Advanced: Customizing Optimization

You can customize the optimization process for specific operations using the ``optimizer_config in your pipeline.

Global Configuration

The following options can be applied globally to all operations in your pipeline during optimization:

- `num_retries`: The number of times to retry optimizing if the LLM agent fails. Default is 1.
- `sample_sizes`: Override the default sample sizes for each operator type. Specify as a dictionary with operator types as keys and integer sample sizes as values.

Default sample sizes:

```
SAMPLE_SIZE_MAP = {
    "reduce": 40,
    "map": 5,
    "resolve": 100,
    "equijoin": 100,
    "filter": 5,
}
```

- `judge_agent_model`: Specify the model to use for the judge agent. Default is `gpt-4o-mini`.
- `rewrite_agent_model`: Specify the model to use for the rewrite agent. Default is `gpt-4o`.
- `litellm_kwargs`: Specify the litellm kwargs to use for the optimization. Default is `{}`.

Equijoin Configuration

- `target_recall`: Change the default target recall (default is 0.95).

Resolve Configuration

- `target_recall`: Specify the target recall for the resolve operation.

Reduce Configuration

- `synthesize_resolve`: Set to `False` if you definitely don't want a resolve operation synthesized or want to turn off this rewrite rule.

Map Configuration

- `force_chunking_plan`: Set to `True` if you want the the optimizer to force plan that breaks up the input documents into chunks.
- `plan_types`: Specify the plan types to consider for the map operation. The available plan types are:
- `chunk`: Breaks up the input documents into chunks (i.e., data decomposition).
- `proj_synthesis`: Synthesizes 1+ projections (i.e., task decomposition).
- `glean`: Synthesizes a glean plan (i.e., uses LLM as a judge to refine the output).

Example Configuration

Here's an example of how to use the `optimizer_config` in your pipeline:

```yaml
optimizer_config:
  rewrite_agent_model: gpt-4o-mini
  judge_agent_model: gpt-4o-mini
  litellm_kwargs:
    temperature: 0.5
  num_retries: 2
  sample_sizes:
    map: 10
    reduce: 50
  reduce:
    synthesize_resolve: false
  map:
    plan_types: # Considers all these plan types
      - chunk
      - proj_synthesis
      - glean

operations:
  - name: extract_medications
    type: map
    optimize: true
    recursively_optimize: true # Recursively optimize the map operation (i.e.,
optimize any new operations that are synthesized)
    # ... other configuration ...

  - name: summarize_prescriptions
    type: reduce
    optimize: true
```

```
    # ... other configuration ...
# ... rest of the pipeline configuration ...
```

This configuration will:

1. Retry optimization up to 2 times for each operation if the LLM agent fails.
2. Use custom sample sizes for map (10) and reduce (50) operations.
3. Prevent the synthesis of resolve operations for reduce operations.
4. Consider all plan types for map operations.