Data-To-Question Generation Using Deep Learning

1st Nicole Rachel Koshy
Applied Data Science
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
nickoshy@iu.edu

2nd Anshuman Dixit
Applied Data Science
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
anshumandixit1996@gmail.com

3rd Siddhi Shrikant Jadhav
Applied Data Science
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
ssjadhav@iu.edu

4th Arun V Penmatsa
Applied Data Science
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
rpenmats@iu.edu

5th Sagar V Samanthapudi
Applied Data Science
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
svsamant@iu.edu

6th Mothi Gowtham Ashok Kumar
Applied Data Science
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
moashok@iu.edu

7th Sydney Oghenetega Anuyah
Applied Data Science
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
sanuyah@iu.edu

8th Gourav Vemula
Applied Data Science
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
gvemula@iu.edu

9th Patricia Snell Herzog
Philanthropic Studies
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
psherzog@iupui.edu

10th Davide Bolchini
Human-Centered Computing
Indiana Univeristy Purdue University Indianapolis
Indianapolis, United States
dbolchin@iupui.edu

***Abstract*— Many publicly available datasets exist that can provide factual answers to a wide range of questions that benefit the public. Indeed, datasets created by governmental and non-governmental organizations often have a mandate to share data with the public. However, these datasets are often underutilized by knowledge workers due to the cumbersome amount of expertise and embedded implicit information needed for everyday users to access, analyze, and utilize their information. To seek solutions to this problem, this paper discusses the design of an automated process for generating questions that provide insight into a dataset. Given a relational dataset, our prototype system architecture follows a five-step process from data extraction, cleaning, pre-processing, entity recognition using deep learning, and questions formulation. Through examples of our results, we show that the questions generated by our approach are similar and, in some cases, more accurate than the ones generated by an AI engine like ChatGPT, whose question outputs while more fluent, are often not true to the facts represented in the original data. We discuss key limitations of our approach and the work to be done to bring to life a fully generalized pipeline that can take any data set and automatically provide the user with factual questions that the data can answer.**

Keywords*— *question generation, data analytics, semantic typing, meta categories, Sherlock, Spacy, semantic distance calculation, ChatGPT, LDA, NLP, deep learning, knowledge extraction, topic modeling

I. Introduction

Since the advent of Natural Language Processing (NLP), accurate semantic typing of data attributes has been a crucial part of automating deep learning approaches in the exploration and identification of important characteristics of datasets. While subject matter experts can delve into most domain specific datasets with minimal assistance, big data and the ease of connecting disparate datasets into meaningful supersets has compounded the challenges we face with extracting meaningful information from such data in a timely manner.

The paradigm of generating questions to aid in the understanding of scientific datasets is a natural language problem that requires the use of a plethora of deep learning models for semantic typing, topic modeling and semantic distance calculation to determine the most relevant questions that can aid in the exploration of a dataset. Designing templates that can tackle a variety of data requires combining variables with differing semantic attributes within a dataset, along with their associated operators, before they can be converted into meaningful questions. This conversion process can follow a few methods. One method is to generate questions using only column names. Another combines column names and associated values to generate questions whose answers come directly from the data.

In this context, this project seeks answers to the question: How do we generate the maximum number of insights from a given dataset without spending an inordinate amount of time understanding and exploring the data? Can we develop pipelines that can generate preformulated queries whose answers can serve as the starting point in the exploration of a new dataset? Our work addresses the broader issues involved in this paradigm and uses datasets within the Indiana Datahub to build a program that can take a scientific dataset with its data dictionary and description, to generate questions that aid in understanding the data.

To this end, we previously worked on a question generation program that could automatically generate meaningful questions from official statistic datasets with a geo-spatial emphasis by using a semantic parser [1] to identify meta-categories, a Latent Dirichlet Allocation (LDA) model

This project was supported by funding from the National Science Foundation under Grant IIS #1909845. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSF.

to identify domain specific keywords and a randomized query generator to generate natural language factual questions.

In that model, we were limited in the scope of the data we could analyze by the semantic parser we had created. To address this problem, in the current iteration of our work, we took this a step further by combining our approach with Sherlock, a deep-learning neural network model for semantic detection that has been trained on 686, 765 data attributes to identify and assign one of 78 semantic meta-categories for a given data column [2].

The core of our Question Generation from Datasets (QGD) pipeline is split into 5 phases that handle data extraction and cleaning, entity recognition, semantic categorization, similarity index calculation for correlating identified meta-categories with domain specific keywords and a question generation module that formulates natural language questions by semantically transforming a question string generated by combining related columns in the dataset with associated operators and dataset values. We have used the Indiana Hub [3] data repository as the data source for our project. Figure 1 provides an outline of the workflow of our pipeline.

One limitation of our program is Sherlock's ability to correctly detect semantic types. Being a pre-trained model that cannot take inputs, our ability to generate questions is dependent on being able to combine correlated column data that together would create a logical question whose answer can be found within the dataset.

Currently, we use a combination of pre-defined hardcoded templates that perform this operation by preferentially combining columns with location, categorical, datetime data with numeric data, so that the questions generated center around the non-numeric data. As an added exercise, we have also compared the output of our program against ChatGPT when it was provided with the same datasets to determine the differences in our approaches.

We briefly describe the functioning of our modules before discussing a comparative analysis of our output versus ChatGPT [4]. Our work can be found at https://github.com/NicoleK286/Automated-Question-Generation.

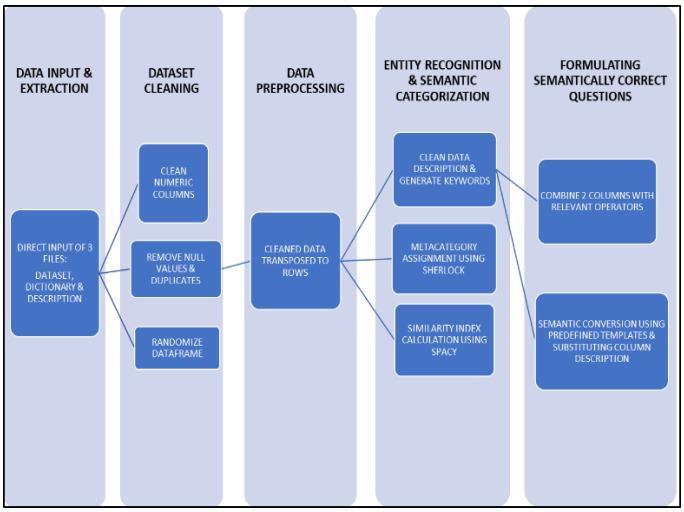

Fig 1: Flowchart of the functioning of the QGD pipeline

II. Methodology

A. Data Input

Our pipeline was designed using publicly available statistical datasets from Indiana Hub as use cases. For a given dataset, we input 3 files: the raw data, a data dictionary and a dataset description file. The preferred file format for raw data and the dictionary is *.csv and, *.txt for description files.

B. Dataset Cleaning and Pre-Processing

The raw data file is subjected to a series of steps for cleaning. Since most statistical datasets would require cleaning of mainly numeric data values that contain suppressed or other non-numeric data, we focused our module on this premise.

One of the challenges we initially faced was correct datatype detection since any column with non-numeric data is treated as an object, in python. To circumvent this, we devised a technique to correctly classify numeric data by basing detection on the number of rows with numeric data in a column. If this count exceeds the alphanumeric count, the column is then reclassified as numeric. We then split the dataset into numeric and non-numeric columns and remove rows with non-numeric data from the relevant columns. A left join is performed to concatenate the amended dataset together before subjecting the entire data to null value and duplicates removal. In edge cases, where a numeric column has excessive suppressed or non-numeric data, the dataset will not be cleaned.

For input into Sherlock, the cleaned data is then stripped of column names and all rows in a column are transposed to a horizontal list of values. This transposed data is stored in a separate file, which is now ready for meta category identification.

C. Cleaning the Data Description File

The data description file is cleaned to replace all separators with spaces and then subjected to stopwords removal to eliminate common words that would hinder semantic distance calculation. Stopwords removal is performed using the gensim module.

D. Semantic Categorization: Keyword Generation

The data description file is used to generate keywords that are compared with column names to calculate semantic distance indices which are later used in question prioritization. Topic modeling is performed on the cleaned wordlist obtained from the previous step using LDA [5] and gensim [6] to generate keywords, that are stored in a list for later use. Before distance calculation, all "_" separators present in column names are removed so that individual words in each field can be compared with the keywords.

E. Entity Recognition: Sherlock Meta Category Assignment

The Sherlock model takes a horizontal list of values as input. From each row, meta category assignment is done by extracting and analyzing 1588 features ranging from cardinality, unique identifiers, semantic content and character distributions [2]. The output for each column is then used to look up associated operators which can be combined with the variables to generate questions.

The role of these operators is to provide added semantic context to a variable. For example, when formulating a question to a geographic location like a city or a particular

county, like how many flu shots were administered within a region, the logical operator associated with the field should be "COUNT" which would translate into the "number of doses" on semantic conversion.

F. Semantic Distance Calculation for Question Prioritization Using Spacy

Semantic distance is a metric that is used to define the contextual likeness or similarity between two sets of words. The Spacy model in Python uses several techniques including part-of-speech (POS) tagging, Lemmatization, Inflection Morphology, and syntactic dependency parsing [7] to generate a similarity index for a pair of words. The higher the score for a pair of words, the more contextually similar they are considered.

In our program, we leverage these features to determine the relevance of each column by calculating the similarity index for each column with respect to all the keywords generated from the data description. The output stores the highest similarity index obtained for each column and the associated keywords in a list that is referenced for question prioritization later in the program.

G. Query Space Initiation and Question Formulation

The final module of our pipeline generates common language questions in two steps. First, we create a question string that concatenates 2 columns from the dataset together along with any one of the logical operators associated with the assigned meta category.

Second, we convert this string into a natural language question by replacing each column name with its description from the data dictionary and choosing the conversion template based on the operator associated with the second column. This is discussed in detail ahead.

H. Using Structured Statement Strings to Generate Question Combinations

The structured string that is used to generate the question series is designed in the form of a SELECT SQL statement using the following rules.

The first column selected either has the highest similarity index or one of the following associated meta-categories: Address, Location, county, age, gender, race, collection, category, result, city, club, year, day. The second column is randomly selected from the remaining list.

The meta category of the second column is checked, and an operator is randomly selected from the list of operators relevant to its semantic type. For example, if the second column has a meta category of "address", the operator selected could be one of the following – not equal to, equal to, count. Finally, a data value is randomly selected from the second column.

I. Semantic Conversion of Questions

The final phase in the pipeline is the conversion of the question string into semantically correct, common language questions. This involves first, substituting the column name with the column description, followed by selecting the question template for conversion by looping through a permutation of hardcoded combinations, based on the operator and meta category type.

The program can generate a series of up to 20 questions by randomly varying the question string parameters. Questions are generated in the following order of priority. The first two questions in a set will contain the column name with the highest similarity index. Next, cases where the second column is a numeric meta-category (rank, age, birth date, day, region, symbol) that have the comparison operators (min, max, average, count, sum) are given priority.

A set of 5 questions are generated in a single run with a total of 20 questions that can be created, for a single dataset. We are also testing various paraphrasers, such as Pegasus [8], to improve the quality of our output.

Examples of what our current output looks are shown below:

For a dataset that reports the "Prescription Related Claims of Mothers with Substance Use by Recipient County" [9].

Names of columns within the dataset: ['TOTAL PRESCRIPTIONS', 'TOTAL PRESCRIPTION COST', 'TOTAL PRESCRIPTIONS PREBIRTHING EVENT', 'PRESCRIPTION COST PREBIRTHING EVENT', 'PREBIRTHING MEDIAN COST', 'TOTAL PRESCRIPTIONS POSTBIRTHING EVENT', 'PRESCRIPTION COST POSTBIRTHING EVENT', 'POSTBIRTHING MEDIAN COST', 'COUNTY']

The meta categories predicted by Sherlock in order of columns are: ['code', 'address', 'day', 'address', 'symbol', 'code', 'address', 'depth', 'county'].

The question strings would have this pattern:
1st column: COUNTY
2nd column: POSTBIRTHING_MEDIAN_COST.

Question String 1: ['SELECT', 'COUNTY', 'FROM', 'test', 'WHERE', 'POSTBIRTHING_MEDIAN_COST', 'Minimum', 12.57]

Semantic Question 1: "What is the minimum Median cost of a prescription related to mothers two years after the delivery of the child with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery among all Mother's recorded county of residence as indicated closest to the delivery date?"

Paraphrased Question 1: "What is the minimum cost of a prescription related to mothers two years after the delivery of the child with a live birth during the time period of July 7, 2016 to July 7, 2017, and have a substance use related claim within 2 years prior to the delivery among all Mother's recorded county of residence".

1st column:
PRESCRIPTION_COST_POSTBIRTHING_EVENT
2nd column: PREBIRTHING_MEDIAN_COST

Question String 2: ['SELECT', PRESCRIPTION_COST _POSTBIRTHING_EVENT','FROM', 'test', 'WHERE', 'PREBIRTHING_MEDIAN_COST', 'Minimum', 11.92].

Semantic Question 2: "What is the minimum Median cost of a prescription related to mothers two years prior to the delivery of the child with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery among all Total cost of prescriptions prescribed to mothers two years after the delivery of the child with a live birth during the time period

07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery?"

Paraphrased Question 2: "What is the minimum cost of a prescription related to mothers two years prior to the delivery of the child with a live birth, and have a substance use related claim within 2 years prior to the delivery, among all Total cost of prescriptions prescribed to mothers two years?"

When we input a sample of the dataset into ChatGPT and ask it to generate questions, the result was a set of questions generated by combining a paraphrased likeness of the column names and associated values.

- Which county has the highest prescription cost per postbirthing event?
- How many counties are there in the dataset?
- What is the median cost of prebirthing medication in Blackford County?

However, when we input the dataset along with the data dictionary, the format of the questions shifts to focus on the text heavy descriptions and the questions start to resemble those generated by our module:

- What is the total number of prescriptions for mothers with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery?
- What is the overall total cost of prescriptions of mothers with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery?
- How many prescriptions were prescribed to mothers two years prior to the delivery of the child with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery?

In another example about "Hoosier Health and Well-being by County and Demographics" [10] where the column names are coded, the importance of the column description in interpreting the data becomes evident.

Keywords: ['demographics', 'health', 'county', 'hoosier'].

Column names: ['FEMALE COUNT', 'MALE COUNT', '18-22 COUNT', '23-27 COUNT', '28-32 COUNT', '33-37 COUNT', '38-42 COUNT', '43-47 COUNT', '48-52 COUNT', '53-57 COUNT', '58-62 COUNT', 'MARRIED COUNT', 'SEPARATED COUNT', 'DIVORCED COUNT', 'SINGLE COUNT', 'WIDOWED COUNT', 'BLACK COUNT', 'WHITE COUNT', 'CITIZEN COUNT', 'NO FORMAL ED COUNT', 'NO HS DIPLOMA COUNT', 'HS DIPLOMA COUNT', 'ATTENDING SCHOOL COUNT', 'QUESTION NUM', 'QUESTION COUNT', 'COUNTY COUNT', 'COUNTY', 'QUESTION LONG DESC', 'QUESTION SHORT DESC', 'ETL RUN TIMESTAMP'].

Meta categories predicted: ['address', 'day', 'ranking', 'ranking', 'day', 'day', 'ranking', 'ranking', 'ranking', 'ranking', 'ranking', 'ranking', 'ranking', 'address', 'ranking', 'position', 'address', 'address', 'ranking', 'code', 'day', 'rank', 'position', 'address', 'address', 'county', 'notes', 'name', 'address'].

Question Strings:
1st column: SINGLE_COUNT
2nd column: 28-32_COUNT

Question String 1: ['SELECT', 'SINGLE_COUNT', 'FROM', 'test', 'WHERE', '28-32_COUNT', 'Minimum', 351.0]

Semantic Question 1: "What is the minimum Count of survey respondents grouped by age as derived from provided date of birth among all Count of survey respondents grouped by provided marital status?"

Paraphrased Question 1: "What is the minimum Count of survey respondents grouped by age as a result of the provided date of birth among all respondents?"

1st column: FEMALE_COUNT
2nd column: ATTENDING_SCHOOL_COUNT

Question String 2: ['SELECT', 'FEMALE_COUNT', 'FROM', 'test', 'WHERE', 'ATTENDING_SCHOOL_COUNT', 'Maximum', 494.0]

Semantic Question 2: "Which Count of survey records where the respondent indicated gender was female has the highest Count of survey respondents grouped by self-reported as attending school at the time of the survey?"

Paraphrased Question 2: "Which Count of survey records has the highest number of respondents who said they were attending school at the time of the survey?"

For this example, ChatGPT generated only broad questions that could be formulated using easily readable column names and some descriptive attributes in the values:

- Which county has the highest number of males?
- What is the total number of people in the dataset?
- What is the total number of people who experienced not having enough money for food in the last 12 months?
- What is the total number of people who are citizens in the dataset?

Even after including the column names, most of the questions were very generalized, although a few provided some insight into the dataset:

- What is the meaning of the column 58-62_COUNT?
- What is the meaning of the column CITIZEN_COUNT?
- What is the range of the age group represented in the dataset?
- What is the total number of people represented in Bartholomew County?
- How many people in Blackford County have not completed high school?

A slightly different example is a financial dataset about resource allocation related to "COVID-19 Funds Transparency" [11].

K**eywords**: ['transparency', 'covid', 'funds'].

Column names: ['FUND SOURCE', 'BUSINESS UNIT', 'AWARD', 'EXPENSE', 'REMAINING', 'DESCRIPTION', 'FEDERAL AWARDING AGENCY', 'STATE AGENCY NAME'].

Meta categories predicted: ['day', 'symbol', 'elevation', 'elevation', 'elevation', 'collection', 'collection', 'collection']

Question strings:
1st column: FEDERAL AWARDING AGENCY

2nd column: EXPENSE

Question String 1: [SELECT', 'FEDERAL AWARDING AGENCY', 'FROM', 'test', 'WHERE', 'EXPENSE', 'Minimum', 0.0]

Semantic Question 1: "What is the minimum $ amount for transaction among all Federal aid agency name?"

Paraphrased Question 1: "What is the minimum amount for a transaction among all Federal aid agency names?"

1st column: STATE AGENCY NAME
2nd column: AWARD

Question String 2: [SELECT', 'STATE AGENCY NAME', 'FROM', 'test', 'WHERE', 'AWARD', 'Minimum', 1068762.0]

Semantic Question 2: "Which Agency name has the least Spend limit against funding source for BU/project?"

Paraphrased Question 2: "Which agency has the lowest spend limit against funding sources?"

In this case, ChatGPT delved heavily into the column values to generate questions from the dataset alone.

- Which federal awarding agency provided funds for the 2020 TIII-Congregate Meals?
- How much was awarded under the 2020 Pandemic Unemployment Assistance Implementation Grants Admin (PUA)
- Which state agency received funds under the 2020 Cooperative Agreement for Emergency Response: Public Health Crisis Response?

Including the column description, did not change the pattern of the questions in this case.

- What is the total amount of funding received by the State Department of Health for emergency response in 2020?
- How much money was allocated for Congregate Meals under the Families First Coronavirus Response Act?
- What is the purpose of the Pandemic Unemployment Assistance Implementation Grants Admin?

From the results, we can see that our program consistently produces factual natural language questions that shed light on the information within a dataset. Comparing our results with ChatGPT, we can see several similarities in the patterns. Since ChatGPT focuses more heavily on the text heavy section of a submission, the pattern shifts from using paraphrased column names to focusing on dataset values or column descriptors when they have larger text content, that can be used to independently formulate grammatically sound sentences.

III. Discussion, Limitations, Future Work

In this paper, we have discussed an approach that enables the automated generation of factual questions from a given dataset. We take a dataset, its data dictionary and description as inputs and generate a prioritized set of questions by combining columns within the dataset with semantically relevant operators and converting this string into a natural language sentence using predefined templates and substituting column names with their descriptions. We also use a paraphraser to improve contextual quality. Our approach produced relevant questions for several public health datasets, whose answers could be found within the data.

Comparison with ChatGPT shows how the AI adopts different approaches depending on the content. However, as we have seen in the above examples, this has the disadvantage of generating vague questions when the fields definitions are not very enlightening from a descriptive standpoint. Since many statistical datasets have coded or abbreviated column names, this was the primary logic behind our rationale of substituting field descriptions for column names during the semantic conversion process.

A contribution of our approach is attending to a particularly important data type for public questions: geo-identifiers. Our application of the Sherlock package indicated that it improved upon prior models by automating detection of semantic types by including a greater number of data columns with high performance in predicting meaningful classifications. Interestingly, since most datasets available in the public domain include zip codes as a geographic identifier, Sherlock is particularly successful in identifying this geographic identifier, which can be meaningful in considering event attendance rates, for events such as for sporting events. However, since zip codes were developed to aid postal workers in delivering mail to physical addresses, this identifier is not particularly relevant in official datasets.

Most official statistics are reported within geographic units other than zip codes. Smaller geographic units are designed to approximate social and relational spaces, such as neighborhoods, and larger units align with geopolitical units, such as counties, metropolitan areas, and states [12]. To broaden the applications of semantic type detection in advancing big data analytics, this project paid particular attention to the geographic identifiers in official statistics by initially advancing a separate module for Federal Information Processing Series (FIPS) data identifiers, independent of Sherlock. As a particular instance of a broader set of administratively constructed systems, the FIPS classification was developed to aggregate smaller geographies into large units without duplication [13]. Therefore, the FIPS units are key in harnessing the insights that public datasets can offer to citizens since geopolitical units have governance, elected officials, and administrative units related to the issues embedded in the data, e.g., public health.

A. Limitations

While the applications of our program are promising, there are some limitations in the current protype that need to be addressed. First, our approach can handle several datasets within a particular public data hub, since our questions template was designed with these use cases, but we are limited in the scope of datasets we can process.

Secondly, the accuracy of the questions our program generates depends on several factors. Sherlock is a pre-trained heuristic neural network model that depends on repeated input to improve its predictions. We depend on this prediction to correctly pair operators with columns, which in turn affects our ability to correctly combine diverse semantic types. However, Sherlock is still limited in the variety of data it can correctly identify. We have observed several instances of it misidentifying important datatypes, such as datetime and float based numeric values being classified as addresses. Additionally, Sherlock is also inconsistent in its prediction of

FIPS and other numeric data. With different datasets, FIPS was classified as an address or a symbol, depending on whether the datatype detected was a float or an integer. Additionally, there were several instances where numeric data was misclassified as addresses if the bulk of the data were 3-5 digit numbers that matched geo-identifiers.

Finally, the overall quality of our semantic conversion needs improvement. Our initial results with Pegasus showed that the ability of a paraphraser to improve the quality of our questions is dependent on the readability of the field descriptions available in the data dictionary. The more technical the description, the more grammatical and logical construction of the final question is affected. In such cases, all paraphrasers can do is rearrange the sentence, which may or may not improve its overall quality. To this end, we will be expanding the sentence libraries we work with, to determine if this can help improve the final output. Since ChatGPT encounters similar difficulties, as is evident from the second example we discussed, this may not be a problem that can be completely resolved.

B. *Future Work*

To mitigate the first limitation in future work, we are working towards building a dynamic template set that would be able to combine specific sets of semantic types together and expand on the associated range of operators to generate meaningful questions. This would facilitate scaling to include a larger set of publicly available datasets in future phases of this project. This includes developing a user interface that will facilitate knowledge workers in accessing datasets, while also collecting data regarding the inquiries requested. In the long-term, these data could be utilized to provide users with ratings regarding the most frequently asked questions, by recommending the type of questions asked by specific types of users, and so on.

Second, future steps in this project are to continue testing the inaccuracies that Sherlock generates to find solutions to circumvent its mistyping of important meta categories such as datetime and addresses. To circumvent this, we are in the process of implementing regex checks for date formats so that we can convert such values into formats recognizable by Sherlock. Our initial attempts have shown that even after date conversion, Sherlock is unable to correctly recognize a converted date in all instances. For FIPS codes, we also plan to implement a conversion snippet which will check for county level FIPS codes and convert these to County names. Since Sherlock does a better job of recognizing County names correctly, we hope that this will improve the accuracy of our predictions. Future work in this direction will include expanding the diversity of the datasets we test and devising additional feasible solutions that can be integrated into future iterations of our pipeline.

As an added step, we may also consider experimenting with models such as SATO [14] which looks at column names along with their values to further contextualize semantic type detection.

In conclusion, improving the overall quality of the output of our program will be a multistep process that will require improving semantic type recognition, defining more combination of semantically diverse datatypes can be combined to generate logically correct questions and expanding the templates that are used for conversion. We have discussed several approaches we are exploring to this end, in this manuscript. Another aspect to consider is determining whether cleaning a dataset to remove non-numeric values from numeric columns is necessary. In our study of ChatGPT's methodology of generating questions, we also used datasets with null and suppressed values to observe how this would affect question generation since such data could be used to identify areas of low incidence and/or statistical significance. Our results showed that ChatGPT was able to successfully generate questions using such data which proved to be factually relevant in the exploration of the dataset. Therefore, another potential approach we could consider as our program evolves, is defining branch-off points where users may be given a choice to decide if they would like to receive results using a cleaned dataset or the original version. This would also require us to significantly reconstruct the current structure of our question templates. Success in this endeavor however, may result in a significant augmentation of our ability to delve into a dataset to generate statistically relevant questions.

C. *Broader Significance*

This project contributes to broader efforts to advance concept learning from semantic categories using natural language parsing and finds further support for NLP advancing generalization and accessibility through supervised deep learning [1]. Many curiosities held by knowledge workers and everyday people are advanced through the prompting of related inquiries. For example, a knowledge worker interested in public health issues could begin their inquiry by exploring the costs of prescription medications in relation to birth rates within specific locations, such as counties. After examining an available dataset, and the questions the dataset is prepared to answer, this knowledge worker could be prompted to ask related questions, such as what the costs of prescription medications were in the years prior to and after the selected year, or in locations that are contingent to the selected county. To respond to this inquiry with fact-based information, it is necessary to parse data within relevant and semantically near categories. Additionally, the project advances on the prior work which developed the deep learning tools, Sherlock and Sato [2; 14], by applying this tool in service to big data analytics that are generated by the public sector and have the capability to inform knowledge workers and the public more generally [15].

REFERENCES

[1] S. Srivastava, I. Labutov, and T. Mitchell, "Joint Concept Learning and Semantic Parsing from Natural Language Explanations," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1527–1536. doi: 10.18653/v1/D17-1161.

[2] M. Hulsebos et al., "Sherlock: A Deep Learning Approach to Semantic Data Type Detection," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, in KDD '19. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 1500–1508. doi: 10.1145/3292500.3330993.

[3] "Indiana Data Hub." https://hub.mph.in.gov/ (accessed Apr. 28, 2023).

[4] ChatGPT, response to author query. OpenAI [Online]. https://chatgpt.pro/ (accessed Apr. 28, 2023)

[5] M. Kelechava, "Using LDA Topic Models as a Classification Model Input," Medium, Aug. 06, 2020. https://towardsdatascience.com/unsupervised-nlp-topic-models-as-a-supervised-learning-input-cf8ee9e5cf28 (accessed Apr. 28, 2023).

[6] R. Rehurek, "gensim: Python framework for fast Vector Space Modelling." [OS Independent]. Available: https://pypi.org/project/gensim/ (accessed: Apr. 28, 2023).

[7] “spaCy Linguistic Features.” (n.d.) https://spacy.io/usage/linguistic-features (accessed Apr. 28, 2023).

[8] “Pegasus.” https://huggingface.co/docs/transformers/model_doc/pegasus (accessed Apr. 28, 2023).

[9] “Prescription Related Claims of Mothers with Substance Use by Recipient County.” The Indiana Data Hub. https://hub.mph.in.gov/dataset/prescription-related-claims-of-mothers-with-substance-use-by-recipient-county. (accessed on April 28, 2023).

[10] “Hoosier Health and Well-being By County and Demographics.” https://hub.mph.in.gov/dataset/hoosier-health-and-well-being-by-county-and-demographics. (accessed on April 28, 2023).

[11] “COVID-19 Funds Transparency.” https://hub.mph.in.gov/dataset/covid-19-funds-transparency/resource/ecd4f142-6568-4f2b-88d3-73d5cd39f2a7 . (accessed on April 28, 2023).

[12] US Census Bureau. (2021). “Understanding Geographic Identifiers (GEOIDs): What geographic identifiers are, how they are formed and what they are used for plus details on the differences between FIPS and GNIS codes,” United States Census Bureau, https://www.census.gov/programs-surveys/geography/guidance/geo-identifiers.html (accessed Apr. 28, 2023).

[13] “US Census Bureau Geographic Entities and Concepts,” [Online]. Available: https://www.census.gov/content/dam/Census/data/developers/geoareaconcepts.pdf (accessed Apr. 28, 2023).

[14] D. Zhang, M. Hulsebos, Y. Suhara, Ç. Demiralp, J. Li, and W.-C. Tan, “Sato: contextual semantic type detection in tables,” Proc. VLDB Endow., vol. 13, no. 12, pp. 1835–1848, Aug. 2020, doi: 10.14778/3407790.3407793.

[15] R. Connelly, C. J. Playford, V. Gayle, and C. Dibben, “The role of administrative data in the big data revolution in social science research,” Social Science Research, vol. 59, pp. 1–12, Sep. 2016, doi: 10.1016/j.ssresearch.2016.04.015.