large number of data points.

6 CONCLUSION AND FUTURE WORK

We argue that data source *selection* is a necessary post-discovery step that current data lake systems lack. While data discovery tools return schema-compatible tables, they usually do not determine which tables are useful for the downstream task. We present Splice, a source selection algorithm based on the idea of swapping harmful sources for useful ones, and expedite the process using meta learning. We show that this distinction matters: on the IPO benchmark, while one of the discovered sources achieves zero F1 score, a careful selection of 7 sources exceeds the oracle self-training bound. Splice (and its variant, Splice-DP) identifies high-quality sources efficiently with 26x (56x) lower evaluation cost. Future work includes further reducing the cost of model evaluation through gradient-based gain estimation techniques, pruning the set of discovered sources early on, and addressing multiple target selections where a set of sources must serve several downstream tasks simultaneously.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) under Grant #2237149.

REFERENCES

[1] Susan M Berget, Claire Moore, and Phillip A Sharp. 1977. Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences* 74, 8 (1977), 3171–3175.

[2] Michael J Cafarella, Alon Halevy, and Nodira Khoussainova. 2009. Data integration for the relational web. *Proceedings of the VLDB Endowment* 2, 1 (2009), 1090–1101.

[3] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems* 34 (2021), 6478–6490.

[4] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1001–1012.

[5] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and robust automated machine learning. *Advances in neural information processing systems* 28 (2015).

[6] Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. 2014. Using meta-learning to initialize bayesian optimization of hyperparameters.. In *MetaSel@ ECAI*. 3–10.

[7] Amirata Ghorbani and James Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. PMLR, 2242–2251.

[8] Alon Halevy, Flip Korn, Natalya F Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. 2016. Goods: Organizing google's datasets. In *Proceedings of the 2016 International Conference on Management of Data*. 795–806.

[9] Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2026. Bridging Language and Items for Retrieval and Recommendation: Benchmarking LLMs as Semantic Encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (Eds.). Association for Computational Linguistics, San Diego, California, United States, 3251–3265. https://doi.org/10.18653/v1/2026.acl-long.147

[10] Aamod Khatiwada, Grace Fan, Roee Shraga, Zixuan Chen, Wolfgang Gatterbauer, Renée J Miller, and Mirek Riedewald. 2023. Santos: Relationship-based semantic table union search. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–25.

[11] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. 1885–1894.

[12] Stanley Chun-Wei Lee and Omar Abdel-Wahab. 2016. Therapeutic targeting of splicing in cancer. *Nature medicine* 22, 9 (2016), 976–986.

[13] Oliver Lehmberg and Christian Bizer. 2017. Stitching web tables for improving matching quality. *Proceedings of the VLDB Endowment* 10, 11 (2017), 1502–1513.

[14] Jun Li, Xiangbing Meng, Yuan Zong, Kunling Chen, Huawei Zhang, Jinxing Liu, Jiayang Li, and Caixia Gao. 2016. Gene replacements and insertions in rice by intron targeting using CRISPR–Cas9. *Nature plants* 2, 10 (2016), 1–6.

[15] Xiang Lin, Haizhu Chen, Ying-Qian Lu, Shunyan Hong, Xinde Hu, Yanxia Gao, Lu-Lu Lai, Jin-Jing Li, Zishuai Wang, Wenqin Ying, et al. 2020. Base editing-mediated splicing correction therapy for spinal muscular atrophy. *Cell research* 30, 6 (2020), 548–550.

[16] Fatemeh Nargesian, Erkang Zhu, Renée J Miller, Ken Q Pu, and Patricia C Arocena. 2019. Data lake management: challenges and opportunities. *Proceedings of the VLDB Endowment* 12, 12 (2019), 1986–1989.

[17] Fatemeh Nargesian, Erkang Zhu, Ken Q Pu, and Renée J Miller. 2018. Table union search on open data. *Proceedings of the VLDB Endowment* 11, 7 (2018), 813–825.

[18] Celso C Ribeiro, Pierre Hansen, S Binato, WJ Hery, DM Loewenstern, and MGC Resende. 2002. A GRASP for job shop scheduling. *Essays and surveys in metaheuristics* (2002), 59–79.

[19] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Y Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. 2012. Finding related tables.. In *SIGMOD Conference*, Vol. 10. 2213836–2213962.

[20] Joaquin Vanschoren. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548* (2018).

[21] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J Miller. 2019. Josie: Overlap set similarity search for finding joinable tables in data lakes. In *Proceedings of the 2019 International Conference on Management of Data*. 847–864.