
OpenCL

Release v0.1.4

Shanghai AI Lab, SharkNLP

Mar 13, 2023

GETTING STARTED

1	Citation	3
1.1	Installation	3
1.1.1	Using Pip	3
1.1.2	Installation for local development	3
1.2	A Simple Example	4
1.2.1	Step 1: Load and prepare data	4
1.2.2	Step 2: Define the prompt template (Optional)	4
1.2.3	Step 3: Initialize the Retriever	4
1.2.4	Step 4: Initialize the Inferencer	4
1.2.5	Step 5: Inference and scoring	5
1.3	Tutorials List	5
1.4	DatasetReader	5
1.5	PromptTemplate	8
1.6	Inferencer	10
1.6.1	BaseInferencer	10
1.6.2	PPLInferencer	11
1.6.3	GenInferencer	13
1.6.4	CoTInferencer	16
1.7	Retriever	18
1.7.1	BaseRetriever	18
1.7.2	RandomRetriever	19
1.7.3	BM25Retriever	21
1.7.4	TopkRetriever	23
1.7.5	VotekRetriever	25
1.7.6	DPPRetriever	27
1.7.7	MDLRetriever	29
1.7.8	ZeroRetriever	31
2	Indices and tables	33
	Index	35



OpenICL is an open-source framework to facilitate research, development, and prototyping of in-context learning.

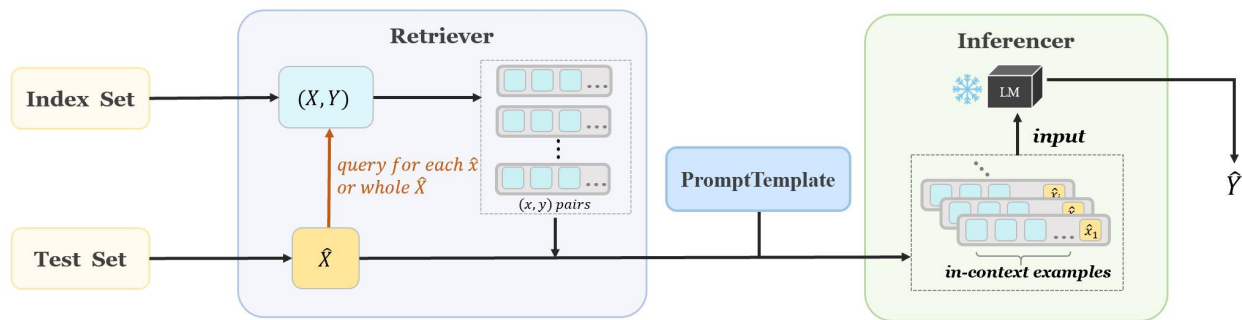


Fig. 1: Overview of the architecture in OpenICL.

It provides an easy interface for in-context learning, with many state-of-the-art retrieval and inference methods built in to facilitate systematic comparison of LMs and fast research prototyping. Users can easily incorporate different retrieval and inference methods, as well as different prompt instructions into their workflow.

Note: This project is under active development.

CITATION

If you find this repository helpful, feel free to cite our paper:

```
@article{wu2023openicl,  
title={OpenICL: An Open-Source Framework for In-context Learning},  
author={Zhenyu Wu, Yaoxiang Wang, Jiacheng Ye, Jiangtao Feng, Jingjing Xu, Yu Qiao,  
↪Zhiyong Wu},  
journal={arXiv preprint arXiv:2303.02913},  
year={2023}  
}
```

1.1 Installation

Note: OpenICL requires [Python 3.8+](#)

1.1.1 Using Pip

```
pip install openicl
```

1.1.2 Installation for local development

You can install the latest version of OpenICL from our [github repository](#).

```
git clone https://github.com/Shark-NLP/OpenICL  
cd OpenICL  
pip install -e .
```

1.2 A Simple Example

Following example shows you how to perform ICL on sentiment classification dataset. More examples and tutorials can be found at our [github repository](#).

1.2.1 Step 1: Load and prepare data

```
from datasets import load_dataset
from openicl import DatasetReader

# Loading dataset from huggingface
dataset = load_dataset('gpt3mix/sst2')

# Define a DatasetReader, with specified column names where input and output are stored.
data = DatasetReader(dataset, input_columns=['text'], output_column='label')
```

1.2.2 Step 2: Define the prompt template (Optional)

```
from openicl import PromptTemplate
tp_dict = {
    0: "</E>Positive Movie Review: </text>",
    1: "</E>Negative Movie Review: </text>"
}

template = PromptTemplate(tp_dict, {'text': '</text>'}, ice_token='</E>')
```

The placeholder `</E>` and `</text>` will be replaced by in-context examples and testing input, respectively.

1.2.3 Step 3: Initialize the Retriever

```
from openicl import TopkRetriever
# Define a retriever using the previous `DataLoader`.
# `ice_num` stands for the number of data in in-context examples.
retriever = TopkRetriever(data, ice_num=8)
```

Here we use the popular `TopK` method to build the retriever.

1.2.4 Step 4: Initialize the Inferencer

```
from openicl import PPLInferencer
inferencer = PPLInferencer(model_name='distilgpt2')
```

1.2.5 Step 5: Inference and scoring

```
from openicl import AccEvaluator
# the inferencer requires retriever to collect in-context examples, as well as a
# template to wrap up these examples.
predictions = inferencer.inference(retriever, ice_template=template)
# compute accuracy for the prediction
score = AccEvaluator().score(predictions=predictions, references=data.references)
print(score)
```

1.3 Tutorials List

tutorials can be found at our [github repository](#).

Note: Tutorials are still being updated.

Notebook	Description	Colab Link
Getting Started with OpenICL	Introduction to the main components of OpenICL	
Using Different Language Models with OpenICL	Run different language models with OpenICL	

1.4 DatasetReader

```
class openicl.DatasetReader(dataset: Dataset | DatasetDict | str, input_columns: List[str] | str,
                             output_column: str, name: str | None = None, data_files: str | None = None,
                             input_template: PromptTemplate | None = None, output_template:
                             PromptTemplate | None = None, input_output_template: PromptTemplate |
                             None = None, ds_size: None | int | float = None, split: NamedSplit | None =
                             None, test_split: str | None = 'test')
```

In-context Learning Dataset Reader Class

Generate an DatasetReader instance through ‘dataset’.

dataset

The dataset to be read.

Type

Dataset or DatasetDict

input_columns

A list of column names (a string of column name) in the dataset that represent(s) the input field.

Type

List[str] or str

output_column

A column name in the dataset that represents the prediction field.

Type
str

ds_size

The number of pieces of data to return. When `ds_size` is an integer and greater than or equal to 1, `ds_size` pieces of data are randomly returned. When $0 < ds_size < 1$, $\text{int}(\text{len}(\text{dataset}) * ds_size)$ pieces of data are randomly returned. (used for testing)

Type
int or float, optional

references

The list of references, initialized by `self.dataset[self.test_split][self.output_column]`.

Type
list, optional

input_template

An instance of the *PromptTemplate* class, used to format the input field content during the retrieval process. (in some retrieval methods)

Type
PromptTemplate, optional

output_template

An instance of the *PromptTemplate* class, used to format the output field content during the retrieval process. (in some learnable retrieval methods)

Type
PromptTemplate, optional

input_output_template

An instance of the *PromptTemplate* class, used to format the input-output field content during the retrieval process. (in some retrieval methods)

Type
PromptTemplate, optional

generate_input_field_corpus(*dataset: Dataset | DatasetDict, split: str | None = None*) → List[str]

Generate corpus for input field.

Parameters

- **dataset** (Dataset or DatasetDict) – A `datasets.Dataset` or `datasets.DatasetDict` instance.
- **split** (str, optional) – The split of the dataset to use. If `None`, the entire dataset will be used. Defaults to `None`.

Returns

A list of generated input field prompts.

Return type
List[str]

generate_input_field_prompt(*entry: Dict*) → str

Generate a prompt for the input field based on the provided entry data.

Parameters

entry (Dict) – A piece of data to be used for generating the prompt.

Returns

The generated prompt.

Return type

str

generate_input_output_field_corpus(*dataset: Dataset | DatasetDict, split: str | None = None*) → List[str]

Generate corpus for input-output field.

Parameters

- **dataset** (Dataset or DatasetDict) – A datasets.Dataset or datasets.DatasetDict instance.
- **split** (str, optional) – The split of the dataset to use. If None, the entire dataset will be used. Defaults to None.

Returns

A list of generated input-output field prompts.

Return type

List[str]

generate_input_output_field_prompt(*entry: Dict*) → str

Generate a prompt for the input-output field based on the provided:obj:entry data.

Parameters

entry (Dict) – A piece of data to be used for generating the prompt.

Returns

The generated prompt.

Return type

str

generate_output_field_prompt(*entry: Dict*) → str

Generate a prompt for the output field based on the provided entry data.

Parameters

entry (Dict) – A piece of data to be used for generating the prompt.

Returns

The generated prompt.

Return type

str

generate_output_field_corpus(*dataset: Dataset | DatasetDict, split: str | None = None*) → List[str]

Generate corpus for output field.

Parameters

- **dataset** (Dataset or DatasetDict) – A datasets.Dataset or datasets.DatasetDict instance.
- **split** (str, optional) – The split of the dataset to use. If None, the entire dataset will be used. Defaults to None.

Returns

A list of generated output field prompts.

Return type

List[str]

set_references(column: str, split: str | None = None) → None

Set self.references based on column and optional split.

Parameters

- **column** (str) – A string of column name.
- **split** (str, optional) – A string of dataset split. Defaults to None.

1.5 PromptTemplate

```
class openicl.PromptTemplate(template: Dict | str, column_token_map: Dict, selected_column_name: str |  
                             None = None, selected_column_map: Dict | None = None, ice_token: str |  
                             None = None)
```

In-context Learning Prompt Template Class

This class represents a template that guides the generation of prompts in the retrieval or inference process.

template

A custom template dictionary or string. If a dictionary, the keys of the dictionary represent the values of the output_column, and the values represent the corresponding generated statement. If a string, it represents a string template.

Type

Dict or str

column_token_map

A dictionary mapping column names to specific tokens. The tokens will be replaced by data in the corresponding column (one piece each time) during the retrieval or inference process.

Type

Dict

selected_column_name

Used only with string-type templates. A specific column that needs its value to be mapped.

Type

str, optional

selected_column_map

Used only with string-type templates. Maps the value of the column *selected_column_name*.

Type

Dict, optional

ice_token

A string that represents the specific token mapping from in-context examples. None if you want to use this template only to generate in-context examples, otherwise it can be used to generate the final prompt that is fed into the PLM. The ice_token will be invisible when generating in-context examples.

Type

str, optional

generate_ice_item(*entry*: Dict, *label*: Hashable) → str

Generate in-context example based on the provided **entry** data.

Parameters

- **entry** (Dict) – A piece of data to be used for generating the in-context example.
- **label** (Hashable) – The value of the output field.

Returns

The generated in-context example.

Return type

str

generate_item(*entry*: Dict, *output_field*: Hashable | None = None, *output_field_replace_token*: str | None = "", *ice_field_replace_token*: str | None = "") → str

Generate an item based on the provided **entry** data, as well as optional output field and ice field tokens.

Parameters

- **entry** (Dict) – A piece of data.
- **output_field** (Hashable, optional) – Column name of output field. Defaults to None.
- **output_field_replace_token** (str, optional) – Tokens used to replace output field. Defaults to ' '.
- **ice_field_replace_token** (str, optional) – Tokens used to replace the *ice_token*. Defaults to ' '.

Returns

The generated item.

Return type

str

generate_label_prompt_item(*entry*: Dict, *ice*: str, *label*: Hashable) → str

Generate prompt based on entry data, ice in-context example, and the corresponding label.

Parameters

- **entry** (Dict) – A piece of data containing the input field content.
- **ice** (str) – The generated in-context example.
- **label** (Hashable) – The value of the output field.

Raises

ValueError – If the *ice_token* attribute of the *PromptTemplate* instance is None.

Returns

The generated prompt.

Return type

str

1.6 Inferencer

1.6.1 BaseInferencer

```
class openicl.icl_inferencer.icl_base_inferencer.BaseInferencer(model_name: str | None =  
                                                                'gpt2-xl', tokenizer_name: str |  
                                                                None = None,  
                                                                max_model_token_num: int |  
                                                                None = None, model_config:  
                                                                PretrainedConfig | None = None,  
                                                                batch_size: int | None = 1,  
                                                                accelerator: Accelerator | None  
                                                                = None, output_json_filepath: str  
                                                                | None = './icl_inference_output',  
                                                                output_json_filename: str | None  
                                                                = 'predictions', api_name: str |  
                                                                None = None, model_parallel:  
                                                                bool | None = False, **kwargs)
```

Basic In-context Learning Inferencer Class

Base class of In-context Learning Inferencer, with no inference method.

model

Local PLM (loaded from Hugging Face), which can be initialized by name or a config class.

Type

AutoModelForCausalLM, optional

tokenizer

Tokenizer for [model](#).

Type

AutoTokenizer or GPT2Tokenizer, optional

max_model_token_num

Maximum number of tokenized words allowed by the LM.

Type

int, optional

batch_size

Batch size for the DataLoader.

Type

int, optional

accelerator

An instance of the *Accelerator* class, used for multiprocessing.

Type

Accelerator, optional

output_json_filepath

File path for output *JSON* file.

Type

str, optional

output_json_filename

File name for output *JSON* file.

Type

str, optional

api_name

Name of API service.

Type

str, optional

call_api

If True, an API for LM models will be used, determined by [api_name](#).

Type

bool

inference(*retriever*: [BaseRetriever](#), *ice_template*: [PromptTemplate](#) | None = None, *prompt_template*: [PromptTemplate](#) | None = None, *output_json_filepath*: str | None = None, *output_json_filename*: str | None = None) → List

Perform In-Context Inference given a retriever and optional templates.

Parameters

- **retriever** ([BaseRetriever](#)) – An instance of a Retriever class that will be used to retrieve in-context examples
- **ice_template** ([PromptTemplate](#), optional) – A template for generating the in-context examples prompt. Defaults to None.
- **prompt_template** ([PromptTemplate](#), optional) – A template for generating the final prompt. Defaults to None.
- **output_json_filepath** (str, optional) – The file path to save the results as a *JSON* file. Defaults to None.
- **output_json_filename** (str, optional) – The file name to save the results as a *JSON* file. Defaults to None.

Raises

NotImplementedError – If the function is not implemented in the subclass.

Returns

List: A list of string, each representing the results of one inference.

1.6.2 PPLInferencer

```
class openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer(model_name: str | None = 'gpt2-xl',
                                                             tokenizer_name: str | None = None,
                                                             max_model_token_num: int | None = None,
                                                             model_config: PretrainedConfig | None = None,
                                                             batch_size: int | None = 1,
                                                             accelerator: Accelerator | None = None,
                                                             output_json_filepath: str | None = './icl_inference_output',
                                                             output_json_filename: str | None = 'predictions',
                                                             api_name: str | None = None,
                                                             labels: List | None = None,
                                                             model_parallel: bool | None = False,
                                                             **kwargs)
```

PPL In-context Learning Inferencer Class

Perplexity-based In-context Learning Inferencer.

model

Local PLM (loaded from Hugging Face), which can be initialized by name or a config class.

Type

AutoModelForCausalLM, optional

tokenizer

Tokenizer for [model](#).

Type

AutoTokenizer or GPT2Tokenizer, optional

max_model_token_num

Maximum number of tokenized words allowed by the LM.

Type

int, optional

batch_size

Batch size for the DataLoader.

Type

int, optional

accelerator

An instance of the *Accelerator* class, used for multiprocessing.

Type

Accelerator, optional

output_json_filepath

File path for output *JSON* file.

Type

str, optional

output_json_filename

File name for output *JSON* file.

Type

str, optional

api_name

Name of API service.

Type

str, optional

call_api

If True, an API for LM models will be used, determined by [api_name](#).

Type

bool

labels

A list of labels for all classes.

Type

List, optional

inference(*retriever*: [BaseRetriever](#), *ice_template*: [PromptTemplate](#) | None = None, *prompt_template*: [PromptTemplate](#) | None = None, *output_json_filepath*: str | None = None, *output_json_filename*: str | None = None) → List

Perform In-Context Inference given a retriever and optional templates.

Parameters

- **retriever** ([BaseRetriever](#)) – An instance of a Retriever class that will be used to retrieve in-context examples
- **ice_template** ([PromptTemplate](#), optional) – A template for generating the in-context examples prompt. Defaults to None.
- **prompt_template** ([PromptTemplate](#), optional) – A template for generating the final prompt. Defaults to None.
- **output_json_filepath** (str, optional) – The file path to save the results as a *JSON* file. Defaults to None.
- **output_json_filename** (str, optional) – The file name to save the results as a *JSON* file. Defaults to None.

Raises

NotImplementedError – If the function is not implemented in the subclass.

Returns

List: A list of string, each representing the results of one inference.

1.6.3 GenInferencer

```
class openicl.icl_inferencer.icl_gen_inferencer.GenInferencer(model_name: str | None = 'gpt2-xl',
                                                             tokenizer_name: str | None = None,
                                                             max_model_token_num: int | None
                                                             = None, model_config:
                                                             PretrainedConfig | None = None,
                                                             batch_size: int | None = 1,
                                                             gen_field_replace_token: str | None
                                                             = "",
                                                             generation_kwargs={'do_sample':
                                                             False, 'max_new_tokens': 100},
                                                             accelerator: Accelerator | None =
                                                             None, output_json_filepath: str |
                                                             None = './icl_inference_output',
                                                             output_json_filename: str | None =
                                                             'predictions', api_name: str | None
                                                             = None, model_parallel: bool |
                                                             None = False, **kwargs)
```

Generation In-context Learning Inferencer Class

In-context Learning Inferencer for Directly Generation.

model

Local PLM (loaded from Hugging Face), which can be initialized by name or a config class.

Type

AutoModelForCausalLM, optional

tokenizer

Tokenizer for [model](#).

Type

AutoTokenizer or GPT2Tokenizer, optional

max_model_token_num

Maximum number of tokenized words allowed by the LM.

Type

int, optional

batch_size

Batch size for the DataLoader.

Type

int, optional

accelerator

An instance of the *Accelerator* class, used for multiprocessing.

Type

Accelerator, optional

output_json_filepath

File path for output *JSON* file.

Type

str, optional

output_json_filename

File name for output *JSON* file.

Type
str, optional

api_name
Name of API service.

Type
str, optional

call_api
If True, an API for LM models will be used, determined by [api_name](#).

Type
bool

gen_field_replace_token
Used to replace the generation field token when generating prompts.

Type
str, optional

generation_kwargs
Parameters for the `model.generate()` method.

Type
Dict, optional

inference(*retriever*: [BaseRetriever](#), *ice_template*: [PromptTemplate](#) | *None* = *None*, *prompt_template*: [PromptTemplate](#) | *None* = *None*, *output_json_filepath*: str | *None* = *None*, *output_json_filename*: str | *None* = *None*) → List

Perform In-Context Inference given a retriever and optional templates.

Parameters

- **retriever** ([BaseRetriever](#)) – An instance of a Retriever class that will be used to retrieve in-context examples
- **ice_template** ([PromptTemplate](#), optional) – A template for generating the in-context examples prompt. Defaults to *None*.
- **prompt_template** ([PromptTemplate](#), optional) – A template for generating the final prompt. Defaults to *None*.
- **output_json_filepath** (str, optional) – The file path to save the results as a *JSON* file. Defaults to *None*.
- **output_json_filename** (str, optional) – The file name to save the results as a *JSON* file. Defaults to *None*.

Raises

NotImplementedError – If the function is not implemented in the subclass.

Returns

List: A list of string, each representing the results of one inference.

1.6.4 CoTInferencer

```
class openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer(cot_list: List[str] | None = [],
                                                             model_name: str | None = 'gpt2-xl',
                                                             tokenizer_name: str | None = None,
                                                             max_model_token_num: int | None = None,
                                                             model_config: PretrainedConfig | None = None,
                                                             batch_size: int | None = 1,
                                                             gen_field_replace_token: str | None = "",
                                                             generation_kwargs={'do_sample': False, 'max_new_tokens': 100},
                                                             accelerator: Accelerator | None = None,
                                                             output_json_filepath: str | None = '/icl_inference_output',
                                                             output_json_filename: str | None = 'predictions',
                                                             api_name: str | None = None,
                                                             model_parallel: bool | None = False,
                                                             **kwargs)
```

COT In-context Learning Inferencer Class

Chain-of-Thought In-context Learning Inferencer.

model

Local PLM (loaded from Hugging Face), which can be initialized by name or a config class.

Type

AutoModelForCausalLM, optional

tokenizer

Tokenizer for [model](#).

Type

AutoTokenizer or GPT2Tokenizer, optional

max_model_token_num

Maximum number of tokenized words allowed by the LM.

Type

int, optional

batch_size

Batch size for the DataLoader.

Type

int, optional

accelerator

An instance of the *Accelerator* class, used for multiprocessing.

Type

Accelerator, optional

output_json_filepath

File path for output *JSON* file.

Type

str, optional

output_json_filename

File name for output *JSON* file.

Type

str, optional

api_name

Name of API service.

Type

str, optional

call_api

If True, an API for LM models will be used, determined by *api_name*.

Type

bool

gen_field_replace_token

Used to replace the generation field token when generating prompts.

Type

str, optional

generation_kwargs

Parameters for the `model.generate()` method.

Type

Dict, optional

cot_list

A list of sentences used for multiple-step generations.

Type

list, optional

inference (*retriever*: [BaseRetriever](#), *ice_template*: [PromptTemplate](#) | *None* = *None*, *prompt_template*: [PromptTemplate](#) | *None* = *None*, *output_json_filepath*: str | *None* = *None*, *output_json_filename*: str | *None* = *None*) → List

Perform In-Context Inference given a retriever and optional templates.

Parameters

- **retriever** ([BaseRetriever](#)) – An instance of a Retriever class that will be used to retrieve in-context examples
- **ice_template** ([PromptTemplate](#), optional) – A template for generating the in-context examples prompt. Defaults to *None*.
- **prompt_template** ([PromptTemplate](#), optional) – A template for generating the final prompt. Defaults to *None*.
- **output_json_filepath** (str, optional) – The file path to save the results as a *JSON* file. Defaults to *None*.
- **output_json_filename** (str, optional) – The file name to save the results as a *JSON* file. Defaults to *None*.

Raises

NotImplementedError – If the function is not implemented in the subclass.

Returns

List: A list of string, each representing the results of one inference.

1.7 Retriever

1.7.1 BaseRetriever

```
class openicl.icl_retriever.icl_base_retriever.BaseRetriever(dataset_reader: DatasetReader,  
                                                            ice_separator: str | None = '\n',  
                                                            ice_eos_token: str | None = '\n',  
                                                            prompt_eos_token: str | None = '',  
                                                            ice_num: int | None = 1, index_split:  
                                                            str | None = 'train', test_split: str |  
                                                            None = 'test', accelerator:  
                                                            Accelerator | None = None)
```

Basic In-context Learning Retriever Class

Base class for In-context Learning Retriever, without any retrieval method.

dataset_reader

An instance of the DatasetReader class.

Type

DatasetReader

ice_separator

A string that separates each in-context example.

Type

str, optional

ice_eos_token

A string that is added to the end of in-context examples.

Type

str, optional

prompt_eos_token

A string that is added to the end of the prompt.

Type

str, optional

ice_num

The number of data in the in-context examples.

Type

int, optional

index_split

A string for the index dataset name. The index dataset is used to select data for in-context examples. Defaults to train.

Type

str, optional

test_split

A string for the generation dataset name. The test dataset is used to generate prompts for each data. Defaults to test.

Type

str, optional

index_ds

The index dataset. Used to select data for in-context examples.

Type

Dataset

test_ds

The test dataset. Used to generate prompts for each data.

Type

Dataset

accelerator

An instance of the Accelerator class, used for multiprocessing.

Type

Accelerator, optional

retrieve() → List[List]

Retrieve for each data in generation_ds.

Returns

the index list of in-context example for each data in *test_ds*.

Return type

List[List]

1.7.2 RandomRetriever

```
class openicl.icl_retriever.icl_random_retriever.RandomRetriever(dataset_reader: DatasetReader,
    ice_separator: str | None = '\n',
    ice_eos_token: str | None = '\n',
    prompt_eos_token: str | None = "",
    ice_num: int | None = 1,
    index_split: str | None = 'train',
    test_split: str | None = 'test',
    seed: int | None = 43,
    accelerator: Accelerator | None = None)
```

Random In-context Learning Retriever Class

Class of Random Retriever.

dataset_reader

An instance of the DatasetReader class.

Type

DatasetReader

ice_separator

A string that separates each in-context example.

Type

str, optional

ice_eos_token

A string that is added to the end of in-context examples.

Type

str, optional

prompt_eos_token

A string that is added to the end of the prompt.

Type

str, optional

ice_num

The number of data in the in-context examples.

Type

int, optional

index_split

A string for the index dataset name. The index dataset is used to select data for in-context examples. Defaults to train.

Type

str, optional

test_split

A string for the generation dataset name. The test dataset is used to generate prompts for each data. Defaults to test.

Type

str, optional

index_ds

The index dataset. Used to select data for in-context examples.

Type

Dataset

test_ds

The test dataset. Used to generate prompts for each data.

Type

Dataset

accelerator

An instance of the `Accelerator` class, used for multiprocessing.

Type

Accelerator, optional

seed

Seed for the random number generator.

Type

int, optional

retrieve()

Retrieve for each data in `generation_ds`.

Returns

the index list of in-context example for each data in `test_ds`.

Return type

List[List]

1.7.3 BM25Retriever

```
class openicl.icl_retriever.icl_bm25_retriever.BM25Retriever(dataset_reader: DatasetReader,
                                                            ice_separator: str | None = '\n',
                                                            ice_eos_token: str | None = '\n',
                                                            prompt_eos_token: str | None = "",
                                                            ice_num: int | None = 1, index_split:
                                                            str | None = 'train', test_split: str |
                                                            None = 'test', accelerator:
                                                            Accelerator | None = None)
```

BM25 In-context Learning Retriever Class

Class of BM25 Retriever.

dataset_reader

An instance of the DatasetReader class.

Type

DatasetReader

ice_separator

A string that separates each in-context example.

Type

str, optional

ice_eos_token

A string that is added to the end of in-context examples.

Type

str, optional

prompt_eos_token

A string that is added to the end of the prompt.

Type

str, optional

ice_num

The number of data in the in-context examples.

Type

int, optional

index_split

A string for the index dataset name. The index dataset is used to select data for in-context examples. Defaults to `train`.

Type

str, optional

test_split

A string for the generation dataset name. The test dataset is used to generate prompts for each data. Defaults to test.

Type

str, optional

index_ds

The index dataset. Used to select data for in-context examples.

Type

Dataset

test_ds

The test dataset. Used to generate prompts for each data.

Type

Dataset

accelerator

An instance of the Accelerator class, used for multiprocessing.

Type

Accelerator, optional

index_corpus

A corpus created from the input field data of *index_ds*.

Type

List[str]

test_corpus

A corpus created from the input field data of *test_ds*.

Type

List[str]

bm25

An instance of BM250kapi class, initialized using *index_ds*.

Type

BM250kapi

retrieve() → List[List]

Retrieve for each data in generation_ds.

Returns

the index list of in-context example for each data in *test_ds*.

Return type

List[List]

1.7.4 TopkRetriever

```
class openicl.icl_retriever.icl_topk_retriever.TopkRetriever(dataset_reader: DatasetReader,
                                                           ice_separator: str | None = '\n',
                                                           ice_eos_token: str | None = '\n',
                                                           prompt_eos_token: str | None = "",
                                                           sen-
                                                           tence_transformers_model_name:
                                                           str | None = 'all-mpnet-base-v2',
                                                           ice_num: int | None = 1, index_split:
                                                           str | None = 'train', test_split: str |
                                                           None = 'test', tokenizer_name: str |
                                                           None = 'gpt2-xl', batch_size: int |
                                                           None = 1, accelerator: Accelerator |
                                                           None = None)
```

Topk In-context Learning Retriever Class

Class of Topk Retriever.

dataset_reader

An instance of the DatasetReader class.

Type

DatasetReader

ice_separator

A string that separates each in-context example.

Type

str, optional

ice_eos_token

A string that is added to the end of in-context examples.

Type

str, optional

prompt_eos_token

A string that is added to the end of the prompt.

Type

str, optional

ice_num

The number of data in the in-context examples.

Type

int, optional

index_split

A string for the index dataset name. The index dataset is used to select data for in-context examples. Defaults to train.

Type

str, optional

test_split

A string for the generation dataset name. The test dataset is used to generate prompts for each data. Defaults to test.

Type

str, optional

index_ds

The index dataset. Used to select data for in-context examples.

Type

Dataset

test_ds

The test dataset. Used to generate prompts for each data.

Type

Dataset

acceleratorAn instance of the `Accelerator` class, used for multiprocessing.**Type**

Accelerator, optional

batch_sizeBatch size for the `DataLoader`.**Type**

int, optional

modelAn instance of `SentenceTransformer` class, used to calculate embeddings.**Type**

SentenceTransformer

tokenizerTokenizer for *model*.**Type**

AutoTokenizer

index

Index generated with FAISS.

Type

IndexIDMap

retrieve()Retrieve for each data in `generation_ds`.**Returns**the index list of in-context example for each data in *test_ds*.**Return type***List[List]*

1.7.5 VotekRetriever

```
class openicl.icl_retriever.icl_votek_retriever.VotekRetriever(dataset_reader: DatasetReader,
                                                             ice_separator: str | None = '\n',
                                                             ice_eos_token: str | None = '\n',
                                                             prompt_eos_token: str | None = "",
                                                             sen-
                                                             tence_transformers_model_name:
                                                             str | None = 'all-mpnet-base-v2',
                                                             ice_num: int | None = 1,
                                                             index_split: str | None = 'train',
                                                             test_split: str | None = 'test',
                                                             tokenizer_name: str | None =
                                                             'gpt2-xl', batch_size: int | None =
                                                             1, votek_k: int | None = 3,
                                                             accelerator: Accelerator | None =
                                                             None)
```

Vote-k In-context Learning Retriever Class

Class of Vote-k Retriever.

dataset_reader

An instance of the DatasetReader class.

Type

DatasetReader

ice_separator

A string that separates each in-context example.

Type

str, optional

ice_eos_token

A string that is added to the end of in-context examples.

Type

str, optional

prompt_eos_token

A string that is added to the end of the prompt.

Type

str, optional

ice_num

The number of data in the in-context examples.

Type

int, optional

index_split

A string for the index dataset name. The index dataset is used to select data for in-context examples. Defaults to train.

Type

str, optional

test_split

A string for the generation dataset name. The test dataset is used to generate prompts for each data. Defaults to `test`.

Type

str, optional

index_ds

The index dataset. Used to select data for in-context examples.

Type

Dataset

test_ds

The test dataset. Used to generate prompts for each data.

Type

Dataset

accelerator

An instance of the `Accelerator` class, used for multiprocessing.

Type

Accelerator, optional

batch_size

Batch size for the `DataLoader`.

Type

int, optional

model

An instance of `SentenceTransformer` class, used to calculate embeddings.

Type

SentenceTransformer

tokenizer

Tokenizer for `:obj:model`.

Type

AutoTokenizer

index

Index generated with FAISS.

Type

IndexIDMap

votek_k

k value of Voke-k Selective Annotation Algorithm. Defaults to 3.

Type

int, optional

retrieve()

Retrieve for each data in `generation_ds`.

Returns

the index list of in-context example for each data in `test_ds`.

Return type
List[List]

1.7.6 DPPRetriever

```
class openicl.icl_retriever.icl_dpp_retriever.DPPRetriever(dataset_reader: DatasetReader,
                                                         ice_separator: str | None = '\n',
                                                         ice_eos_token: str | None = '\n',
                                                         prompt_eos_token: str | None = "",
                                                         sentence_transformers_model_name:
                                                         str | None = 'all-mpnet-base-v2',
                                                         ice_num: int | None = 1,
                                                         candidate_num: int | None = 1,
                                                         index_split: str | None = 'train',
                                                         test_split: str | None = 'test',
                                                         tokenizer_name: str | None = 'gpt2-xl',
                                                         batch_size: int | None = 1, accelerator:
                                                         Accelerator | None = None, seed: int |
                                                         None = 1, scale_factor: float | None =
                                                         0.1)
```

DPP In-context Learning Retriever Class

Class of DPP Retriever. Two-stage DPP is used, where first stage is to get results of TopK to reduce candidate sets checkout <https://arxiv.org/abs/2302.05698> for details.

dataset_reader

An instance of the DatasetReader class.

Type
DatasetReader

ice_separator

A string that separates each in-context example.

Type
str, optional

ice_eos_token

A string that is added to the end of in-context examples.

Type
str, optional

prompt_eos_token

A string that is added to the end of the prompt.

Type
str, optional

ice_num

The number of data in the in-context examples.

Type
int, optional

index_split

A string for the index dataset name. The index dataset is used to select data for in-context examples. Defaults to train.

Type

str, optional

test_split

A string for the generation dataset name. The test dataset is used to generate prompts for each data. Defaults to test.

Type

str, optional

index_ds

The index dataset. Used to select data for in-context examples.

Type

Dataset

test_ds

The test dataset. Used to generate prompts for each data.

Type

Dataset

accelerator

An instance of the Accelerator class, used for multiprocessing.

Type

Accelerator, optional

batch_size

Batch size for the DataLoader.

Type

int, optional

model

An instance of SentenceTransformer class, used to calculate embeddings.

Type

SentenceTransformer

tokenizer

Tokenizer for [model](#).

Type

AutoTokenizer

index

Index generated with FAISS.

Type

IndexIDMap

seed

Seed for the random number generator. (random_state in sample_exact_k_dpp method)

Type

int, optional

scale_factor

A factor when gets the kernel.

Type

float, optional

retrieve()

Retrieve for each data in `generation_ds`.

Returns

the index list of in-context example for each data in `test_ds`.

Return type

`List[List]`

1.7.7 MDLRetriever

```
class openicl.icl_retriever.icl_mdل_retriever.MDLRetriever(dataset_reader: DatasetReader,
                                                         ice_separator: str | None = '\n',
                                                         ice_eos_token: str | None = '\n',
                                                         prompt_eos_token: str | None = '',
                                                         sentence_transformers_model_name:
                                                         str | None = 'all-mpnet-base-v2',
                                                         ice_num: int | None = 1,
                                                         candidate_num: int | None = 1,
                                                         index_split: str | None = 'train',
                                                         test_split: str | None = 'test',
                                                         tokenizer_name: str | None = 'gpt2-xl',
                                                         ce_model_name: str | None = 'gpt2-xl',
                                                         batch_size: int | None = 1, select_time:
                                                         int | None = 5, accelerator: Accelerator
                                                         | None = None, ice_template:
                                                         PromptTemplate | None = None,
                                                         prompt_template: PromptTemplate |
                                                         None = None, labels: List | None =
                                                         None, seed: int | None = 1)
```

MDL In-context Learning Retriever Class

Class of MDL Retriever.

dataset_reader

An instance of the `DatasetReader` class.

Type

`DatasetReader`

ice_separator

A string that separates each in-context example.

Type

str, optional

ice_eos_token

A string that is added to the end of in-context examples.

Type

str, optional

prompt_eos_token

A string that is added to the end of the prompt.

Type

str, optional

ice_num

The number of data in the in-context examples.

Type

int, optional

candidate_num

The number of data selected in TopK stage.

Type

int, optional

index_split

A string for the index dataset name. The index dataset is used to select data for in-context examples. Defaults to train.

Type

str, optional

test_split

A string for the generation dataset name. The test dataset is used to generate prompts for each data. Defaults to test.

Type

str, optional

index_ds

The index dataset. Used to select data for in-context examples.

Type

Dataset

test_ds

The test dataset. Used to generate prompts for each data.

Type

Dataset

accelerator

An instance of the Accelerator class, used for multiprocessing.

Type

Accelerator, optional

batch_size

Batch size for the DataLoader.

Type

int, optional

model

An instance of `SentenceTransformer` class, used to calculate embeddings.

Type

`SentenceTransformer`

tokenizer

Tokenizer for *model*.

Type

`AutoTokenizer`

index

Index generated with FAISS.

Type

`IndexIDMap`

select_time

Number of random selections in the MDL stage.

Type

`int`, optional

labels

A list of labels for all classes used to generate prompts when calculating MDL.

Type

`List`, optional

seed

Seed for the random number generator.

Type

`int`, optional

retrieve()

Retrieve for each data in `generation_ds`.

Returns

the index list of in-context example for each data in *test_ds*.

Return type

`List[List]`

1.7.8 ZeroRetriever

```
class openicl.icl_retriever.icl_zero_retriever.ZeroRetriever(dataset_reader: DatasetReader,
                                                             ice_eos_token: str | None = "",
                                                             prompt_eos_token: str | None = "",
                                                             index_split: str | None = 'train',
                                                             test_split: str | None = 'test',
                                                             accelerator: Accelerator | None =
                                                             None)
```

Zero In-context Learning Retriever Class

Retriever for Zero-shot.

dataset_reader

An instance of the `DatasetReader` class.

Type

`DatasetReader`

ice_eos_token

A string that is added to the end of in-context examples.

Type

`str`, optional

prompt_eos_token

A string that is added to the end of the prompt.

Type

`str`, optional

index_split

A string for the index dataset name. The index dataset is used to select data for in-context examples. Defaults to `train`.

Type

`str`, optional

test_split

A string for the generation dataset name. The test dataset is used to generate prompts for each data. Defaults to `test`.

Type

`str`, optional

index_ds

The index dataset. Used to select data for in-context examples.

Type

`Dataset`

test_ds

The test dataset. Used to generate prompts for each data.

Type

`Dataset`

accelerator

An instance of the `Accelerator` class, used for multiprocessing.

Type

`Accelerator`, optional

retrieve() → `List[List]`

Retrieve for each data in `generation_ds`.

Returns

the index list of in-context example for each data in `test_ds`.

Return type

`List[List]`

INDICES AND TABLES

- `genindex`
- `search`

INDEX

A

`accelerator` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` attribute), 10
`accelerator` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 16
`accelerator` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` attribute), 14
`accelerator` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 12
`accelerator` (`openicl.icl_retriever.icl_base_retriever.BaseRetriever` attribute), 19
`accelerator` (`openicl.icl_retriever.icl_bm25_retriever.BM25Retriever` attribute), 22
`accelerator` (`openicl.icl_retriever.icl_dpp_retriever.DPPRetriever` attribute), 28
`accelerator` (`openicl.icl_retriever.icl_md1_retriever.MDLRetriever` attribute), 30
`accelerator` (`openicl.icl_retriever.icl_random_retriever.RandomRetriever` attribute), 20
`accelerator` (`openicl.icl_retriever.icl_topk_retriever.TopkRetriever` attribute), 24
`accelerator` (`openicl.icl_retriever.icl_votek_retriever.VotekRetriever` attribute), 26
`accelerator` (`openicl.icl_retriever.icl_zero_retriever.ZeroRetriever` attribute), 32
`api_name` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` attribute), 11
`api_name` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 17
`api_name` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` attribute), 15
`api_name` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 13
`api_name` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 12
`batch_size` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` attribute), 10
`batch_size` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 16
`batch_size` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` attribute), 14
`batch_size` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 12
`batch_size` (`openicl.icl_retriever.icl_dpp_retriever.DPPRetriever` attribute), 28
`batch_size` (`openicl.icl_retriever.icl_md1_retriever.MDLRetriever` attribute), 30
`batch_size` (`openicl.icl_retriever.icl_topk_retriever.TopkRetriever` attribute), 24
`batch_size` (`openicl.icl_retriever.icl_votek_retriever.VotekRetriever` attribute), 26
`bm25` (`openicl.icl_retriever.icl_bm25_retriever.BM25Retriever` attribute), 22
`BM25Retriever` (class in `openicl.icl_retriever.icl_bm25_retriever`), 21
`call_api` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` attribute), 11
`call_api` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 17
`call_api` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` attribute), 15
`call_api` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 13
`candidate_num` (`openicl.icl_retriever.icl_md1_retriever.MDLRetriever` attribute), 30
`column_token_map` (`openicl.PromptTemplate` attribute), 8
`cot_list` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 17
`CoTInferencer` (class in `openicl.icl_inferencer.icl_cot_inferencer`), 16

B

`BaseInferencer` (class in `openicl.icl_inferencer.icl_base_inferencer`), 10
`BaseRetriever` (class in `openicl.icl_retriever.icl_base_retriever`), 18

D

`dataset` (`openicl.DatasetReader` attribute), 5

dataset_reader (*openicl.icl_retriever.icl_base_retriever.BaseRetriever*
 attribute), 18
 dataset_reader (*openicl.icl_retriever.icl_bm25_retriever.BM25Retriever*
 attribute), 21
 dataset_reader (*openicl.icl_retriever.icl_dpp_retriever.DPPRetriever*
 attribute), 27
 dataset_reader (*openicl.icl_retriever.icl_md1_retriever.MDLRetriever*
 attribute), 29
 dataset_reader (*openicl.icl_retriever.icl_random_retriever.RandomRetriever*
 attribute), 19
 dataset_reader (*openicl.icl_retriever.icl_topk_retriever.TopkRetriever*
 attribute), 23
 dataset_reader (*openicl.icl_retriever.icl_votek_retriever.VotekRetriever*
 attribute), 25
 dataset_reader (*openicl.icl_retriever.icl_zero_retriever.ZeroRetriever*
 attribute), 31
 DatasetReader (*class in openicl*), 5
 DPPRetriever (*class in openicl.icl_retriever.icl_dpp_retriever*), 27
 ds_size (*openicl.DatasetReader attribute*), 6
G
 gen_field_replace_token
 (*openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer*
 attribute), 17
 gen_field_replace_token
 (*openicl.icl_inferencer.icl_gen_inferencer.GenInferencer*
 attribute), 15
 generate_ice_item() (*openicl.PromptTemplate*
 method), 8
 generate_input_field_corpus()
 (*openicl.DatasetReader method*), 6
 generate_input_field_prompt()
 (*openicl.DatasetReader method*), 6
 generate_input_output_field_corpus()
 (*openicl.DatasetReader method*), 7
 generate_input_output_field_prompt()
 (*openicl.DatasetReader method*), 7
 generate_item() (*openicl.PromptTemplate method*), 9
 generate_label_prompt_item()
 (*openicl.PromptTemplate method*), 9
 generate_ouput_field_prompt()
 (*openicl.DatasetReader method*), 7
 generate_output_field_corpus()
 (*openicl.DatasetReader method*), 7
 generation_kwargs (*openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer*
 attribute), 17
 generation_kwargs (*openicl.icl_inferencer.icl_gen_inferencer.GenInferencer*
 attribute), 15
 GenInferencer (*class in openicl.icl_inferencer.icl_gen_inferencer*),
 13
 ice_eos_token (*openicl.icl_retriever.icl_base_retriever.BaseRetriever*
 attribute), 18
 ice_eos_token (*openicl.icl_retriever.icl_bm25_retriever.BM25Retriever*
 attribute), 21
 ice_eos_token (*openicl.icl_retriever.icl_dpp_retriever.DPPRetriever*
 attribute), 27
 ice_eos_token (*openicl.icl_retriever.icl_md1_retriever.MDLRetriever*
 attribute), 29
 ice_eos_token (*openicl.icl_retriever.icl_random_retriever.RandomRetriever*
 attribute), 19
 ice_eos_token (*openicl.icl_retriever.icl_topk_retriever.TopkRetriever*
 attribute), 20
 ice_eos_token (*openicl.icl_retriever.icl_votek_retriever.VotekRetriever*
 attribute), 23
 ice_eos_token (*openicl.icl_retriever.icl_zero_retriever.ZeroRetriever*
 attribute), 32
 ice_num (*openicl.icl_retriever.icl_base_retriever.BaseRetriever*
 attribute), 18
 ice_num (*openicl.icl_retriever.icl_bm25_retriever.BM25Retriever*
 attribute), 21
 ice_num (*openicl.icl_retriever.icl_dpp_retriever.DPPRetriever*
 attribute), 27
 ice_num (*openicl.icl_retriever.icl_md1_retriever.MDLRetriever*
 attribute), 30
 ice_num (*openicl.icl_retriever.icl_random_retriever.RandomRetriever*
 attribute), 20
 ice_num (*openicl.icl_retriever.icl_topk_retriever.TopkRetriever*
 attribute), 23
 ice_num (*openicl.icl_retriever.icl_votek_retriever.VotekRetriever*
 attribute), 25
 ice_separator (*openicl.icl_retriever.icl_base_retriever.BaseRetriever*
 attribute), 18
 ice_separator (*openicl.icl_retriever.icl_bm25_retriever.BM25Retriever*
 attribute), 21
 ice_separator (*openicl.icl_retriever.icl_dpp_retriever.DPPRetriever*
 attribute), 27
 ice_separator (*openicl.icl_retriever.icl_md1_retriever.MDLRetriever*
 attribute), 29
 ice_separator (*openicl.icl_retriever.icl_random_retriever.RandomRetriever*
 attribute), 19
 ice_separator (*openicl.icl_retriever.icl_topk_retriever.TopkRetriever*
 attribute), 23
 ice_separator (*openicl.icl_retriever.icl_votek_retriever.VotekRetriever*
 attribute), 25
 ice_token (*openicl.PromptTemplate attribute*), 8
 index (*openicl.icl_retriever.icl_dpp_retriever.DPPRetriever*
 attribute), 28
 index (*openicl.icl_retriever.icl_md1_retriever.MDLRetriever*
 attribute), 31
 index (*openicl.icl_retriever.icl_topk_retriever.TopkRetriever*
 attribute), 24
 index (*openicl.icl_retriever.icl_votek_retriever.VotekRetriever*
 attribute), 26

`index_corpus` (`openicl.icl_retriever.icl_bm25_retriever.BM25Retriever` attribute), 22
`index_ds` (`openicl.icl_retriever.icl_base_retriever.BaseRetriever` attribute), 19
`index_ds` (`openicl.icl_retriever.icl_bm25_retriever.BM25Retriever` attribute), 22
`index_ds` (`openicl.icl_retriever.icl_dpp_retriever.DPPRetriever` attribute), 28
`index_ds` (`openicl.icl_retriever.icl_md1_retriever.MDLRetriever` attribute), 30
`index_ds` (`openicl.icl_retriever.icl_random_retriever.RandomRetriever` attribute), 20
`index_ds` (`openicl.icl_retriever.icl_topk_retriever.TopkRetriever` attribute), 24
`index_ds` (`openicl.icl_retriever.icl_votek_retriever.VotekRetriever` attribute), 26
`index_ds` (`openicl.icl_retriever.icl_zero_retriever.ZeroRetriever` attribute), 32
`index_split` (`openicl.icl_retriever.icl_base_retriever.BaseRetriever` attribute), 18
`index_split` (`openicl.icl_retriever.icl_bm25_retriever.BM25Retriever` attribute), 21
`index_split` (`openicl.icl_retriever.icl_dpp_retriever.DPPRetriever` attribute), 27
`index_split` (`openicl.icl_retriever.icl_md1_retriever.MDLRetriever` attribute), 30
`index_split` (`openicl.icl_retriever.icl_random_retriever.RandomRetriever` attribute), 20
`index_split` (`openicl.icl_retriever.icl_topk_retriever.TopkRetriever` attribute), 23
`index_split` (`openicl.icl_retriever.icl_votek_retriever.VotekRetriever` attribute), 25
`index_split` (`openicl.icl_retriever.icl_zero_retriever.ZeroRetriever` attribute), 32
`inference()` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` method), 11
`inference()` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` method), 17
`inference()` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` method), 15
`inference()` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` method), 13
`input_columns` (`openicl.DatasetReader` attribute), 5
`input_output_template` (`openicl.DatasetReader` attribute), 6
`input_template` (`openicl.DatasetReader` attribute), 6
L
`labels` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 13
`labels` (`openicl.icl_retriever.icl_md1_retriever.MDLRetriever` attribute), 31

`max_model_token_num` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` attribute), 10
`max_model_token_num` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 16
`max_model_token_num` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` attribute), 14
`max_model_token_num` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 12
`MDLRetriever` (class in `openicl.icl_retriever.icl_md1_retriever`), 29
`model` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` attribute), 10
`model` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 16
`model` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` attribute), 14
`model` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 12
`model` (`openicl.icl_retriever.icl_dpp_retriever.DPPRetriever` attribute), 28
`model` (`openicl.icl_retriever.icl_md1_retriever.MDLRetriever` attribute), 30
`model` (`openicl.icl_retriever.icl_topk_retriever.TopkRetriever` attribute), 24
`model` (`openicl.icl_retriever.icl_votek_retriever.VotekRetriever` attribute), 26
`output_column` (`openicl.DatasetReader` attribute), 5
`output_json_filename` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` attribute), 10
`output_json_filename` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 17
`output_json_filename` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` attribute), 14
`output_json_filename` (`openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer` attribute), 12
`output_json_filepath` (`openicl.icl_inferencer.icl_base_inferencer.BaseInferencer` attribute), 10
`output_json_filepath` (`openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer` attribute), 16
`output_json_filepath` (`openicl.icl_inferencer.icl_gen_inferencer.GenInferencer` attribute), 14

attribute), 14
 output_json_filepath (openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer attribute), 12
 output_template (openicl.DatasetReader attribute), 6

P

PPLInferencer (class in openicl.icl_inferencer.icl_ppl_inferencer), 11
 prompt_eos_token (openicl.icl_retriever.icl_base_retriever.BaseRetriever attribute), 18
 prompt_eos_token (openicl.icl_retriever.icl_bm25_retriever.BM25Retriever attribute), 21
 prompt_eos_token (openicl.icl_retriever.icl_dpp_retriever.DPPRetriever attribute), 27
 prompt_eos_token (openicl.icl_retriever.icl_mdل_retriever.MDLRetriever attribute), 30
 prompt_eos_token (openicl.icl_retriever.icl_random_retriever.RandomRetriever attribute), 20
 prompt_eos_token (openicl.icl_retriever.icl_topk_retriever.TopkRetriever attribute), 23
 prompt_eos_token (openicl.icl_retriever.icl_votek_retriever.VotekRetriever attribute), 25
 prompt_eos_token (openicl.icl_retriever.icl_zero_retriever.ZeroRetriever attribute), 32
 PromptTemplate (class in openicl), 8

R

RandomRetriever (class in openicl.icl_retriever.icl_random_retriever), 19
 references (openicl.DatasetReader attribute), 6
 retrieve() (openicl.icl_retriever.icl_base_retriever.BaseRetriever method), 19
 retrieve() (openicl.icl_retriever.icl_bm25_retriever.BM25Retriever method), 22
 retrieve() (openicl.icl_retriever.icl_dpp_retriever.DPPRetriever method), 29
 retrieve() (openicl.icl_retriever.icl_mdل_retriever.MDLRetriever method), 31
 retrieve() (openicl.icl_retriever.icl_random_retriever.RandomRetriever method), 20
 retrieve() (openicl.icl_retriever.icl_topk_retriever.TopkRetriever method), 24
 retrieve() (openicl.icl_retriever.icl_votek_retriever.VotekRetriever method), 26
 retrieve() (openicl.icl_retriever.icl_zero_retriever.ZeroRetriever method), 32

S

scale_factor (openicl.icl_retriever.icl_dpp_retriever.DPPRetriever attribute), 28
 seed (openicl.icl_retriever.icl_dpp_retriever.DPPRetriever attribute), 28
 seed (openicl.icl_retriever.icl_mdل_retriever.MDLRetriever attribute), 31
 seed (openicl.icl_retriever.icl_random_retriever.RandomRetriever attribute), 20
 select_time (openicl.icl_retriever.icl_mdل_retriever.MDLRetriever attribute), 31
 selected_column_map (openicl.PromptTemplate attribute), 8
 selected_column_name (openicl.PromptTemplate attribute), 8
 set_references() (openicl.DatasetReader method), 8

T

template (openicl.PromptTemplate attribute), 8
 test_ds (openicl.icl_retriever.icl_bm25_retriever.BM25Retriever attribute), 22
 test_ds (openicl.icl_retriever.icl_base_retriever.BaseRetriever attribute), 19
 test_ds (openicl.icl_retriever.icl_bm25_retriever.BM25Retriever attribute), 22
 test_ds (openicl.icl_retriever.icl_dpp_retriever.DPPRetriever attribute), 28
 test_ds (openicl.icl_retriever.icl_mdل_retriever.MDLRetriever attribute), 30
 test_ds (openicl.icl_retriever.icl_random_retriever.RandomRetriever attribute), 20
 test_ds (openicl.icl_retriever.icl_topk_retriever.TopkRetriever attribute), 24
 test_ds (openicl.icl_retriever.icl_votek_retriever.VotekRetriever attribute), 26
 test_ds (openicl.icl_retriever.icl_zero_retriever.ZeroRetriever attribute), 32
 test_split (openicl.icl_retriever.icl_base_retriever.BaseRetriever attribute), 18
 test_split (openicl.icl_retriever.icl_bm25_retriever.BM25Retriever attribute), 22
 test_split (openicl.icl_retriever.icl_dpp_retriever.DPPRetriever attribute), 28
 test_split (openicl.icl_retriever.icl_mdل_retriever.MDLRetriever attribute), 30
 test_split (openicl.icl_retriever.icl_random_retriever.RandomRetriever attribute), 20
 test_split (openicl.icl_retriever.icl_topk_retriever.TopkRetriever attribute), 23
 test_split (openicl.icl_retriever.icl_votek_retriever.VotekRetriever attribute), 25
 test_split (openicl.icl_retriever.icl_zero_retriever.ZeroRetriever attribute), 32
 tokenizer (openicl.icl_inferencer.icl_base_inferencer.BaseInferencer attribute), 10
 tokenizer (openicl.icl_inferencer.icl_cot_inferencer.CoTInferencer attribute), 16

[tokenizer](#) (*openicl.icl_inferencer.icl_gen_inferencer.GenInferencer*
attribute), [14](#)
[tokenizer](#) (*openicl.icl_inferencer.icl_ppl_inferencer.PPLInferencer*
attribute), [12](#)
[tokenizer](#) (*openicl.icl_retriever.icl_dpp_retriever.DPPRetriever*
attribute), [28](#)
[tokenizer](#) (*openicl.icl_retriever.icl_mdل_retriever.MDLRetriever*
attribute), [31](#)
[tokenizer](#) (*openicl.icl_retriever.icl_topk_retriever.TopkRetriever*
attribute), [24](#)
[tokenizer](#) (*openicl.icl_retriever.icl_votek_retriever.VotekRetriever*
attribute), [26](#)
[TopkRetriever](#) (class in
openicl.icl_retriever.icl_topk_retriever), [23](#)

V

[votek_k](#) (*openicl.icl_retriever.icl_votek_retriever.VotekRetriever*
attribute), [26](#)
[VotekRetriever](#) (class in
openicl.icl_retriever.icl_votek_retriever),
[25](#)

Z

[ZeroRetriever](#) (class in
openicl.icl_retriever.icl_zero_retriever), [31](#)