Fast Natural Language Based Data Exploration with Samples

Shubham Agarwal, Gromit Yeuk-Yin Chan, Shaddy Garg, Tong Yu, Subrata Mitra*
Adobe Research

ABSTRACT

The ability to extract insights from large amounts of data in a timely manner is a crucial problem. Exploratory Data Analysis (EDA) is commonly used by analysts to uncover insights using a sequence of SQL commands and associated visualizations. However, in many cases, this process is carried out by non-programmers who must work within tight time constraints, such as in a marketing campaign where a marketer must quickly analyze large amounts of data to reach a target revenue. This paper presents ApproxEDA –a system that combines a natural-language processing (NLP) interface for insight discovery with an underlying sample-based EDA engine. The NLP interface can convert high-level questions into contextual SQL queries of the dataset, while the backend EDA engine significantly speeds up insight discovery by selecting the most optimum sample from among many pre-created samples using various sampling strategies. We demonstrate that ApproxEDA addresses two key aspects: converting high-level NLP inputs to contextual SQL and intelligently selecting samples using a reinforcement-learning agent. This protects users from diverging from their original intent of analysis, which can occur due to approximation errors in results and visualizations, while still providing optimal latency reduction through the use of samples.

ACM Reference Format:
Shubham Agarwal, Gromit Yeuk-Yin Chan, Shaddy Garg, Tong Yu, Subrata Mitra. 2023. Fast Natural Language Based Data Exploration with Samples. In *Proceedings of ACM Conference (SIGMOD'23).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Business-critical decisons are often made based on insights derived from a massive amount of data in a time-critical manner. Let us consider the life of a digital marketer's day on a *prime-day* with an expected target revenue that the designed advertisement campaign must meet. When terabytes of live customer purchasing data flows in, the marketers have to adjust their campaigns quickly based on the insights derived from this data to meet the revenue target. These insights are usually a sequence of approximate queries that return attributes from samples step by step so that users could reason from the data quickly. However, there are two main challenges. First, although queries from samples could be used for quick insight-generations, the results could deviate from the actual due to approximation errors and this can get more pronouced and ultimately lead to *intent-divergence* [4] in a sequential exploration such as exploratory data analysis (EDA), where next query is chosen based on the previous queries and their corresponding outcomes. Second, marketers are often not proficient with common interfaces for processing big-data such as SQL and python during time-critival scenarios. There may not be enough time to seek or communicate the need to SQL experts or analysts. Thus, more no-code interfaces are needed to aid such scenarios.

To address the challenge of diverging insights from multiple approximated queries, we could use different sampling strategies (combination of sampling algorithms and sampling rates) to preserve different levels of statistical aspects of the data and provide different speedup in query-latency. Our recent work [4] addresses the problem of how to select the correct sample for each context in the sequential EDA process using a reinforcement-learning (RL) based approach. On the other hand, to reduce the programming barrier of EDA, prior work has explored the use of natural language processing (NLP) to generate SQL queries [10]. Given large scale human labelled text-to-sql datasets [11], we can fine tune pre trained language models like T5 [9] and BERT [2] to obtain a query interface suitable for novice and domain experts. Together, it creates an opportunity to design an interactive interface for EDA that utilize fast queries and NLP. In this demo paper, we will present an interacive exploratory data analysis system, ApproxEDA, that integrates NLP-based easy to use interface (with recommendations) with an intelligent contextual sample selection mechanism for the SQL-queries using our recently proposed RL-based technique [4].

ApproxEDA bootstraps the insight generation process through an interactive NLP-based interface that can accept high-level questions about the data and produces SQL query recommendations. User can directly run these SQL queries or perform further edits to these queries. Since ApproxEDA uses different sampling strategy to speedup these queries, it also indicates which sampling strategy was chosen for each query and how confident it is about the results using confidence intervals and color coding. When the system can detect the *intent*, i.e. the kind of insight the user is looking for based on historical analysis patterns from expert SQL users, it can start progressively recommending SQL queries that will help the user to go into to one of those detected flow of analysis methodology. ApproxEDA's underlying RL-based sample selection agent takes the sequence of previously run SQL queries and corresponding results/visulizations to understand context of the generated SQL query and runs it using the optimal sampling strategy to optimize for speedup while protecting against intent-divergence. When the user edits the orginally recommended queries, the choice of sampling strategy and confidence on the outcome can change based on the choice of the sampling strategy by the agent. User has the option to override the selected sample strategy or can can choose to run the query on full data accurately. Note, while intent-based recommendation can be switched off in ApproxEDA, it helps in real usecases as expert users often perform data exploration with some latent intent [4].

A video demonstration is attached as a supplementary material.

*Corresponding Author

SIGMOD'23, June 2023, Seattle, WA, USA
none. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

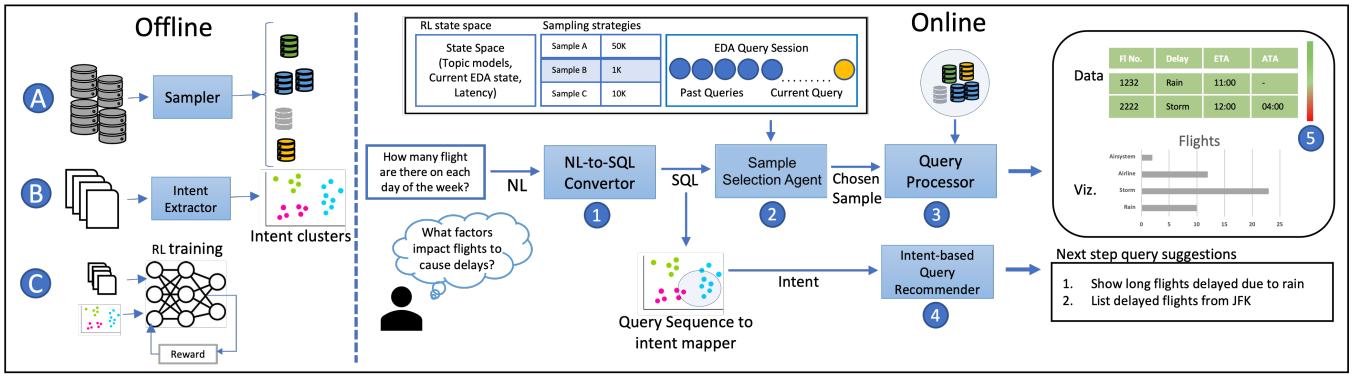

Figure 1: Architecture overview of ApproxEDA. Left part shows the offline components including (A) Offlie sample creation (B) Intent discovery from historical EDA sessions (C) Reinforcement Learning (RL) based training of contexual sample selection agent. Right side shows runtime (online) components: (1) Natural Language query to SQL converter (2) Trained sample-selection agnet for chosing optimal sampling strategy given the context in analysis sequence (3) Query processor that runs the SQL on the recommended sample to generates results/ visualization for query and also compute confidence on the answer (4) Recommendation generator for next set of queries based on intent (5) Shows visualization of confidence score for on the UI using background color.

2 SYSTEM OVERVIEW

2.1 ApproxEDA at a high-level

ApproxEDA is a tool that enables interactive, efficient, and guided exploratory data analysis. We describe the system architecture of ApproxEDA as shown in Figure 1 highlighting its offline and online (i.e. runtime) components.

2.2 ApproxEDA Offline

ApproxEDA is required to perform three major tasks offline as shown on the left side in Figure 1.

(A) **Sampler:** It creates k downsampled datasets from the original dataset using various sampling strategies and associated parameter combinations as listed in Table 1. Each downsampled dataset preserves different statistical aspects of the data and the effective sampling rate dictates the size of these samples giving a trade-off between loss of inforation due to sampling and the associated speedup each sample can provide because of its reduced size.

(B) **Intent Extractor:** ApproxEDA uses unsupervised learning, specifically bi-term model (BTM) modeling on historical analysis sequences to create a model of intent for data exploration based on the sequence of queries and visualization used [4]. Each intent represents a particular latent style of data exploration in search of a specific type of insight. These intents are used at runtime by ApproxEDA for both intelligent sample selection and query recommendations.

(C) **Reinforcement-Learning:** ApproxEDA trains an contextual and intent-aware sample selection agent using reinforcement-learning (RL). This agent learns to select the optimal sample from k precreated downsamples. In RL-training we model the selection of down-sampling strategy as the action-space and optimize the divergence of sequential interactions made by the user, taking into account the interactivity of the system and the presence of approximations in an intent-aware manner. Our proposed RL-based training uses novel reward functions and RL learning stop criteria to achieve this goal much better than any prior-work in [4]. This approach allows us to ensure that the user's analysis flow is as close as possible to an ideal flow if they were using the full dataset with accurate visualizations.

2.3 ApproxEDA Runtime

ApproxEDA's runtime or online components are shown on the right side in Figure 1 as the following major components.

(1) **Question to SQL:** ApproxEDA uses a NLP-based interactive system in the frontend for non-SQL-experts to quickly extract insights from the data using natural langauge queries. It also provides query recommendations based on past EDA sessions and the current session's intent. The system consists of three main components: (1) a text-to-SQL view to generate SQL recommendations from user text input; (2) a SQL panel showing the executed SQL with the approximation errors from the sampling process, and; (3) a visualization view to display the data from the query results.

To understand text input and translate them into SQLs query data from the tables, we use a text-to-SQL model that is trained by finetuning T5 model [9] with Spider dataset [11]. The model takes input in a format (<*question*> | <*table*> | <*schema*>) to generate SQLs that both capture the user requirements and column names from the table. The queries could act as inputs to our recommendation engine in the next section.

(2) **Query Recommendations:** Systems like [6], [5] and [7] provide the users with personalized query recommendations for analysis based on past user sessions. We use an approach based on *REACT* proposed in [7] for recommending the next set of queries to the user. It is chosen for its lightweight design and ability to incorporate the implicit intent identified by the RL-agent into the query recommendation process. The proposed system ApproxEDA takes in a repository of previously performed EDA sessions performed by human data analysts and employs Topic Modeling techniques to extract a set of implicit intents from these sequences. It then uses the current user session and the historical sessions, both modeled as trees, to identify the top-k similar subtrees. This is accomplished by using a similarity score that is a combination of tree edit distance and an intent similarity component. The edit script is built by applying edit operations, such as delete/add a node or an edge, and

alter the label of a node or an edge, on the nodes and edges. The edit distance is computed by summing the cost of these operations required to transform one tree into another, where add/delete operations have a unit cost, and the cost of an alter operation reflects the similarity between data displays and analysis actions, respectively. Additionally, it also calculates a component based on the cosine similarity of the intent vectors between the current user session and the previous sessions. The set of similar trees identified using these techniques is then used to determine the set of next actions by considering the nodes (in the retrieved set of similar subtrees) that correspond to the current user node and taking the edges (actions) outgoing from those nodes.

(3) **Query Processor and Confidence estimation:** ApproxEDA also indicates its confidence about the result or visualzation created by running a query on the sample selected. The confidence is calculated based on the estimation of the variance for the result using a sample. If the variance is large, ApproxEDA predicts that it is less certain about the displayed result and indicates that with a red-ish highlighting color in its UI as an information to the user. If the variance is low, ApproxEDA indicates it is more confident with a green-ish highlight. For `GROUP BY` queries, this variance is computed per group and average is taken. Deatils of variance computation for different aggregates and sampling strategies using closed-form expressions can be found in [8].

(4) **Sample Selector:** At runtime the contextual sample selection is done by the trained RL-agent that takes into account the following at runtime for each step in the sequential analysis: (1) the knowledge about the latent intents extracted from historical data, (2) the ongoing EDA exploration sequence including the query used so far and the corresponding display-outputs (graphs and dataframes), (3) the next query the user is intending to run, and (4) the set of available samples created with different sampling strategies along with the size of each. The agent, which is parameterized by a deep neural network, is trained offline to choose the optimal sampling strategy as the best *action* for different context of the analyses and intent. The best action corresponding to each step, i.e. for each query in the EDA session attempts to minimize the divergence of intents due to approximation error caused by different samples, while optimizing the latency reduction of queries.

2.4 Deployment Aspects

Real expert users often follow standard exploration strategies and look for similar types of insights even on new datasets as observed by multiple prior works [1, 7]. This means that the overall characteristics of intents of the analysts do not change significantly. In a production scenario, samples are periodically regenerated offline as new batches of data are ingested. If the statistical characteristics change significantly, then ApproxEDA is retrained with new samples in the background. On Flights data with 5.2 M rows and 29 samples created using different strategies, ApproxEDA took approx. 15 hrs to be fully trained from scratch on a single machine with 32 core Intel(R)Xeon(R) CPU E5-2686 with 4 Tesla V100-SXM2 GPU(s). The model-size is 363 KB.

Sample Name	Short Name	Parameters
Uniform	Uni@τ	τ= [0.01, 0.05, 0.1]
Systematic	Sys@k	k= [100, 20, 10]
Proportional stratified	Strat@τ	τ= [0.01, 0.05, 0.1]
At most K stratified	Strat-K@K	K= [2k, 10k, 20k]
Cluster	Clus@k	k= 10, τ= [0.01, 0.05, 0.1]
MaxMin Diversity	MaxMin@k	k= 0.1*$\|T\|$
MaxSum Diversity	MaxSum@k	k= 0.1*$\|T\|$

Table 1: Sampling strategies and corresponding parameters that together creates our action space. We use the *Short Name* to refer these. $|T|$ denotes the size of the original data.

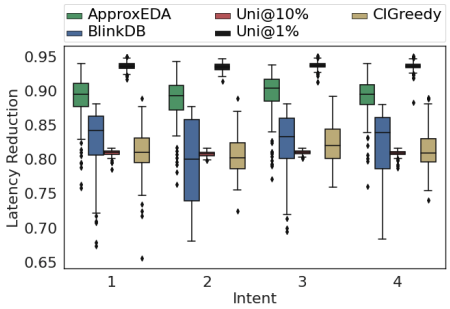

Figure 2: The fraction of time saved by queries to run on the sampled dataset when using ApproxEDA compared with different baseline approches for latency reduction. Higher is better.

2.5 ApproxEDA Evaluation

In the absence of public datasets that combines both exploratory insight generation query sequences and the corresponding results on a given data that is also publicly available, we use an EDA simulator from prior work [1], called ATENA. It uses high-level statistical characteristics (e.g. entropy, skew, # unique categories etc.) of the data along with its schema structure (e.g. numerical, categorical, primary-key column information) and it's evaluation shows [1] that it can generate realistic analysis sequence patterns similar human experts.

We consider sampling strategies listed in Table 1 with different associated parameters that control the sizes of the subsamples and consequently the amount of statistical information. In total we use 29 actions based on 6 sampling strategies and corresponding parameter combinations as presented in Table 1. Details of these well known sampling strategies and associated paraemeters can be found in [3].

In Figure 2 we compare the query latency improvement by ApproxEDA compared to different baselines that can be used for latency reduction to make natural language based interaction faster. As can be observed ApproxEDA can provide impressive speedup and only textttUni@1% (1% uniform sample) across the board can give lower latency but can introduce significant approximation error resulting in intent-divergence. Detailed evaluation can be found in full AAAI 2023 paper [3, 4].

3 DEMONSTRATION

This demonstration illustrates how ApproxEDA can be utilized by individuals with limited SQL experience to analyze a large dataset. Figure 3 shows different components in ApproxEDA's UI.

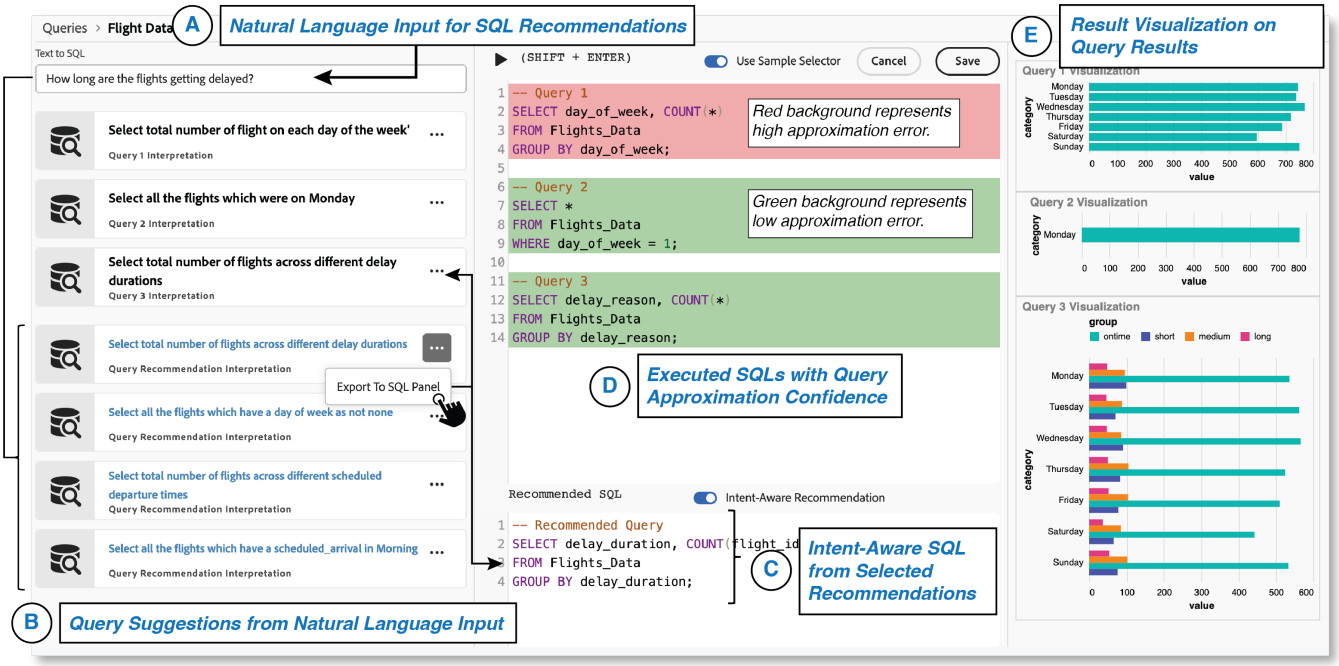

Figure 3: User Interface of ApproxEDA. (A) Text input from users to ask questions to the database. (B) Recommended queries based on user input (C) User-selected queries in SQL format. (D) Executed queries with approximation error. (E) Visualization generated from query results.

(A) is the natural language interface for converting natural language queries into SQL. Panel that provides NL query recommendation based on the current session and intent is shown in (B). These queries can be selected to view the corresponding SQL in (C) and finally executed where ApproxEDA's intelligent sample selection strategy decides which samples to be used to generate the results and visualization in (E). Once queries are executed it gets listed in panel (D) and ApproxEDA also calculates its confidence on the answer and highlights the background of the query in (D) to indicate how accurate the results might be. A green-ish background for the query indicates higher confidence. In the demonstration we plan to show the following workflow:

(a) The user queries the flight dataset using natural language queries that are recommended or by initiating a new query.

(b) ApproxEDA takes the input and converts it to the corresponding SQL query, which can be edited.

(c) ApproxEDA selects a sample using a trained RL model and runs the query to generate results. As the exploration continues, and the user runs a few more queries, ApproxEDA begins to more carefully select samples based on its understanding of the user's implicit intent behind the queries.

(d) The user can also see the approximation error made by using a sample using the color coding of the queries.

(e) The interface provides with a list of suggested next queries based on the current session queries and the implicit intent captured by the agent.

In summary, ApproxEDA with its NLP based interaction mode and inttilgent sample selection technique can provide an interactive, progressive, and guided workflow for discovering insights from large datasets for users who may not have SQL expertise or working with a new, unfamiliar dataset. A video of the demonstration is provided as supplementary material.

4 CONCLUSION

In this paper we introduce ApproxEDA and demonstrate how it can enable fast and natural langugage based sequential big data exploration and insight generation guided by associated query recommendation system. ApproxEDA achieves low latency that is suitable for natural langugae based data exploration using sampling. It uses reinforcement learning based technique for intelligent sample selecion based on the context of the sequential exploration, its sample selection strategy protects users from doing misleading exploration due to sampling error and guides the users by providing recommendations by understanding the latent intent of their analysis flow. Systems like ApproxEDA can democratize big-data exploration to non-SQL users and help in providing time-critical insights over massive amounts of data.

REFERENCES

[1] Ori Bar El, Tova Milo, and Amit Somech. 2020. Automatically generating data exploration sessions using deep reinforcement learning. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1527–1537.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Shaddy Garg, Subrata Mitra, Tong Yu, Yash Gadhia, and Arjun Kashettiwar. 2022. Reinforced Approximate Exploratory Data Analysis. *arXiv preprint arXiv:2212.06225* (2022).

[4] Shaddy Garg, Subrata Mitra, Tong Yu, Yash Gadhia, and Arjun Kashettiwar. 2023. Reinforced Approximate Exploratory Data Analysis. In *AAAI*.

[5] Eugenie Y Lai, Mostafa Milani, Omar AlOmeir, and Rachel Pottinger. [n.d.]. Sequence-Aware Query Recommendation Using Deep Learning. ([n. d.]).

[6] Eugenie Y Lai, Zainab Zolaktaf, Mostafa Milani, Omar AlOmeir, Jianhao Cao, and Rachel Pottinger. 2023. Workload-Aware Query Recommendation Using Deep

Learning. (2023).
[7] Tova Milo and Amit Somech. 2018. Next-step suggestions for modern interactive data analysis platforms. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 576–585.
[8] Barzan Mozafari and Ning Niu. 2015. A Handbook for Building an Approximate Query Engine. *IEEE Data Eng. Bull.* 38, 3 (2015), 3–29.
[9] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 140 (2020), 1–67.
[10] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. *arXiv preprint arXiv:2109.05093* (2021).
[11] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887* (2018).