

Encoding Seed Instructions

Base Prompt

You are asked to generate diverse sets of instructions and outputs using PyBGPStream library.
Requirements:
1. Instructions should include analysis with record collection, MOAS (Multiple Origin AS) prefixes, Measuring AS path inflation,
...
Here are the parameters of PyBGPStream library, and the filter rules with their abbreviations:
1. project: The name of the project from which to collect BGP data. It accepts string values that list one or more projects or collectors separated by commas: "ris,route-views"
...
Here is a complete example of the instruction and the output:

Seed Instructions

Instruction: "Collect BGP update data for the target prefix (41.221.215.0/24) over a specified collection period ..."
Input: "Target Prefix: 41.221.215.0/24
Total Collection Period: 900 seconds (15 minutes)"
Output:

```
import pybgpstream
import pandas as pd
import time

target_prefix =
"41.221.215.0/24"
prefix_announcements_df =
pd.DataFrame(columns=['Time',
'AS Path', 'Count'])
withdrawals_df =
pd.DataFrame(columns=['Time',
'AS Path', 'Count'])...
```

Creating Batches of Prompts

IP Routing Analysis Instruction Pool

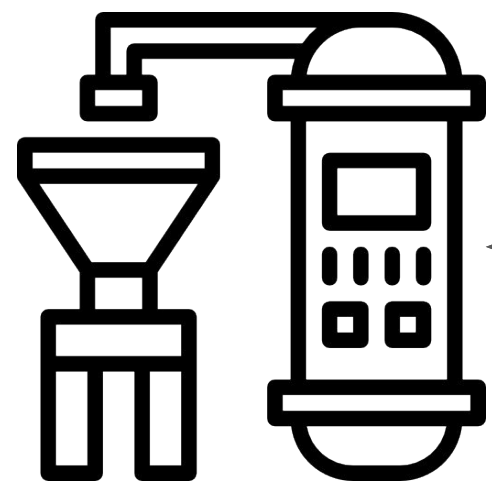
Feeding Prompts to OpenAI's PM



OpenAI's PM

Generating New Instructions

Filtering



Save

Discard

Instruction: "Perform time-series analysis of AS path length and preference value through live stream data sources from RouteViews suit of collectors."

Input: "Collectors: route-views2, route-views3, route-views4, ris-ripe
Record Type: updates"

Instruction: "Retrieve BGP update data for the specified IP prefix (41.221.215.0/24) over a designated collection period of 900 seconds, equivalent to 15 minutes. Within each 5-minute interval during the collection ..."

Input: "IP Prefix: 41.221.215.0/24; Total Collection Duration: 900 seconds (15 minutes)"

Rouge-L Score above 70%

No

Yes

Instruction: "Perform time-series analysis of AS path length and preference value through live stream data sources from RouteViews suit of collectors."

Input: "Collectors: route-views2, route-views3, route-views4, ris-ripe
Record Type: updates"

Output:

```
import pybgpstream
```

```
as_path_lengths = []
preference_values = []
```

```
stream = pybgpstream.BGPStream(
    from_time="2020-10-01 00:00:00",
    until_time="2020-10-31 00:00:00",
    collectors=["route-views2",
"route-views3", "route-views4",
"ris-ripe"],
    record_type="updates"
)
...
```