Prompt Development(Different Versions)

Earlier Versions of Prompt

```
Consider the below JSON array of objects describing low-level analy
tic tasks as a list of their “name”, “description” and “pro forma a
bstract" :
[
    {
        "name": "Correlation",
        "description": "Given a set of data cases and two attribute
s, determine useful relationships between the values of those attri
butes.",
        "pro forma abstract": "What is the correlation between attr
ibutes X and Y over a given set S of data cases?",
        "examples": ["Is there a correlation between carbohydrates
and fat?", "Do different genders have a preferred payment method?",
"What is the relationship between budget and gross?"],
        "Attributes":["Quantitative", "x", "Quantitative", "x", {"N
ominal", "Ordinal", "Quantative", "Temporal"}],
        "Attributes description":"encodings: x, y, color, size, col
umn, row, column to visualize up to three attributes at a time. "x"
indicates that the first attribute is used for the horizontal or x-
axis of the visualization. similarly the second "x" is used for the
vertical or y-axis of the visualization. Attributes within the curl
y brackets {} are optional.",
        "Visualization Recommendation":["Scatterplot"]
    },
    {
```

```
        "name": "Derived Value",
        "description": "Given a set of data cases, compute an aggre
gate numeric representation of those data cases.",
        "pro forma abstract": "What is the value of aggregation fun
ction F over a given set S of data cases?",
        "examples": ["What is the average calorie content of Post c
ereals?", "What is the gross income of all stores combined?", "How
many manufacturers of cars are there?"],
        "Attributes":["Nominal", "x", "Ordinal", "x", "Quantitativ
e", {"Nominal","Ordinal", "Quantative", "Temporal"}],
        "Attributes description":"encodings: x, y, color, size, col
umn, row, column to visualize up to three attributes at a time. "x"
indicates that the first attribute is used for the horizontal or x-
axis of the visualization. similarly the second "x" is used for the
vertical or y-axis of the visualization. Attributes within the curl
y brackets {} are optional.",
        "Visualization Recommendation":["Bar Chart"]
    },
    {
        "name": "Filter",
        "description": "Given some concrete conditions on attribute
values, find data cases satisfying those conditions.",
        "pro forma abstract": "Which data cases satisfy conditions
{A, B, C...}?",
        "examples": ["What Kellogg's cereals have high fiber?", "Wh
at comedies have won awards?", "Which funds underperformed the SP-5
00?"],
        "Attributes":["Quantitative", "x", "Nominal", "x", "Ordina
l", {"Nominal", "Ordinal", "Quantative", "Temporal"}, "x", {"Quanti
tative"}],
        "Attributes description":"encodings: x, y, color, size, col
umn, row, column to visualize up to three attributes at a time. "x"
indicates that the first attribute is used for the horizontal or x-
axis of the visualization. similarly the second "x" is used for the
vertical or y-axis of the visualization. Attributes within the curl
y brackets {} are optional.",
        "Visualization Recommendation": ["Strip Plot", "Histogram",
"Bar Chart", "Heatmap"]
```

```
    },
    {
        "name": "Distribution",
        "description": "Given a set of data cases and a quantitativ
e attribute of interest, characterize the distribution of that attr
ibute’s values over the set",
        "pro forma abstract": "What is the distribution of values o
f attribute A in a set S of data cases?",
        "examples": ["What is the age distribution of shoppers?",
"What is the distribution of carbohydrates in cereals?"],
        "Attributes":["Temporal", "x", {"Quantitative"}, "x", {"Nom
inal","Ordinal"}],
        "Attributes description":"encodings: x, y, color, size, col
umn, row, column to visualize up to three attributes at a time. "x"
indicates that the first attribute is used for the horizontal or x-
axis of the visualization. similarly the second "x" is used for the
vertical or y-axis of the visualization. Attributes within the curl
y brackets {} are optional.",
        "Visualization Recommendation":["Line Chart"]
    }
]

Using the above definitions, classify the below natural language qu
eries into the respective analytic tasks they map to. List the inpu
t query along with the “name” of the mapped tasks and attributes. R
ecommend the visualization type and add explanations on how you det
ermined the task from the input query, not the input query itself.
Also return the visualization recommendation in the form of a Vega-
Lite specification that reads data from the url:https://raw.githubu
sercontent.com/nl4dv/nl4dv/master/examples/assets/data/movies-w-yea
r.csv. Below is a sample dataset (in the CSV format) as reference:

Title,Worldwide Gross,Production Budget,Release Year,Content Ratin
g,Running Time,Genre,Creative Type,Rotten Tomatoes Rating,IMDB Rati
ng
From Dusk Till Dawn,25728961,20000000,1996,R,107,Horror,Fantasy,63,
7.1
Broken Arrow,148345997,65000000,1996,R,108,Action,Contemporary Fict
```

```
ion,55,5.8
City Hall,20278055,40000000,1996,R,111,Drama,Contemporary Fiction,5
5,6.1
Happy Gilmore,38623460,10000000,1996,PG-13,92,Comedy,Contemporary F
iction,58,6.9
Fargo,51204567,7000000,1996,R,87,Thriller,Contemporary Fiction,94,
8.3

Note that each query can have more than one attribute, task inferen
ces, and visualization recommendations. The input query is:

Show how content ratings are associated with gross receipts by genr
e.
```

```
Consider the below JSON array of objects describing low-level analyti
[
{
"name": "Correlation",
"description": "Given a set of data cases and two attributes, determi
"pro forma abstract": "What is the correlation between attributes X a
"examples": ["Is there a correlation between carbohydrates and fat?",
"Attributes":["Quantitative", "x" ,"Quantitative", "x" ,{"Nominal","C
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Scatterplot"]
},
{
"name": "Derived Value",
"description": "Given a set of data cases, compute an aggregate numer
"pro forma abstract": "What is the value of aggregation function F ov
"examples": ["What is the average calorie content of Post cereals?",
"Attributes":["Nominal", "x" ,"Ordinal", "x" ,"Quantitative",{"Nomina
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Bar Chart"]
},
{
```

```
"name": "Filter",
"description": "Given some concrete conditions on attribute values, f
"pro forma abstract": "Which data cases satisfy conditions {A, B, C..
"examples": ["What Kellogg's cereals have high fiber?", "What comedie
"Attributes":["Quantitative", "x" ,"Nominal", "x" ,"Ordinal",{"Nomina
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Strip Plot", "Histogram", "Bar Chart
},
{
"name": "Distribution",
"description": "Given a set of data cases and a quantitative attribut
"pro forma abstract": "What is the distribution of values of attribut
"examples": ["What is the age distribution of shoppers?", "What is th
"Attributes":["Temporal", "x",{"Quantitative"}, "x" ,{"Nominal","Ordi
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Line Chart"]
}
]

Using the above definitions, classify the below natural language quer

Title,Worldwide Gross,Production Budget,Release Year,Content Rating,R
From Dusk Till Dawn,25728961,20000000,1996,R,107,Horror,Fantasy,63,7.
Broken Arrow,148345997,65000000,1996,R,108,Action,Contemporary Fictio
City Hall,20278055,40000000,1996,R,111,Drama,Contemporary Fiction,55,
Happy Gilmore,38623460,10000000,1996,PG-13,92,Comedy,Contemporary Fic
Fargo,51204567,7000000,1996,R,87,Thriller,Contemporary Fiction,94,8.3

Convert the response into the following format:

for attribute Map do it for all the attributes detected:
{'Attribute': {'Attribute name': 'Worldwide Gross',
  'queryPhrase': ['fill in with queryPhrase detected'],
  'inferenceType': 'implicit/explicit',
  'matchScore': 0.9,
  'metric': ['attribute_similarity_match'],
  'isLabel': False/True,
  'encode': False/True,
```

```
  'isAmbiguous': False/True,
  'ambiguity': [],
  'meta': {'score': score out of 100,
   'dataType': 'Quantitative/Qualitative/Nominal/Ordinal/Temporal',
   'confidence': score out of 100}},
 }}}

for task Map do it for all the task detected:
{'task detected': [{'task Name': 'fill in with the task name detected
   'queryPhrase': 'query phrase detected',
   'matchScore': what is the match score out of 1,
   'attributes': ['fill in with the attribute detected'],
   'inferenceType': 'implicit/explicit',
   'isAttrAmbiguous': False/True,
   'isValueAmbiguous': False/True,
  },
  ]}

for the Visualization list do it for all the recommended visualizatio

[{'score': 5.39,
  'scoreObj': {'by_attributes': 1.8, 'by_task': 2, 'by_vis': 0},
  'confidenceObj': {'attributed detected': score of confidence, 'attr
  'attributes': ['fill in as a list of attributed detected'],
  'visType': ,
  'tasks': ['fill in as a list of task detected'],
  'inferenceType': 'implicit/explicit',
  add the vegalite specification for the visualization here for all t

Each Query can have more than one task and visualization type they ca

1. Show me the production budget against the worldwide gross split by
```

```
Consider the below JSON array of objects describing low-level analyti

[
{
"name": "Correlation",
"description": "Given a set of data cases and two attributes, determi
"pro forma abstract": "What is the correlation between attributes X a
"examples": ["Is there a correlation between carbohydrates and fat?",
"Attributes":["Quantitative", "x" ,"Quantitative", "x" ,{"Nominal","C
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Scatterplot"]
},
{
"name": "Derived Value",
"description": "Given a set of data cases, compute an aggregate numer
"pro forma abstract": "What is the value of aggregation function F ov
"examples": ["What is the average calorie content of Post cereals?",
"Attributes":["Nominal", "x" ,"Ordinal", "x" ,"Quantitative",{"Nomina
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Bar Chart"]
},
{
"name": "Filter",
"description": "Given some concrete conditions on attribute values, f
"pro forma abstract": "Which data cases satisfy conditions {A, B, C..
"examples": ["What Kellogg's cereals have high fiber?", "What comedie
"Attributes":["Quantitative", "x" ,"Nominal", "x" ,"Ordinal",{"Nomina
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Strip Plot", "Histogram", "Bar Chart
},
{
"name": "Distribution",
"description": "Given a set of data cases and a quantitative attribut
"pro forma abstract": "What is the distribution of values of attribut
"examples": ["What is the age distribution of shoppers?", "What is th
"Attributes":["Temporal", "x",{"Quantitative"}, "x" ,{"Nominal","Ordi
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Line Chart"]
```

```
}
]

Using the above definitions, classify the below natural language quer

Title,Worldwide Gross,Production Budget,Release Year,Content Rating,R
From Dusk Till Dawn,25728961,20000000,1996,R,107,Horror,Fantasy,63,7.
Broken Arrow,148345997,65000000,1996,R,108,Action,Contemporary Fictio
City Hall,20278055,40000000,1996,R,111,Drama,Contemporary Fiction,55,
Happy Gilmore,38623460,10000000,1996,PG-13,92,Comedy,Contemporary Fic
Fargo,51204567,7000000,1996,R,87,Thriller,Contemporary Fiction,94,8.3

Detect any attributes, tasks and visualization in the dataset that th
Put each attribute into the attributeMap, task into taskMap and Visua

Furthermore, NL queries can also use ambiguous language with partial


Here is the JSON object that the response should be returned as:
{
    "classified_task": <add list of classified tasks here>,
    "query": <add NL query that is being parsed here>,
    "query_raw": <add NL query that is being parsed here>,
    "dataset": <add dataset URL here>,
    "vlSpec": <add generated Vegalite specification here. Make sure t
    "attributeMap": {
        <dataset column that was detected (should have same value as
        "name": <dataset column that was detected>,
        "queryPhrase": [<keywords found in query that were used to de
        "encode": <boolean value depending on if the attribute appear
        }
        "metric": <[Can be one of two values: "attribute_exact_match"
        "inferenceType": <Can be one of two values: explicit or Impli
        "isAmbiguous": <Can be either True or False. Set the field to
        "ambiguity": [<Populate this list with all the different attr
       },
        "taskMap": {
        <query task that was detected (should have same value as "nam
```

```
        "name": <query task that was detected>,
        "queryPhrase": [<keywords found in query that were used to de
        "encode": <boolean value depending on if the task appears on
        }
        "metric": <[Can be one of two values: "attribute_exact_match"
        "inferenceType": <Can be one of two values: explicit or Impli
        "isAmbiguous": <Can be either True or False. Set the field to
        "ambiguity": [<Populate this list with all the different task
      },
        "visList": {
        <Visualization Recommendation that was detected for the query
        'attributes': [list of attributes detected],
        "queryPhrase": [<keywords found in query that were used to de
        "visType": None,
        'tasks': [add the list of tasks detected here for example lis
        "inferenceType": <Can be one of two values: explicit or Impli
        'vlSpec': add the vegalite specification of the visualization
      }
}

Only show the JSON object as output. Please do not add any extra pros

Relation between worldwide gross and production budget for each major
```

11/03/23

```
Consider the below JSON array of objects describing low-level analyti

[
{
"name": "Correlation",
"description": "Given a set of data cases and two attributes, determi
"pro forma abstract": "What is the correlation between attributes X a
"examples": ["Is there a correlation between carbohydrates and fat?",
"Attributes":["Quantitative", "x" ,"Quantitative", "x" ,{"Nominal","C
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Scatterplot"]
```

```
},
{
"name": "Derived Value",
"description": "Given a set of data cases, compute an aggregate numer
"pro forma abstract": "What is the value of aggregation function F ov
"examples": ["What is the average calorie content of Post cereals?",
"Attributes":["Nominal", "x" , "Ordinal", "x" , "Quantitative",{"Nomi
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Bar Chart"]
},
{
"name": "Filter",
"description": "Given some concrete conditions on attribute values, f
"pro forma abstract": "Which data cases satisfy conditions {A, B, C..
"examples": ["What Kellogg's cereals have high fiber?", "What comedie
"Attributes":["Quantitative", "x", "Nominal", "x", "Ordinal",{"Nomina
"Attributes description": "encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Strip Plot", "Histogram", "Bar Chart
},
{
"name": "Distribution",
"description": "Given a set of data cases and a quantitative attribut
"pro forma abstract": "What is the distribution of values of attribut
"examples": ["What is the age distribution of shoppers?", "What is th
"Attributes":["Temporal", "x",{"Quantitative"}, "x" ,{"Nominal","Ordi
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Line Chart"]
},
{
"name": "Sort",
"description": "Given a set of data cases, rank them according to som
"pro forma abstract": "What is the sorted order of a set S of data ca
"examples": ["Order the cars by weight", "Rank the cereals by calorie
"Attributes":["Temporal", "x",{"Quantitative"}, "x" ,{"Nominal","Ordi
"Attributes description":"encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Bar Chart"]
},
{
```

```
"name": "Find Extremum",
"description": "Find data cases possessing an extreme value of an att
"pro forma abstract": "What are the top/bottom N data cases with resp
"examples": ["What is the car with the highest MPG?", "What director/
"Attributes":["Temporal", "x",{"Quantitative"}, "x" ,{"Nominal","Ordi
"Attributes description": "encodings: x, y, color, size, column, row,
"Visualization Recommendation":["Scatter plot", "Box Plot"]
},
]

Using the above definitions, classify the below natural language quer

Title, Worldwide Gross, Production Budget, Release Year, Content Rati
From Dusk Till Dawn,25728961,20000000,1996, R,107, Horror, Fantasy,63
Broken Arrow,148345997,65000000,1996, R,108, Action, Contemporary Fic
City Hall,20278055,40000000,1996, R,111, Drama, Contemporary Fiction,
Happy Gilmore,38623460,10000000,1996,PG-13,92,Comedy,Contemporary Fic
Fargo,51204567,7000000,1996,R,87, Thriller, Contemporary Fiction,94,8

Detect any attributes, tasks, and visualization in the dataset that t
Put each attribute into the attributeMap, task into taskMap, and Visu

Furthermore, NL queries can also use ambiguous language with partial


Here is the JSON object that the response should be returned as:
{
    "classified_task": <add list of classified tasks here>,
    "query": <add NL query that is being parsed here>,
    "query_raw": <add NL query that is being parsed here>,
    "dataset": <add dataset URL here>,
    "vlSpec": <add generated Vegalite specification here. Make sure t
    "attributeMap": {
        <dataset column that was detected (should have the same value
        "name": <dataset column that was detected>,
        "queryPhrase": [<keywords found in query that were used to de
        "encode": <boolean value depending on if the attribute appear
        }
```

```
        "metric": <[Can be one of two values: "attribute_exact_match"
        "inferenceType": <Can be one of two values: explicit or Impli
        "isAmbiguous": <Can be either True or False. Set the field to
        "ambiguity": [<Populate this list with all the different attr
      },
        "taskMap": {
        <query task that was detected (should have same value as "nam
        "name": <query task that was detected>,
        "queryPhrase": [<keywords found in query that were used to de
        "encode": <boolean value depending on if the task appears on
        }
        "metric": <[Can be one of two values: "attribute_exact_match"
        "inferenceType": <Can be one of two values: explicit or Impli
        "isAmbiguous": <Can be either True or False. Set the field to
        "ambiguity": [<Populate this list with all the different task
      },
        "visList": {
        <Visualization Recommendation that was detected for the query
        'attributes': [list of attributes detected],
        "queryPhrase": [<keywords found in query that were used to de
        "visType": None,
        'tasks': [add the list of tasks detected here for example lis
        "inferenceType": <Can be one of two values: explicit or Impli
        'vlSpec': add the vegalite specification of the visualization
      }
}

Only show the JSON object as output. Please do not add any extra pros
```

12/18/23

Consider the below JSON array of objects describing low-level analytic tasks as a list of their “name”, “description” and “pro forma abstract" :

```
[ { "name": "Correlation", "description": "Given a set of data cases and two attributes, determine useful relationships between the values of those attributes.", "pro forma abstract": "What is the correlation between attributes X and Y over a given set S of data cases?", "examples": ["Is there a correlation between carbohydrates and fat?", "Do different genders have a preferred payment method?", "What is the relationship between budget and gross?"], "Attributes":["Quantitative", "x"
```

```
,"Quantitative", "x" ,{"Nominal","Ordinal","Quantative","Temporal"}], "Attributes
description":"encodings: x, y, color, size, column, row, column to visualize up to three attributes
at a time. "x" indicates that the first attribute is used for the horizontal or x-axis of the
visualization. similarly the second "x" is used for the vertical or y-axis of the visualization.
Attributes in the curly brackets {} are optional.", "Visualization Recommendation":["Scatterplot"] },
{ "name": "Derived Value", "description": "Given a set of data cases, compute an aggregate numeric
representation of those data cases.", "pro forma abstract": "What is the value of aggregation
function F over a given set S of data cases?", "examples": ["What is the average calorie content of
Post cereals?", "What is the gross income of all stores combined?", "How many manufacturers of cars
are there?"], "Attributes":["Nominal", "x" ,"Ordinal", "x" ,"Quantitative",
{"Nominal","Ordinal","Quantative","Temporal"}], "Attributes description":"encodings: x, y, color,
size, column, row, column to visualize up to three attributes at a time. "x" indicates that the first
attribute is used for the horizontal or x-axis of the visualization. similarly the second "x" is used
for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional.",
"Visualization Recommendation":["Bar Chart"] }, { "name": "Filter", "description": "Given some
concrete conditions on attribute values, find data cases satisfying those conditions.", "pro forma
abstract": "Which data cases satisfy conditions {A, B, C...}?", "examples": ["What Kellogg's cereals
have high fiber?", "What comedies have won awards?", "Which funds underperformed the SP-500?"],
"Attributes":["Quantitative", "x" ,"Nominal", "x" ,"Ordinal",
{"Nominal","Ordinal","Quantative","Temporal"},"x",{"Quantitative"}], "Attributes
description":"encodings: x, y, color, size, column, row, column to visualize up to three attributes
at a time. "x" indicates that the first attribute is used for the horizontal or x-axis of the
visualization. similarly the second "x" is used for the vertical or y-axis of the visualization.
Attributes in the curly brackets {} are optional.", "Visualization Recommendation":["Strip Plot",
"Histogram", "Bar Chart", "Heatmap"] }, { "name": "Distribution", "description": "Given a set of data
cases and a quantitative attribute of interest, characterize the distribution of that attribute's
values over the set", "pro forma abstract": "What is the distribution of values of attribute A in a
set S of data cases?", "examples": ["What is the age distribution of shoppers?", "What is the
distribution of carbohydrates in cereals?"], "Attributes":["Temporal", "x",{"Quantitative"}, "x" ,
{"Nominal","Ordinal"}], "Attributes description":"encodings: x, y, color, size, column, row, column
to visualize up to three attributes at a time. "x" indicates that the first attribute is used for the
horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-
axis of the visualization. Attributes in the curly brackets {} are optional.", "Visualization
Recommendation":["Line Chart"] }, { "name": "Sort", "description": "Given a set of data cases, rank
them according to some ordinal metric.", "pro forma abstract": "What is the sorted order of a set S
of data cases according to their value of attribute A?", "examples": ["Order the cars by weight",
"Rank the cereals by calories."], "Attributes":["Temporal", "x",{"Quantitative"}, "x",{"Nominal",
"Ordinal"}], "Attributes description": "encodings: x, y, color, size, column, row, column to
visualize up to three attributes at a time. "x" indicates that the first attribute is used for the
horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-
axis of the visualization. Attributes in the curly brackets {} are optional.", "Visualization
Recommendation":["Bar Chart"] }, { "name": "Find Extremum", "description": "Find data cases
possessing an extreme value of an attribute over its range within the data set.", "pro forma
abstract": "What are the top/bottom N data cases with respect to attribute A?", "examples": ["What is
the car with the highest MPG?", "What director/film has won the most awards?","What Robin Williams
film has the most recent release date?"], "Attributes":["Temporal", "x",{"Quantitative"}, "x" ,
{"Nominal","Ordinal"}], "Attributes description": "encodings: x, y, color, size, column, row, column
to visualize up to three attributes at a time. "x" indicates that the first attribute is used for the
horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-
axis of the visualization. Attributes in the curly brackets {} are optional.", "Visualization
Recommendation":["Scatter plot", "Box Plot"] }
]
```

Using the above definitions, classify the below natural language queries into the respective analytic tasks they map to. Return the visualization type in the form of

VegaLite specification where it reads data from url:https://raw.githubusercontent.com/nl4dv/nl4dv/master/examples/assets/data/movies-w-year.csv.

Title, Worldwide Gross, Production Budget, Release Year, Content Rating, Running Time, Genre, Creative Type, Rotten Tomatoes Rating, IMDB Rating
From Dusk Till Dawn,25728961,20000000,1996, R,107, Horror, Fantasy,63,7.1
Broken Arrow,148345997,65000000,1996, R,108, Action, Contemporary Fiction,55,5.8
City Hall,20278055,40000000,1996, R,111, Drama, Contemporary Fiction,55,6.1
Happy Gilmore,38623460,10000000,1996,PG-13,92,Comedy,Contemporary Fiction,58,6.9
Fargo,51204567,7000000,1996,R,87, Thriller, Contemporary Fiction,94,8.3

Detect any attributes, tasks and visualizations in the dataset that the provided query references, and place the detected dataset columns in the attributeMap, taskMap and visList key of the JSON below. Each Query can have more than one task and visualization type they can map to. A dataset attribute is defined as one of the column titles in the provided dataset. Each key in the "attributeMap" JSON should be populated with the extracted dataset column (e.g. "Worldwide Gross"). There can multiple attributes, tasks and visualizations that are detected, but make sure that each attribute, task and visualization in the attributeMap, taskMap and visList is unique.
Put each attribute into the attributeMap, task into taskMap and Visualization into visList JSON. There can also be multiple possible visualization specifications as well. For each possible visualization specification, you can add include a dictionary that has the required contents to the "visList" list.

Furthermore, NL queries can also use ambiguous language with partial references to data attributes. The attributeMap also includes an “isAmbiguous” field. The field can either be True or False. Set the field to True if the queryPhrase could refer to other attributes in the dataset. Otherwise set the field to False.
If there are ambiguous attributes detected, generate VegaLite specifications for each ambiguous attribute that encode the ambiguous attribute in the visualization. Also if there are no tasks detected in the NL query, infer the task that is best suited with the detected attributes' datatypes. Generate a visualization specification using this inferred task and detected attributes.

Here is the JSON object that the response should be returned as:
{
"query": <add NL query that is being parsed here>,
"query_raw": <add NL query that is being parsed here>,
"dataset": <add dataset URL here>,
"visList": [

{
'attributes': [list of attributes detected],
"queryPhrase": [<keywords found in query that were used to detect the taskMap and the recommended visualization>],
"visType": None,
'tasks': [add the list of tasks detected here for example list "Correlation","Derived Value","filter","Distribution" etc.],
"inferenceType": <Can be one of two values: explicit or Implicit. Set the value to explicit if the visualization's "queryPhrase" explicitly references a visualization type. Set the value to implicit otherwise.>
'vlSpec': add the vegalite specification of the visualization recommended here.
}
]
“attributeMap”: {
<dataset column that was detected (should have same value as “name”)>: {
“name”: <dataset column that was detected>,
“queryPhrase”: [<keywords found in query that were used to detect the dataset attribute>]
“encode”: <boolean value depending on if the attribute appears on either of the axes or color in the VegaLite specification. The boolean value should be output as true or false in all lowercase letters.>
}
“metric”: <[Can be one of two values: “attribute_exact_match” or “attribute_similarity_match”. Set the value to “attribute_exact_match” if the attribute was found directly in the query. Set the value to “attribute_similarity_match” if the query uses a synonym for the attribute.]>,
“inferenceType”: <Can be one of two values: explicit or Implicit. Set the value to explicit if the attribute’s “queryPhrase” references an attribute name. Set the value to implicit if the queryPhrase directly references values found in the attribute’s values.>
“isAmbiguous”: <Can be either True or False. Set the field to True if the queryPhrase could refer other attributes in the dataset. Otherwise set the field to False.>
“ambiguity”: [<Populate this list with all the different attributes in the dataset that the queryPhrase can refer to if isAmbiguous is set to True. Otherwise keep this list empty.]
}
"taskMap": {
<task that was detected>: [
{
"task": <task that was detected>,

“queryPhrase”: [<keywords found in query that were used to detect the task>],
"values": [<if the "filter" task was detected, put the filter value in here],
"attributes": [<populate with the attributes that the task is mapped to],
"operator": "<Can be one of IN, GT, EQ, AVG, SUM. GT is greater than. EQ is equals. GT and EQ are used for quantitative filters. IN is used for nominal filters. AVG and SUM are used for derived value tasks. SUM is for summation and AVG is for average. Keep the string empty otherwise."
"inferenceType": <Can be one of two values: explicit or Implicit. Set the value to explicit if the attribute’s “query phrase” references a task. Set the value to implicit if the visualization needs a certain task.>
}
]
}
"visType": <put visualization type that was explicitly specified in the query. Put the string "None" if vis type was not specified in the query.>
"visQueryPhrase": <keywords found in query that were used to detect the visType. Put the string "None" if vis type was not specified in the query.>
}
Only show the JSON object as output. Ensure that the JSON object is valid. Please do not add any extra prose in your response.

1/18/24

Consider the below JSON array of objects describing low-level analytic tasks (fundamental operations that users perform when interacting with data visualizations) as a list of their “name”, “description” and “pro forma abstract" :

`[

{

"name": "Correlation",

"description": "Given a set of data cases and two attributes, determine useful relationships between the values of those attributes.",

"pro forma abstract": "What is the correlation between attributes X and Y over a given set S of data cases?"("pro forma abstract" refers to a concise summary outlining the main elements for a given natural language query),

"examples": ["Is there a correlation between carbohydrates and fat?", "Do different genders have a preferred payment method?", "What is the relationship between budget and gross?"],

"Attributes":["Quantitative", "x" ,"Quantitative", "x" , {"Nominal","Ordinal","Quantative","Temporal"}],

"Attributes description":"encodings: x, y, color, size, column, row, column to visualize up to three attributes at a time. "x" indicates that the first attribute is used for the horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional.",

"Visualization Recommendation":["Scatterplot"]

},

{

"name": "Derived Value",

"description": "Given a set of data cases, compute an aggregate numeric representation of those data cases.",

"pro forma abstract": "What is the value of aggregation function F over a given set S of data cases?" ("pro forma abstract" refers to a concise summary outlining the main elements for a given natural language query),

"examples": ["What is the average calorie content of Post cereals?", "What is the gross income of all stores combined?", "How many manufacturers of cars are there?"],

"Attributes":["Nominal", "x" ,"Ordinal", "x" ,"Quantitative", {"Nominal","Ordinal","Quantative","Temporal"}],

"Attributes description":"encodings: x, y, color, size, column, row, column to visualize up to three attributes at a time. "x" indicates that the first attribute is used for the horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional.",

"Visualization Recommendation":["Bar Chart"]

},

{

"name": "Filter",

"description": "Given some concrete conditions on attribute values, find data cases satisfying those conditions.",

"pro forma abstract": "Which data cases satisfy conditions {A, B, C...}?" ("pro forma abstract" refers to a concise summary outlining the main elements for a given natural language query),

"examples": ["What Kellogg's cereals have high fiber?", "What comedies have won awards?", "Which funds underperformed the SP-500?”, “Show the visualization only for films that have grossed more than 100 million dollars”],

"Attributes":["Quantitative", "x" ,"Nominal", "x" ,"Ordinal", {"Nominal","Ordinal","Quantative","Temporal"},"x",{"Quantitative"}],

"Attributes description":"encodings: x, y, color, size, column, row, column to visualize up to three attributes at a time. "x" indicates that the first attribute is used for the horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional.",

"Visualization Recommendation":["Strip Plot", "Histogram", "Bar Chart", "Heatmap"]

},

{

"name": "Distribution",

"description": "Given a set of data cases and a quantitative attribute of interest, characterize the distribution of that attribute’s values over the set",

"pro forma abstract": "What is the distribution of values of attribute A in a set S of data cases?"("pro forma abstract" refers to a concise summary outlining the main elements for a given natural language query),

"examples": ["What is the age distribution of shoppers?", "What is the distribution of carbohydrates in cereals?"],

"Attributes":["Temporal", "x",{"Quantitative"}, "x" ,{"Nominal","Ordinal"}],

"Attributes description":"encodings: x, y, color, size, column, row, column to visualize up to three attributes at a time. "x" indicates that the first attribute is used for the horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional.",

"Visualization Recommendation":["Line Chart"]

},

{

"name": "Sort",

"description": "Given a set of data cases, rank them according to some ordinal metric.",

"pro forma abstract": "What is the sorted order of a set S of data cases according to their value of attribute A?" ("pro forma abstract" refers to a concise summary outlining the main elements for a given natural language query),

"examples": ["Order the cars by weight", "Rank the cereals by calories."],

"Attributes":["Temporal", "x",{"Quantitative"}, "x" ,{"Nominal","Ordinal"}],

"Attributes description":"encodings: x, y, color, size, column, row, column to visualize up to three attributes at a time. "x" indicates that the first attribute is used for the horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional.",

"Visualization Recommendation":["Bar Chart"]

},

{

"name": "Find Extremum",

"description": "Find data cases possessing an extreme value of an attribute over its range within the data set.",

"pro forma abstract": "What are the top/bottom N data cases with respect to attribute A?" ("pro forma abstract" refers to a concise summary outlining the main elements for a given natural language query),

"examples": ["What is the car with the highest MPG?", "What director/film has won the most awards?","What Robin Williams film has the most recent release date?"],

"Attributes":["Temporal", "x",{"Quantitative"}, "x" ,{"Nominal","Ordinal"}],

"Attributes description": "encodings: x, y, color, size, column, row, column to visualize up to three attributes at a time. "x" indicates that the first attribute is used for the horizontal or x-axis of the visualization. similarly the second "x" is used for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional.",

"Visualization Recommendation":["Scatter plot", "Box Plot"]

}’

`]`

Also consider the below JSON array of objects describing low-level operations for follow-up queries as a list of their “name”, “instruction”, and “examples”:

‘[

{

“name”: “Add Attribute”,

“instruction”: “Given a previous analytic specification, add the attribute that is detected in the natural language query to the previous specification.”,

“examples”: [“Add genre”, “Group by genre”, “Put genre as well”]

},

{

“name”:”Remove Attribute

“instruction”: “Given a previous analytic specification, remove the attribute that is detected in the natural language query to the previous specification.”,

“examples”: [“Remove genre”, “Show only production budget”]

},

{

“name”: “Replace Attribute”,

“instruction”: “Given a previous analytic specification, replace the attribute that is detected in the natural language query and is also in the previous analytic specification with the attribute that is detected in the natural language query, but is not in the previous specification.”,

“examples”: [“Replace gross with imdb rating”, “Show gross instead of imdb rating”]

},

{

“name”: “Add Task”,

“instruction”: “Given a previous analytic specification, add the task that is detected in the natural language query to the previous specification. The visualization specification should reflect this new task that is added from the natural language query. Also please keep the previous tasks from the previous specification in the new modified analytic specification.”,

“examples”: [“Sort by worldwide gross in the descending order”, “Show only horror movies”, “What are their average production budgets?”]

},

{

“name”: “Remove Task”,

“instruction”: “Given a previous analytic specification, remove the task that is detected in the natural language query to the previous specification.”,

“examples”: [“Undo the sorting”, “Undo the average”]

},

{“name”: “Replace task”,

“instruction”: “Given a previous analytic specification, replace the attribute that is detected in the natural language query and is also in the previous analytic specification with the attribute that is detected in the natural language query, but is not in the previous specification. Also please keep the other tasks from the previous specification that will not be replaced in the new modified analytic specification.”,

“examples”: [“Replace average with sum”, “Show me total instead of average”, “What about the lowest instead of highest?”]

},

{

“name”: “Replace vis”,

“instruction”: “Given a previous analytic specification, replace the visualization type that is detected in the natural language query and is also in the previous analytic specification with the visualization type that is detected in the natural language query, but is not in the previous specification.”,

“examples”: [“Replace the vis with a bar chart”, “As a line chart now”, “Show a tick plot instead”]

}

]’

If a field called Previous Analytic Specification appears below the natural language query, classify the below natural language query into the respective follow-up operations they map to. Utilize the previous analytic specification (including the attributeMap, taskMap, and visList) and modify this specification to reflect the changes that are specified and requested in the natural language query. Return the visualization type in the form of VegaLite specification where it reads data from url:https://raw.githubusercontent.com/nl4dv/nl4dv/master/examples/assets/data/movies-w-year.csv.

However, if there is no field called Previous Analytic Specification that is below the natural language query, then using the above definitions, classify the below natural language queries into the respective analytic tasks they map to. Return the visualization

type in the form of VegaLite specification where it reads data from url:https://raw.githubusercontent.com/nl4dv/nl4dv/master/examples/assets/data/movies-w-year.csv.

Title, Worldwide Gross, Production Budget, Release Year, Content Rating, Running Time, Genre,Creative Type, Rotten Tomatoes Rating,IMDB Rating

From Dusk Till Dawn,25728961,20000000,1996, R,107, Horror, Fantasy,63,7.1

Broken Arrow,148345997,65000000,1996, R,108, Action, Contemporary Fiction,55,5.8

City Hall,20278055,40000000,1996, R,111, Drama, Contemporary Fiction,55,6.1

Happy Gilmore,38623460,10000000,1996,PG-13,92,Comedy,Contemporary Fiction,58,6.9

Fargo,51204567,7000000,1996,R,87, Thriller, Contemporary Fiction,94,8.3

If there is no Previous Analytic Specification that appears below the query, detect any attributes, tasks and visualizations in the dataset that the provided query references, and place the detected dataset columns in the attributeMap, taskMap and visList key of the JSON below. Each Query can have more than one task and visualization type they can map to. A dataset attribute is defined as one of the column titles in the provided dataset. Each key in the "attributeMap" JSON should be populated with the extracted dataset column (e.g. "Worldwide Gross"). There can multiple attributes, tasks and visualizations that are detected, but make sure that each attribute, task and visualization in the attributeMap, taskMap and visList is unique. Put each attribute into the attributeMap, task into taskMap and Visualization into visList JSON. There can also be multiple possible visualization specifications as well. For each possible visualization specification, you can include a dictionary that has the required contents to the "visList" list.

Furthermore, NL queries can also use ambiguous language with partial references to data attributes. The attributeMap also includes an “isAmbiguous” field. The field can either be True or False. Set the field to True if the queryPhrase could refer to other attributes in the dataset. Otherwise set the field to False. If there are ambiguous attributes detected, generate VegaLite specifications for each ambiguous attribute that encode the ambiguous attribute in the visualization. Furthermore, if there are no tasks detected in the NL query, infer the task that is best suited with the detected attributes' datatypes. Generate a visualization specification using this inferred task and detected attributes.

If there is a “Filter” task detected in the natural language query, please add the “transform” property in VegaLite to the VegaLite specification. This action will apply the filter specified in the natural language query. (Link to VegaLite “transform” property: https://vega.github.io/vega-lite/docs/filter.html)

Here is the JSON object that the response should be returned as if and only if there is no Previous Analytic Specification attached to the natural language query:

{

"query": <add NL query that is being parsed here>,

"query_raw": <add NL query that is being parsed here>,

"dataset": <add dataset URL here>,

"visList": [

{

'attributes': [list of attributes detected],

"queryPhrase": [<keywords found in query that were used to detect the taskMap and the recommended visualization>],

"visType": None,

'tasks': [add the list of tasks detected here for example list "Correlation","Derived Value","filter","Distribution" etc.],

"inferenceType": <Can be one of two values: explicit or Implicit. Set the value to explicit if the visualization's "queryPhrase" explicitly references a visualization type. Set the value to implicit otherwise.>

'vlSpec': add the vegalite specification of the visualization recommended here.

}

]

“attributeMap”: {

<dataset column that was detected (should have same value as “name”)>: {

“name”: <dataset column that was detected>,

“queryPhrase”: [<keywords found in query that were used to detect the dataset attribute>]

“encode”: <boolean value depending on if the attribute appears on either of the axes or color in the VegaLite specification. The boolean value should be output as true or false in all lowercase letters.>

}

“metric”: <[Can be one of two values: “attribute_exact_match” or “attribute_similarity_match”. Set the value to “attribute_exact_match” if the attribute was

found directly in the query. Set the value to “attribute_similarity_match” if the query uses a synonym for the attribute.]>,

“inferenceType”: <Can be one of two values: explicit or Implicit. Set the value to explicit if the attribute’s “queryPhrase” references an attribute name. Set the value to implicit if the queryPhrase directly references values found in the attribute’s values.>

“isAmbiguous”: <Can be either True or False. Set the field to True if the queryPhrase could refer other attributes in the dataset. Otherwise set the field to False.>

“ambiguity”: [<Populate this list with all the different attributes in the dataset that the queryPhrase can refer to if isAmbiguous is set to True. Otherwise keep this list empty.]

}

"taskMap": {

<task that was detected>: [

{

"task": <task that was detected>,

“queryPhrase”: [<keywords found in query that were used to detect the task>],

"values": [<if the "filter" task was detected, put the filter value in here],

"attributes": [<populate with the attributes that the task is mapped to],

"operator": "<Can be one of IN, GT, EQ, AVG, SUM. GT is greater than. EQ is equals. GT and EQ are used for quantitative filters. IN is used for nominal filters. AVG and SUM are used for derived value tasks. SUM is for summation and AVG is for average. Keep the string empty otherwise."

"inferenceType": <Can be one of two values: explicit or Implicit. Set the value to explicit if the attribute’s “queryPhrase” references a task. Set the value to implicit if the visualization needs a certain task.>

}

]

}

"visType": <put visualization type that was explicitly specified in the query. Put the string "None" if vis type was not specified in the query.>

"visQueryPhrase": <keywords found in query that were used to detect the visType. Put the string "None" if vis type was not specified in the query.>

}

If there is a Previous Analytic Specification attached, then utilize the previous analytic specification (including the attributeMap, taskMap, and visList) and modify this specification to reflect the changes that are specified and requested in the natural language query. Then return this modified JSON object as output. The VegaLite specification should also change and a new visualization should be generated as a result of the modification. Also, in the modified JSON object, include a new parameter called “followupType”. This “followupType” should include the classified followup operation (add attribute, replace task, etc.) that was detected in the natural language query.

Please do not add any extra prose to your response. I only want to see the JSON output.