
GNES Documentation

Release 0.0.46

Han Xiao, Feng Wang, Jianfeng Yan, Jie Fu

Oct 24, 2019

1	Highlights	3
1.1	All Microservices in GNES	3
1.1.1	Named Arguments	3
1.1.2	GNES sub-commands	3
1.1.3	Sub-commands:	4
1.2	gnes package	19
1.2.1	Subpackages	19
1.2.2	Submodules	75
1.2.3	Module contents	76
1.3	Troubleshooting	76
1.3.1	Check if docker swarm/stack runs successfully	76
1.3.2	Investigate the reason of a failed service	77
1.3.3	Delete a failed service	77
1.3.4	Locate internal errors by looking at logs	78
1.4	Protobuf Implementation	78
1.4.1	Generating <code>gnes_pb2.py</code> and <code>gnes_pb2_grpc.py</code>	78
1.5	Environment Variables	79
1.5.1	GNES_PROFILING	79
1.5.2	GNES_PROFILING_MEM	79
1.5.3	GNES_WARN_UNNAMED_COMPONENT	79
1.5.4	GNES_VCS_VERSION	80
1.5.5	GNES_CONTROL_PORT	80
1.5.6	GNES_CONTRIB_MODULE	80
1.5.7	GNES_IPC SOCK_TMP	80
1.6	Using GNES with Docker Swarm	80
1.6.1	Build your first GNES app on local machine	80
1.6.2	Scale your GNES app to the cloud	82
1.6.3	Customize GNES to your need	83
1.7	Contributing to GNES	83
1.7.1	Making Your First Commit	83
1.7.2	Table of Content	84
1.7.3	Commit Message Naming	84
1.7.4	Merging Process	85
1.7.5	Release Process	85
1.7.6	Testing Locally	86
1.7.7	Interesting Points	86

1.8	Release Note (v0.0.46)	86
1.8.1	New Features	86
1.8.2	Bug fixes	87
1.8.3	Code Refactoring	87
1.8.4	Documentation	87
1.8.5	Other Improvements	87
1.9	Breaking Changes (v0.0.45 -> v0.0.46)	87
1.10	Release Note (v0.0.45)	88
1.10.1	New Features	88
1.10.2	Bug fixes	88
1.10.3	Code Refactoring	88
1.10.4	Other Improvements	88
1.11	Release Note (v0.0.44)	88
1.11.1	New Features	89
1.11.2	Bug fixes	89
1.11.3	Documentation	89
1.11.4	Other Improvements	89
1.12	Release Note (v0.0.43)	89
1.12.1	New Features	90
1.12.2	Bug fixes	90
1.12.3	Code Refactoring	90
1.12.4	Documentation	90
1.12.5	Other Improvements	91
1.13	Release Note (v0.0.42)	91
1.13.1	New Features	91
1.13.2	Bug fixes	91
1.13.3	Documentation	92
1.13.4	Other Improvements	92
1.14	Release Note (v0.0.41)	92
1.14.1	New Features	92
1.14.2	Bug fixes	92
1.14.3	Code Refactoring	93
1.14.4	Other Improvements	93
1.15	Release Note (v0.0.40)	93
1.15.1	New Features	93
1.15.2	Bug fixes	93
1.15.3	Code Refactoring	93
1.15.4	Documentation	94
1.15.5	Unit Test and CICD	94
1.15.6	Other Improvements	94
1.16	Release Note (v0.0.39)	94
1.16.1	New Features	94
1.16.2	Bug fixes	94
1.16.3	Code Refactoring	95
1.16.4	Documentation	95
1.16.5	Other Improvements	95
1.17	Release Note (v0.0.38)	95
1.17.1	Other Improvements	95
1.18	Release Note (v0.0.37)	95
1.18.1	New Features	96
1.18.2	Bug fixes	96
1.18.3	Code Refactoring	97
1.18.4	Other Improvements	97
1.19	Release Note (v0.0.36)	97

1.19.1	New Features	97
1.19.2	Bug fixes	97
1.19.3	Code Refactoring	98
1.19.4	Unit Test and CICD	98
1.19.5	Other Improvements	98
1.20	Release Note (v0.0.35)	99
1.20.1	New Features	99
1.20.2	Bug fixes	99
1.20.3	Code Refactoring	99
1.20.4	Other Improvements	99
1.21	Release Note (v0.0.34)	99
1.21.1	Other Improvements	99
1.22	Release Note (v0.0.34)	99
1.22.1	Other Improvements	100
1.23	Release Note (v0.0.33)	100
1.23.1	New Features	100
1.23.2	Bug fixes	100
1.23.3	Code Refactoring	100
1.23.4	Documentation	100
1.23.5	Other Improvements	101
1.24	Release Note (v0.0.32)	101
1.24.1	New Features	101
1.24.2	Bug fixes	101
1.24.3	Code Refactoring	101
1.24.4	Unit Test and CICD	101
1.24.5	Other Improvements	101
1.25	Release Note (v0.0.31)	101
1.25.1	New Features	101
1.25.2	Other Improvements	102
1.26	Release Note (v0.0.30)	102
1.26.1	New Features	102
1.26.2	Bug fixes	102
1.26.3	Other Improvements	102
1.27	Release Note (v0.0.29)	102
1.27.1	Bug fixes	102
1.27.2	Code Refactoring	102
1.27.3	Documentation	102
1.27.4	Unit Test and CICD	103
1.27.5	Other Improvements	103
1.28	Release Note (v0.0.28)	103
1.28.1	New Features	103
1.28.2	Bug fixes	103
1.28.3	Documentation	103
1.28.4	Unit Test and CICD	104
1.28.5	Other Improvements	104
1.29	Release Note (v0.0.27)	104
1.29.1	New Features	104
1.29.2	Bug fixes	104
1.29.3	Code Refactoring	105
1.29.4	Unit Test and CICD	105
1.29.5	Other Improvements	105
1.30	Release Note (v0.0.26)	106
1.30.1	New Features	106
1.30.2	Bug fixes	106

1.30.3	Code Refactoring	106
1.30.4	Documentation	106
1.30.5	Other Improvements	107
1.31	Release Note (v0.0.25)	107
1.31.1	New Features	107
1.31.2	Bug fixes	107
1.31.3	Code Refactoring	108
1.31.4	Documentation	108
1.31.5	Unit Test and CICD	108
1.31.6	Other Improvements	109
1.32	Release Note (v0.0.24)	109
1.32.1	New Features	109
1.32.2	Bug fixes	109
1.32.3	Code Refactoring	110
1.32.4	Unit Test and CICD	110
1.32.5	Other Improvements	110
1.33	Release Note (v0.0.23)	110
1.33.1	New Features	110
1.33.2	Bug fixes	110
1.33.3	Documentation	111
1.33.4	Unit Test and CICD	111
1.33.5	Other Improvements	111
2	Tutorials	113
3	Indices and tables	115
	Python Module Index	117
	Index	119



GNES *jee-nes* is **Generic Neural Elastic Search**, a cloud-native semantic search system based on deep neural network.

GNES enables large-scale index and semantic search for **text-to-text**, **image-to-image**, **video-to-video** and *any-to-any* content form.

1.1 All Microservices in GNES

[32mGNES v0.0.46: Generic Neural Elastic Search[0m, a cloud-native semantic search system based on deep neural network. It enables large-scale index and semantic search for text-to-text, image-to-image, video-to-video and any content form. Visit [4m[36m<https://gnes.ai>[0m for tutorials and documentations.

```
usage: gnes [-h] [-v] [--verbose]
           {frontend,encode,index,route,preprocess,grpc,client,compose,healthcheck}
           ...
```

1.1.1 Named Arguments

-v, --version	show program's version number and exit
--verbose	turn on detailed logging for debug
	Default: False

1.1.2 GNES sub-commands

use “gnes [sub-command] –help” to get detailed information about each sub-command

cli	Possible choices: frontend, encode, index, route, preprocess, grpc, client, compose, healthcheck
------------	--

1.1.3 Sub-commands:

frontend

start a frontend service

```
gnes frontend [-h] [--port_in PORT_IN] [--port_out PORT_OUT]
              [--host_in HOST_IN] [--host_out HOST_OUT]
              [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,
↪SUB_CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
              [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,
↪SUB_CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
              [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
              [--dump_interval DUMP_INTERVAL] [--read_only]
              [--parallel_backend {thread,process}]
              [--num_parallel NUM_PARALLEL]
              [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
              [--check_version] [--identity IDENTITY] [--route_table]
              [--squeeze_pb] [--ctrl_with_ipc] [--grpc_host GRPC_HOST]
              [--grpc_port GRPC_PORT] [--max_message_size MAX_MESSAGE_SIZE]
              [--proxy] [--max_concurrency MAX_CONCURRENCY]
              [--dump_route DUMP_ROUTE]
              [--max_pending_request MAX_PENDING_REQUEST]
```

Named Arguments

--port_in	port for input data, default a random port between [49152, 65536] Default: 52205
--port_out	port for output data, default a random port between [49152, 65536] Default: 62219
--host_in	host address for input Default: "0.0.0.0"
--host_out	host address for output Default: "0.0.0.0"
--socket_in	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND
--socket_out	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
--port_ctrl	port for controlling the service, default a random port between [49152, 65536] Default: 60389

- timeout** timeout (ms) of all communication, -1 for waiting forever
Default: -1
- dump_interval** serialize the model in the service every n seconds if model changes. -1 means `--read_only`.
Default: 5
- read_only** do not allow the service to modify the model, `dump_interval` will be ignored
Default: True
- parallel_backend** Possible choices: `thread`, `process`
parallel backend of the service
Default: "thread"
- num_parallel, --replicas** number of parallel services running at the same time (i.e. replicas), `port_in` and `port_out` will be set to random, and routers will be added automatically when necessary
Default: 1
- parallel_type, --replica_type** Possible choices: `PUSH_BLOCK`, `PUSH_NONBLOCK`, `PUB_BLOCK`, `PUB_NONBLOCK`
parallel type of the concurrent services
Default: `PUSH_NONBLOCK`
- check_version, --no-check_version, --no_check_version** comparing the GNES and proto version of incoming message with local setup, mismatch raise an exception
Default: True
- identity** identity of the service, empty by default
Default: ""
- route_table, --no-route_table, --no_route_table** showing a route table with time cost after receiving the result
Default: False
- squeeze_pb, --no-squeeze_pb, --no_squeeze_pb** sending bytes and ndarray separately apart from the protobuf message, usually yields better network efficiency
Default: True
- ctrl_with_ipc** use ipc protocol for control socket
Default: False
- grpc_host** host address of the grpc service
Default: "0.0.0.0"
- grpc_port** host port of the grpc service
Default: 8800
- max_message_size** maximum send and receive size for grpc server in bytes, -1 means unlimited
Default: -1

- proxy, --no-proxy, --no_proxy** respect the `http_proxy` and `https_proxy` environment variables. otherwise, it will unset these proxy variables before start. gRPC seems prefer `--no_proxy`
- Default: False
- max_concurrency** maximum concurrent connections allowed
- Default: 10
- dump_route** dumping route information to a file
- max_pending_request** maximum number of pending requests allowed, when exceed wait until we receive the response
- Default: 100

encode

start an encoder service

```
gnes encode [-h] [--port_in PORT_IN] [--port_out PORT_OUT] [--host_in HOST_IN]
            [--host_out HOST_OUT]
            [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
            [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
            [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
            [--dump_interval DUMP_INTERVAL] [--read_only]
            [--parallel_backend {thread,process}]
            [--num_parallel NUM_PARALLEL]
            [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
            [--check_version] [--identity IDENTITY] [--route_table]
            [--squeeze_pb] [--ctrl_with_ipc] --yaml_path YAML_PATH
            [--py_path PY_PATH [PY_PATH ...]]
```

Named Arguments

- port_in** port for input data, default a random port between [49152, 65536]
- Default: 59435
- port_out** port for output data, default a random port between [49152, 65536]
- Default: 59045
- host_in** host address for input
- Default: "0.0.0.0"
- host_out** host address for output
- Default: "0.0.0.0"
- socket_in** Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT
- socket type for input port
- Default: PULL_BIND

- socket_out** Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT
socket type for output port
Default: PUSH_BIND
- port_ctrl** port for controlling the service, default a random port between [49152, 65536]
Default: 65372
- timeout** timeout (ms) of all communication, -1 for waiting forever
Default: -1
- dump_interval** serialize the model in the service every n seconds if model changes. -1 means `--read_only`.
Default: 5
- read_only** do not allow the service to modify the model, `dump_interval` will be ignored
Default: False
- parallel_backend** Possible choices: thread, process
parallel backend of the service
Default: "thread"
- num_parallel, --replicas** number of parallel services running at the same time (i.e. replicas), `port_in` and `port_out` will be set to random, and routers will be added automatically when necessary
Default: 1
- parallel_type, --replica_type** Possible choices: PUSH_BLOCK, PUSH_NONBLOCK, PUB_BLOCK, PUB_NONBLOCK
parallel type of the concurrent services
Default: PUSH_NONBLOCK
- check_version, --no-check_version, --no_check_version** comparing the GNES and proto version of incoming message with local setup, mismatch raise an exception
Default: True
- identity** identity of the service, empty by default
Default: ""
- route_table, --no-route_table, --no_route_table** showing a route table with time cost after receiving the result
Default: False
- squeeze_pb, --no-squeeze_pb, --no_squeeze_pb** sending bytes and ndarray separately apart from the protobuf message, usually yields better network efficiency
Default: True
- ctrl_with_ipc** use ipc protocol for control socket
Default: False

- yaml_path** yaml config of the service, it should be a readable stream, or a valid file path, or a supported class name.
- py_path** the file path(s) of an external python module(s).

index

start an indexer service

```
gnex index [-h] [--port_in PORT_IN] [--port_out PORT_OUT] [--host_in HOST_IN]
           [--host_out HOST_OUT]
           [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
           [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
           [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
           [--dump_interval DUMP_INTERVAL] [--read_only]
           [--parallel_backend {thread,process}] [--num_parallel NUM_PARALLEL]
           [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
           [--check_version] [--identity IDENTITY] [--route_table]
           [--squeeze_pb] [--ctrl_with_ipc] --yaml_path YAML_PATH
           [--py_path PY_PATH [PY_PATH ...]] [--sorted_response]
           [--as_response AS_RESPONSE]
```

Named Arguments

- port_in** port for input data, default a random port between [49152, 65536]
Default: 55077
- port_out** port for output data, default a random port between [49152, 65536]
Default: 56152
- host_in** host address for input
Default: "0.0.0.0"
- host_out** host address for output
Default: "0.0.0.0"
- socket_in** Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT
socket type for input port
Default: PULL_BIND
- socket_out** Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT
socket type for output port
Default: PUSH_BIND
- port_ctrl** port for controlling the service, default a random port between [49152, 65536]
Default: 58790

- timeout** timeout (ms) of all communication, -1 for waiting forever
Default: -1
- dump_interval** serialize the model in the service every n seconds if model changes. -1 means `--read_only`.
Default: 5
- read_only** do not allow the service to modify the model, `dump_interval` will be ignored
Default: False
- parallel_backend** Possible choices: `thread`, `process`
parallel backend of the service
Default: "thread"
- num_parallel, --replicas** number of parallel services running at the same time (i.e. replicas), `port_in` and `port_out` will be set to random, and routers will be added automatically when necessary
Default: 1
- parallel_type, --replica_type** Possible choices: `PUSH_BLOCK`, `PUSH_NONBLOCK`, `PUB_BLOCK`, `PUB_NONBLOCK`
parallel type of the concurrent services
Default: `PUSH_NONBLOCK`
- check_version, --no-check_version, --no_check_version** comparing the GNES and proto version of incoming message with local setup, mismatch raise an exception
Default: True
- identity** identity of the service, empty by default
Default: ""
- route_table, --no-route_table, --no_route_table** showing a route table with time cost after receiving the result
Default: False
- squeeze_pb, --no-squeeze_pb, --no_squeeze_pb** sending bytes and ndarray separately apart from the protobuf message, usually yields better network efficiency
Default: True
- ctrl_with_ipc** use ipc protocol for control socket
Default: False
- yaml_path** yaml config of the service, it should be a readable stream, or a valid file path, or a supported class name.
- py_path** the file path(s) of an external python module(s).
- sorted_response** sort the response (if exist) by the score
Default: False
- as_response** convert the message type from request to response after indexing. turn it off if you want to chain other services after this index service.
Default: True

route

start a router service

```
gnes route [-h] [--port_in PORT_IN] [--port_out PORT_OUT] [--host_in HOST_IN]
           [--host_out HOST_OUT]
           [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
           [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
           [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
           [--dump_interval DUMP_INTERVAL] [--read_only]
           [--parallel_backend {thread,process}] [--num_parallel NUM_PARALLEL]
           [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
           [--check_version] [--identity IDENTITY] [--route_table]
           [--squeeze_pb] [--ctrl_with_ipc] --yaml_path YAML_PATH
           [--py_path PY_PATH [PY_PATH ...]] [--sorted_response]
           [--num_part NUM_PART]
```

Named Arguments

--port_in	port for input data, default a random port between [49152, 65536] Default: 51079
--port_out	port for output data, default a random port between [49152, 65536] Default: 53005
--host_in	host address for input Default: "0.0.0.0"
--host_out	host address for output Default: "0.0.0.0"
--socket_in	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND
--socket_out	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
--port_ctrl	port for controlling the service, default a random port between [49152, 65536] Default: 65215
--timeout	timeout (ms) of all communication, -1 for waiting forever Default: -1

- dump_interval** serialize the model in the service every n seconds if model changes. -1 means `--read_only`.
Default: 5
- read_only** do not allow the service to modify the model, `dump_interval` will be ignored
Default: True
- parallel_backend** Possible choices: `thread`, `process`
parallel backend of the service
Default: "thread"
- num_parallel, --replicas** number of parallel services running at the same time (i.e. replicas), `port_in` and `port_out` will be set to random, and routers will be added automatically when necessary
Default: 1
- parallel_type, --replica_type** Possible choices: `PUSH_BLOCK`, `PUSH_NONBLOCK`, `PUB_BLOCK`, `PUB_NONBLOCK`
parallel type of the concurrent services
Default: `PUSH_NONBLOCK`
- check_version, --no-check_version, --no_check_version** comparing the GNES and proto version of incoming message with local setup, mismatch raise an exception
Default: True
- identity** identity of the service, empty by default
Default: ""
- route_table, --no-route_table, --no_route_table** showing a route table with time cost after receiving the result
Default: False
- squeeze_pb, --no-squeeze_pb, --no_squeeze_pb** sending bytes and ndarray separately apart from the protobuf message, usually yields better network efficiency
Default: True
- ctrl_with_ipc** use ipc protocol for control socket
Default: False
- yaml_path** yaml config of the service, it should be a readable stream, or a valid file path, or a supported class name.
- py_path** the file path(s) of an external python module(s).
- sorted_response** sort the response (if exist) by the score
Default: False
- num_part** explicitly set the number of parts of message

preprocess

start a preprocessor service

```
gnex preprocess [-h] [--port_in PORT_IN] [--port_out PORT_OUT]
                [--host_in HOST_IN] [--host_out HOST_OUT]
                [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,
↪SUB_CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
                [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,
↪SUB_CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
                [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
                [--dump_interval DUMP_INTERVAL] [--read_only]
                [--parallel_backend {thread,process}]
                [--num_parallel NUM_PARALLEL]
                [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
                [--check_version] [--identity IDENTITY] [--route_table]
                [--squeeze_pb] [--ctrl_with_ipc] --yaml_path YAML_PATH
                [--py_path PY_PATH [PY_PATH ...]]
```

Named Arguments

--port_in	port for input data, default a random port between [49152, 65536] Default: 61289
--port_out	port for output data, default a random port between [49152, 65536] Default: 62380
--host_in	host address for input Default: "0.0.0.0"
--host_out	host address for output Default: "0.0.0.0"
--socket_in	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND
--socket_out	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
--port_ctrl	port for controlling the service, default a random port between [49152, 65536] Default: 64169
--timeout	timeout (ms) of all communication, -1 for waiting forever Default: -1
--dump_interval	serialize the model in the service every n seconds if model changes. -1 means -read_only. Default: 5

- read_only** do not allow the service to modify the model, `dump_interval` will be ignored
Default: True
- parallel_backend** Possible choices: `thread`, `process`
parallel backend of the service
Default: “thread”
- num_parallel, --replicas** number of parallel services running at the same time (i.e. replicas), `port_in` and `port_out` will be set to random, and routers will be added automatically when necessary
Default: 1
- parallel_type, --replica_type** Possible choices: `PUSH_BLOCK`, `PUSH_NONBLOCK`, `PUB_BLOCK`, `PUB_NONBLOCK`
parallel type of the concurrent services
Default: `PUSH_NONBLOCK`
- check_version, --no-check_version, --no_check_version** comparing the GNES and proto version of incoming message with local setup, mismatch raise an exception
Default: True
- identity** identity of the service, empty by default
Default: “”
- route_table, --no-route_table, --no_route_table** showing a route table with time cost after receiving the result
Default: False
- squeeze_pb, --no-squeeze_pb, --no_squeeze_pb** sending bytes and ndarray separately apart from the protobuf message, usually yields better network efficiency
Default: True
- ctrl_with_ipc** use ipc protocol for control socket
Default: False
- yaml_path** yaml config of the service, it should be a readable stream, or a valid file path, or a supported class name.
- py_path** the file path(s) of an external python module(s).

grpc

start a general purpose grpc service

```

gnes grpc [-h] [--port_in PORT_IN] [--port_out PORT_OUT] [--host_in HOST_IN]
          [--host_out HOST_OUT]
          [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
          [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
          [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
          [--dump_interval DUMP_INTERVAL] [--read_only]
          [--parallel_backend {thread,process}] [--num_parallel NUM_PARALLEL]

```

(continues on next page)

(continued from previous page)

```

[--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
[--check_version] [--identity IDENTITY] [--route_table]
[--squeeze_pb] [--ctrl_with_ipc] [--grpc_host GRPC_HOST]
[--grpc_port GRPC_PORT] [--max_message_size MAX_MESSAGE_SIZE]
[--proxy] --pb2_path PB2_PATH --pb2_grpc_path PB2_GRPC_PATH
--stub_name STUB_NAME --api_name API_NAME

```

Named Arguments

--port_in	port for input data, default a random port between [49152, 65536] Default: 57781
--port_out	port for output data, default a random port between [49152, 65536] Default: 50682
--host_in	host address for input Default: "0.0.0.0"
--host_out	host address for output Default: "0.0.0.0"
--socket_in	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND
--socket_out	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
--port_ctrl	port for controlling the service, default a random port between [49152, 65536] Default: 63259
--timeout	timeout (ms) of all communication, -1 for waiting forever Default: -1
--dump_interval	serialize the model in the service every n seconds if model changes. -1 means -read_only. Default: 5
--read_only	do not allow the service to modify the model, dump_interval will be ignored Default: False
--parallel_backend	Possible choices: thread, process parallel backend of the service Default: "thread"

- num_parallel, --replicas** number of parallel services running at the same time (i.e. replicas), *port_in* and *port_out* will be set to random, and routers will be added automatically when necessary
Default: 1
- parallel_type, --replica_type** Possible choices: PUSH_BLOCK, PUSH_NONBLOCK, PUB_BLOCK, PUB_NONBLOCK
parallel type of the concurrent services
Default: PUSH_NONBLOCK
- check_version, --no-check_version, --no_check_version** comparing the GNES and proto version of incoming message with local setup, mismatch raise an exception
Default: True
- identity** identity of the service, empty by default
Default: ""
- route_table, --no-route_table, --no_route_table** showing a route table with time cost after receiving the result
Default: False
- squeeze_pb, --no-squeeze_pb, --no_squeeze_pb** sending bytes and ndarray separately apart from the protobuf message, usually yields better network efficiency
Default: True
- ctrl_with_ipc** use ipc protocol for control socket
Default: False
- grpc_host** host address of the grpc service
Default: "0.0.0.0"
- grpc_port** host port of the grpc service
Default: 8800
- max_message_size** maximum send and receive size for grpc server in bytes, -1 means unlimited
Default: -1
- proxy, --no-proxy, --no_proxy** respect the http_proxy and https_proxy environment variables. otherwise, it will unset these proxy variables before start. gRPC seems prefer `--no_proxy`
Default: False
- pb2_path** the path of the python file protocol buffer compiler
- pb2_grpc_path** the path of the python file generated by the gRPC Python protocol compiler plugin
- stub_name** the name of the gRPC Stub
- api_name** the api name for calling the stub

client

start a GNES client of the selected type

```
gnес client [-h] {http,cli} ...
```

GNES client sub-commands

use “gnес client [sub-command] –help” to get detailed information about each client sub-command

client Possible choices: http, cli

Sub-commands:

http

start a client that allows HTTP requests as input

```
gnес client http [-h] [--grpc_host GRPC_HOST] [--grpc_port GRPC_PORT]
                 [--max_message_size MAX_MESSAGE_SIZE] [--proxy]
                 [--http_port HTTP_PORT] [--http_host HTTP_HOST]
                 [--max_workers MAX_WORKERS] [--top_k TOP_K]
                 [--batch_size BATCH_SIZE]
```

Named Arguments

--grpc_host	host address of the grpc service Default: “0.0.0.0”
--grpc_port	host port of the grpc service Default: 8800
--max_message_size	maximum send and receive size for grpc server in bytes, -1 means unlimited Default: -1
--proxy, --no-proxy, --no_proxy	respect the http_proxy and https_proxy environment variables. otherwise, it will unset these proxy variables before start. gRPC seems prefer –no_proxy Default: False
--http_port	http port to deploy the service Default: 80
--http_host	http host to deploy the service Default: “0.0.0.0”
--max_workers	max workers to deal with the message Default: 100
--top_k	default top_k for query mode Default: 10

--batch_size batch size for feed data for train mode
Default: 2560

cli

start a client that allows stdin as input

```

gnex client cli [-h] [--grpc_host GRPC_HOST] [--grpc_port GRPC_PORT]
                [--max_message_size MAX_MESSAGE_SIZE] [--proxy]
                [--txt_file TXT_FILE | --image_zip_file IMAGE_ZIP_FILE | --video_zip_
↵file VIDEO_ZIP_FILE]
                [--batch_size BATCH_SIZE] --mode {index,query,train}
                [--top_k TOP_K] [--start_doc_id START_DOC_ID]
                [--max_concurrency MAX_CONCURRENCY]

```

Named Arguments

--grpc_host host address of the grpc service
Default: "0.0.0.0"

--grpc_port host port of the grpc service
Default: 8800

--max_message_size maximum send and receive size for grpc server in bytes, -1 means unlimited
Default: -1

--proxy, --no-proxy, --no_proxy respect the http_proxy and https_proxy environment variables. otherwise, it will unset these proxy variables before start. gRPC seems prefer `--no_proxy`
Default: False

--txt_file text file to be used, each line is a doc/query
Default: `<_io.TextIOWrapper name='<stdin>' mode='r' encoding='UTF-8'>`

--image_zip_file image zip file to be used, consists of multiple images

--video_zip_file video zip file to be used, consists of multiple videos

--batch_size the size of the request to split
Default: 100

--mode Possible choices: index, query, train
the mode of the client and the server

--top_k top_k results returned in the query mode
Default: 10

--start_doc_id the start number of doc id
Default: 0

--max_concurrency maximum concurrent connections allowed
Default: 10

compose

start a GNES Board to visualize YAML configs

```
gnex compose [-h] [--port PORT] [--name NAME] [--yaml_path YAML_PATH]
              [--html_path HTML_PATH] [--shell_path SHELL_PATH]
              [--swarm_path SWARM_PATH] [--k8s_path K8S_PATH]
              [--graph_path GRAPH_PATH]
              [--shell_log_redirect SHELL_LOG_REDIRECT] [--mermaid_leftright]
              [--docker_img DOCKER_IMG] [--flask | --serve]
              [--http_port HTTP_PORT]
```

Named Arguments

--port	host port of the grpc service Default: 8800
--name	name of the instance Default: “GNES app”
--yaml_path	yaml config of the service Default: <code><_io.BufferedReader name='/home/docs/checkouts/readthedocs.org/user_builds/gnes/checkouts/example.yml'></code>
--html_path	output path of the HTML file, will contain all possible generations
--shell_path	output path of the shell-based starting script
--swarm_path	output path of the docker-compose file for Docker Swarm
--k8s_path	output path of the docker-compose file for Docker Swarm
--graph_path	output path of the mermaid graph file
--shell_log_redirect	the file path for redirecting shell output. when not given, the output will be flushed to stdout
--mermaid_leftright	showing the flow in left-to-right manner rather than top down Default: False
--docker_img	the docker image used in Docker Swarm & Kubernetes Default: “gnes/gnes:latest-alpine”
--flask	start a Flask server and serve the composer in interactive mode, aka GNES board Default: False
--serve	start a basic HTTP server and serve the composer in interactive mode, aka GNES board Default: False
--http_port	server port for receiving HTTP requests Default: 8080

healthcheck

do health check on any GNES microservice

```
gnes healthcheck [-h] [--host HOST] --port PORT [--timeout TIMEOUT]
                [--retries RETRIES]
```

Named Arguments

--host	host address of the checked service Default: "127.0.0.1"
--port	control port of the checked service
--timeout	timeout (ms) of one check, -1 for waiting forever Default: 1000
--retries	max number of tried health checks before exit 1 Default: 3

1.2 gnes package

1.2.1 Subpackages

gnes.base package

Module contents

class `gnes.base.TrainableBase` (*args, **kwargs)

Bases: object

The base class for preprocessor, encoder, indexer and router

close ()

Release the resources as model is destroyed

dump (filename=None)

Serialize the object to a binary file

Parameters `filename` (Optional[str]) – file path of the serialized file, if not given then `dump_full_path` is used

Return type None

dump_full_path

Get the binary dump path

Returns

dump_yaml (filename=None)

Serialize the object to a yaml file

Parameters `filename` (Optional[str]) – file path of the yaml file, if not given then `dump_yaml_path` is used

Return type None

classmethod `from_yaml` (*constructor, node, stop_on_import_error=False*)

static `load` (*filename=None*)

Return type ~T

classmethod `load_yaml` (*filename*)

Return type ~T

post_init ()

Declare class attributes/members that can not be serialized in standard way

classmethod `pre_init` ()

store_args_kwargs = False

classmethod `to_yaml` (*representer, data*)

train (**args, **kwargs*)

Train the model, need to be overridden

yaml_full_path

Get the file path of the yaml config

Returns

class `gn.es.base.CompositionalTrainableBase` (**args, **kwargs*)

Bases: `gn.es.base.TrainableBase`

close ()

Release the resources as model is destroyed

components

Return type Union[List[~T], Dict[str, ~T]]

classmethod `from_yaml` (*constructor, node*)

is_pipeline

is_trained

classmethod `to_yaml` (*representer, data*)

train (**args, **kwargs*)

Train the model, need to be overridden

gn.es.cli package

Submodules

gn.es.cli.api module

`gn.es.cli.api.client` (*args*)

`gn.es.cli.api.compose` (*args*)

`gn.es.cli.api.encode` (*args*)

`gn.es.cli.api.frontend` (*args*)

`gn.es.cli.api.grpc` (*args*)

`gnes.cli.api.healthcheck` (*args*)

`gnes.cli.api.index` (*args*)

`gnes.cli.api.preprocess` (*args*)

`gnes.cli.api.route` (*args*)

gnes.cli.parser module

class `gnes.cli.parser.ActionNoYes` (*option_strings*, *dest*, *default=None*, *required=False*,
help=None)

Bases: `argparse.Action`

`gnes.cli.parser.get_main_parser` ()

`gnes.cli.parser.random_port` (*port*)

`gnes.cli.parser.resolve_py_path` (*path*)

`gnes.cli.parser.resolve_yaml_path` (*path*, *to_stream=False*)

`gnes.cli.parser.set_base_parser` ()

`gnes.cli.parser.set_client_cli_parser` (*parser=None*)

`gnes.cli.parser.set_client_http_parser` (*parser=None*)

`gnes.cli.parser.set_composer_flask_parser` (*parser=None*)

`gnes.cli.parser.set_composer_parser` (*parser=None*)

`gnes.cli.parser.set_encoder_parser` (*parser=None*)

`gnes.cli.parser.set_frontend_parser` (*parser=None*)

`gnes.cli.parser.set_grpc_service_parser` (*parser=None*)

`gnes.cli.parser.set_healthcheck_parser` (*parser=None*)

`gnes.cli.parser.set_indexer_parser` (*parser=None*)

`gnes.cli.parser.set_preprocessor_parser` (*parser=None*)

`gnes.cli.parser.set_router_parser` (*parser=None*)

`gnes.cli.parser.set_service_parser` (*parser=None*)

Module contents

`gnes.cli.main` ()

gnes.client package

Submodules

gnes.client.base module

class `gnes.client.base.GrpcClient` (*args*)

Bases: `object`

A Base Unary gRPC client which the other client application can build from.

call (*request*)

close ()

handler = <gnes.client.base.ResponseHandler object>

start ()

stream_call (*request_iterator*)

class gnes.client.base.ResponseHandler (*h=None*)

Bases: object

call_routes (*resp*)

register (*resp_type*)

class gnes.client.base.ZmqClient (*args*)

Bases: object

close ()

recv_message (***kwargs*)

Return type Message

send_message (*message*, ***kwargs*)

gnes.client.cli module

class gnes.client.cli.CLIClient (*args*, *start_at_init=True*)

Bases: *gnes.client.base.GrpcClient*

bytes_generator

Return type Iterator[bytes]

index ()

Return type None

query ()

Return type Iterator[Tuple]

start ()

train ()

Return type None

class gnes.client.cli.ProgressBar (*bar_len=20*, *task_name=""*)

Bases: object

update ()

gnes.client.http module

class gnes.client.http.HttpClient (*args=None*)

Bases: object

start ()

gnes.client.stream module

```

class gnes.client.stream.StreamingClient (args)
    Bases: gnes.client.base.GrpcClient

    close ()

    handler = <gnes.client.base.ResponseHandler object>

    send_request (request)

class gnes.client.stream.SyncClient (args)
    Bases: gnes.client.base.GrpcClient

    close ()

    handler = <gnes.client.base.ResponseHandler object>

    send_request (request)

```

Module contents

gnes.composer package

Submodules

gnes.composer.base module

```

class gnes.composer.base.YamlComposer (args)
    Bases: object

    class Layer (layer_id=0)
        Bases: object

        append (comp)

        default_values = {'image': None, 'income': 'pull', 'name': None, 'py_path': None}

        get_component_name

        static get_value (comp, key)

        is_heto_single_component

        is_homo_multi_component

        is_homogenous

        is_single_component

    add_comp (comp)

        Return type None

    add_layer (layer=None)

        Return type None

    build_all ()

    static build_dockerswarm (all_layers, docker_img='gnes/gnes:latest-alpine', volumes=None,
                             networks=None)

        Return type str

```

```
static build_html (generate_dict)
    Return type str
static build_kubernetes (all_layers, *args, **kwargs)
build_layers ()
    Return type List[Layer]
static build_mermaid (all_layers, mermaid_leftright=False)
    Return type str
static build_shell (all_layers, log_redirect=None)
    Return type str
check_fields (comp)
    Return type bool
comp2args = {'Encoder': Namespace(check_version=True, ctrl_with_ipc=False, dump_inter
comp2file = {'Encoder': 'encode', 'Frontend': 'frontend', 'Indexer': 'index', 'Prep
gnes.composer.base.parse_http_data (data, args)
```

gnes.composer.flask module

```
class gnes.composer.flask.YamlComposerFlask (args)
    Bases: object
    run ()
```

gnes.composer.http module

```
class gnes.composer.http.YamlComposerHttp (args)
    Bases: object
    run ()
```

Module contents

gnes.encoder package

Subpackages

gnes.encoder.audio package

Subpackages

gnes.encoder.audio.vggish_cores package

Submodules

gnes.encoder.audio.vggish_cores.vggish_params module

Global parameters for the VGGish model.

See `vggish_slim.py` for more information.

gnes.encoder.audio.vggish_cores.vggish_postprocess module

Post-process embeddings from VGGish.

class `gnes.encoder.audio.vggish_cores.vggish_postprocess.Postprocessor` (*pca_params_npz_path*)
Bases: `object`

Post-processes VGGish embeddings.

The initial release of AudioSet included 128-D VGGish embeddings for each segment of AudioSet. These released embeddings were produced by applying a PCA transformation (technically, a whitening transform is included as well) and 8-bit quantization to the raw embedding output from VGGish, in order to stay compatible with the YouTube-8M project which provides visual embeddings in the same format for a large set of YouTube videos. This class implements the same PCA (with whitening) and quantization transformations.

Constructs a postprocessor.

Args:

pca_params_npz_path: Path to a NumPy-format .npz file that contains the PCA parameters used in postprocessing.

postprocess (*embeddings_batch*)

Applies postprocessing to a batch of embeddings.

Args:

embeddings_batch: An `ndarray` of shape `[batch_size, embedding_size]` containing output from the embedding layer of VGGish.

Returns: An `ndarray` of the same shape as the input but of type `uint8`, containing the PCA-transformed and quantized version of the input.

gnes.encoder.audio.vggish_cores.vggish_slim module

Module contents

Submodules

gnes.encoder.audio.mfcc module

class `gnes.encoder.audio.mfcc.MfccEncoder` (*n_mfcc=13*, *sample_rate=16000*,
max_length=100, *args, **kwargs)
Bases: `gnes.encoder.base.BaseAudioEncoder`

batch_size = 64

encode (*data*, *args, **kwargs)

Return type `ndarray`

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

gnes.encoder.audio.vggish module

```
class gnes.encoder.audio.vggish.VggishEncoder (model_dir, max_length=10, *args,
                                              **kwargs)
```

```
    Bases: gnes.encoder.base.BaseAudioEncoder
```

```
encode (audio, *args, **kwargs)
```

```
    Return type ndarray
```

```
post_init ()
```

```
    Declare class attributes/members that can not be serialized in standard way
```

```
train (*args, **kwargs)
```

```
    Train the model, need to be overridden
```

Module contents

gnes.encoder.image package

Subpackages

gnes.encoder.image.cvae_cores package

Submodules

gnes.encoder.image.cvae_cores.model module

Module contents

gnes.encoder.image.inception_cores package

Submodules

gnes.encoder.image.inception_cores.inception_utils module

gnes.encoder.image.inception_cores.inception_v4 module

Module contents

Submodules

gnes.encoder.image.cvae module

```
class gnes.encoder.image.cvae.CVAEEncoder(model_dir, latent_dim=300, select_method='MEAN', l2_normalize=False, use_gpu=True, *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseImageEncoder*

batch_size = 64

encode (*img*, **args*, ***kwargs*)

Return type ndarray

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (**args*, ***kwargs*)

Train the model, need to be overridden

gnes.encoder.image.inception module

```
class gnes.encoder.image.inception.TFInceptionEncoder(model_dir, select_layer='PreLogitsFlatten', *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseImageEncoder*

batch_size = 64

encode (*img*, **args*, ***kwargs*)

Return type ndarray

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (**args*, ***kwargs*)

Train the model, need to be overridden

gnes.encoder.image.onnx module

```
class gnes.encoder.image.onnx.BaseONNXImageEncoder(model_name, model_dir, *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseImageEncoder*

batch_size = 64

encode (*img*, **args*, ***kwargs*)

Return type ndarray

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (**args*, ***kwargs*)

Train the model, need to be overridden

gnes.encoder.image.torchvision module

```
class gnes.encoder.image.torchvision.TorchvisionEncoder (model_name,          lay-  
                                                    ers, model_dir, *args,  
                                                    **kwargs)
```

Bases: *gnes.encoder.base.BaseImageEncoder*

```
batch_size = 64
```

```
encode (img, *args, **kwargs)
```

Return type ndarray

```
post_init ()
```

Declare class attributes/members that can not be serialized in standard way

```
train (*args, **kwargs)
```

Train the model, need to be overridden

Module contents

gnes.encoder.numeric package

Submodules

gnes.encoder.numeric.hash module

```
class gnes.encoder.numeric.hash.HashEncoder (num_bytes,          num_bits=8,  
                                                    num_idx=3,          kmeans_clusters=100,  
                                                    method='product_uniform',          *args,  
                                                    **kwargs)
```

Bases: *gnes.encoder.base.BaseNumericEncoder*

```
batch_size = 2048
```

```
encode (vecs, *args, **kwargs)
```

Return type ndarray

```
hash (vecs)
```

```
pred_kmeans (vecs)
```

```
ran_gen ()
```

```
train (vecs, *args, **kwargs)
```

Train the model, need to be overridden

```
train_kmeans (vecs)
```

gnes.encoder.numeric.pca module

```
class gnes.encoder.numeric.pca.PCAEncoder (output_dim, whiten=False, *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseNumericEncoder*

```
batch_size = 2048
```

```
encode (vecs, *args, **kwargs)
```

Return type ndarray

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (*vecs*, *args, **kwargs)

Train the model, need to be overridden

Return type None

class gnes.encoder.numeric.pca.**PCALocalEncoder** (*output_dim*, *num_locals*, *args, **kwargs)

Bases: *gnes.encoder.base.BaseNumericEncoder*

batch_size = 2048

encode (*vecs*, *args, **kwargs)

Return type ndarray

train (*vecs*, *args, **kwargs)

Train the model, need to be overridden

Return type None

gnes.encoder.numeric.pooling module

class gnes.encoder.numeric.pooling.**PoolingEncoder** (*pooling_strategy*='REDUCE_MEAN', *backend*='numpy', *args, **kwargs)

Bases: *gnes.encoder.base.BaseNumericEncoder*

encode (*data*, *args, **kwargs)

masked_reduce_max (*x*, *m*)

masked_reduce_mean (*x*, *m*, *jitter*=1e-10)

minus_mask (*x*, *m*, *offset*=1e+30)

mul_mask (*x*, *m*)

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (*args, **kwargs)

Train the model, need to be overridden

gnes.encoder.numeric.pq module

class gnes.encoder.numeric.pq.**PQEncoder** (*num_bytes*, *cluster_per_byte*=255, *args, **kwargs)

Bases: *gnes.encoder.base.BaseBinaryEncoder*

batch_size = 2048

encode (*vecs*, *args, **kwargs)

Return type ndarray

train (*vecs*, *args, **kwargs)

Train the model, need to be overridden

gnes.encoder.numeric.quantizer module

```
class gnes.encoder.numeric.quantizer.QuantizerEncoder (dim_per_byte, cluster_per_byte=255, upper_bound=10000, lower_bound=-10000, partition_method='average', *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseBinaryEncoder*

batch_size = 2048

encode (*vecs*, **args*, ***kwargs*)

Return type ndarray

train (**args*, ***kwargs*)

Train the model, need to be overridden

gnes.encoder.numeric.standarder module

```
class gnes.encoder.numeric.standarder.StandarderEncoder (*args, **kwargs)
```

Bases: *gnes.encoder.base.BaseNumericEncoder*

batch_size = 2048

encode (*vecs*, **args*, ***kwargs*)

Return type ndarray

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (*vecs*, **args*, ***kwargs*)

Train the model, need to be overridden

Return type None

gnes.encoder.numeric.tf_pq module

```
class gnes.encoder.numeric.tf_pq.TFPQEncoder (num_bytes, cluster_per_byte=255, *args, **kwargs)
```

Bases: *gnes.encoder.numeric.pq.PQEncoder*

batch_size = 8192

close ()

Release the resources as model is destroyed

encode (*vecs*, **args*, ***kwargs*)

Return type ndarray

post_init ()

Declare class attributes/members that can not be serialized in standard way

classmethod pre_init ()

train (*vecs*, **args*, ***kwargs*)

gnes.encoder.numeric.vlad module

```
class gnes.encoder.numeric.vlad.VladEncoder (num_clusters, using_faiss_pred=False,
                                             *args, **kwargs)
    Bases: gnes.encoder.base.BaseNumericEncoder

    batch_size = 2048

    encode (vecs, *args, **kwargs)
        Return type ndarray

    faiss_index ()

    kmeans_pred (vecs)

    kmeans_train (vecs)

    train (vecs, *args, **kwargs)
        Train the model, need to be overridden
```

Module contents

gnes.encoder.text package

Submodules

gnes.encoder.text.bert module

```
class gnes.encoder.text.bert.BertEncoder (*args, **kwargs)
    Bases: gnes.encoder.base.BaseTextEncoder

    close ()
        Release the resources as model is destroyed

    encode (text, *args, **kwargs)
        Return type ndarray

    is_trained = True

    post_init ()
        Declare class attributes/members that can not be serialized in standard way

    store_args_kwargs = True

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.encoder.text.bert.BertEncoderServer (*args, **kwargs)
    Bases: gnes.encoder.base.BaseTextEncoder

    close ()
        Release the resources as model is destroyed

    is_trained = True

    post_init ()
        Declare class attributes/members that can not be serialized in standard way

    store_args_kwargs = True
```

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

```
class gnes.encoder.text.bert.BertEncoderWithServer (*args, **kwargs)
```

```
    Bases: gnes.base.CompositionalTrainableBase
```

```
encode (text, *args, **kwargs)
```

```
    Return type ndarray
```

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

gnes.encoder.text.char module

```
class gnes.encoder.text.char.CharEmbeddingEncoder (dim=128, *args, **kwargs)
```

```
    Bases: gnes.encoder.base.BaseTextEncoder
```

```
    A random character embedding model. Only useful for testing
```

```
encode (text, *args, **kwargs)
```

```
    Return type List[ndarray]
```

```
is_trained = True
```

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

gnes.encoder.text.flair module

```
class gnes.encoder.text.flair.FlairEncoder (word_embedding='glove',
                                             flair_embeddings=('news-forward', 'news-
backward'), pooling_strategy='mean', *args,
                                             **kwargs)
```

```
    Bases: gnes.encoder.base.BaseTextEncoder
```

```
encode (text, *args, **kwargs)
```

```
    Return type ndarray
```

```
is_trained = True
```

```
post_init ()
    Declare class attributes/members that can not be serialized in standard way
```

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

gnes.encoder.text.transformer module

gnes.encoder.text.w2v module

```
class gnes.encoder.text.w2v.Word2VecEncoder (model_dir, skiprows=1, dimension=300,
                                             pooling_strategy='REDUCE_MEAN',
                                             *args, **kwargs)
```

```
    Bases: gnes.encoder.base.BaseTextEncoder
```

```
encode (text, *args, **kwargs)
```

Return type ndarray

is_trained = True

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (*args, **kwargs)

Train the model, need to be overridden

Module contents

gnes.encoder.video package

Subpackages

gnes.encoder.video.mixture_core package

Submodules

gnes.encoder.video.mixture_core.model module

Module contents

gnes.encoder.video.yt8m_feature_extractor_cores package

Submodules

gnes.encoder.video.yt8m_feature_extractor_cores.inception_utils module

gnes.encoder.video.yt8m_feature_extractor_cores.inception_v3 module

Module contents

Submodules

gnes.encoder.video.incep_mixture module

```
class gnes.encoder.video.incep_mixture.IncepMixtureEncoder (model_dir_inception,  
model_dir_mixture,  
se-  
lect_layer='PreLogitsFlatten',  
feature_size=300,  
vocab_size=28,  
cluster_size=256,  
method='fvnet',  
input_size=1536,  
vocab_size_2=174,  
max_frames=30,  
multi-  
task_method='Attention',  
*args, **kwargs)
```

Bases: *gnes.encoder.base.BaseVideoEncoder*

batch_size = 64

encode (*data, *args, **kwargs*)

Return type ndarray

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.encoder.video.inception module

```
class gnes.encoder.video.inception.InceptionVideoEncoder (model_dir,  
se-  
lect_layer='PreLogitsFlatten',  
*args, **kwargs)
```

Bases: *gnes.encoder.base.BaseVideoEncoder*

batch_size = 64

encode (*data, *args, **kwargs*)

Return type List[ndarray]

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.encoder.video.yt8m_feature_extractor module

```
class gnes.encoder.video.yt8m_feature_extractor.YouTube8MFeatureExtractor (model_dir,
                                                                    pca_dir,
                                                                    se-
                                                                    lect_layer='PreLogits',
                                                                    ig-
                                                                    nore_audio_feature=True,
                                                                    *args,
                                                                    **kwargs)
```

Bases: *gnes.encoder.base.BaseVideoEncoder*

Extracts YouTube8M features for RGB frames.

First time constructing this class will create directory *yt8m* inside your home directory, and will download inception model (85 MB) and YouTube8M PCA matrix (15 MB). If you want to use another directory, then pass it to argument *model_dir* of constructor.

If the *model_dir* exist and contains the necessary files, then files will be re-used without download.

Usage Example:

```
from PIL import Image
import numpy

# Instantiate extractor. Slow if called first time on your machine, as it # needs to download 100 MB.
extractor = YouTube8MFeatureExtractor()

image_file = os.path.join(extractor._model_dir, 'cropped_panda.jpg')

im = numpy.array(Image.open(image_file))
features = extractor.extract_rgb_frame_features(im)
```

** Note: OpenCV reverses the order of channels (i.e. orders channels as BGR instead of RGB). If you are using OpenCV, then you must do:

```
im = im[:, :, ::-1] # Reverses order on last (i.e. channel) dimension.
```

then call *extractor.extract_rgb_frame_features(im)*

batch_size = 64

encode (*data, *args, **kwargs*)

Return type List[ndarray]

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.encoder.video.yt8m_model module

```
class gnes.encoder.video.yt8m_model.YouTube8MEncoder (model_dir, model_name, *args,
                                                                    **kwargs)
```

Bases: *gnes.encoder.base.BaseVideoEncoder*

batch_size = 64

encode (*data, *args, **kwargs*)

Return type ndarray

post_init ()
Declare class attributes/members that can not be serialized in standard way

train (*args, **kwargs)
Train the model, need to be overridden

Module contents

Submodules

gnes.encoder.base module

class gnes.encoder.base.**BaseAudioEncoder** (*args, **kwargs)
Bases: *gnes.encoder.base.BaseEncoder*

encode (data, *args, **kwargs)
Return type ndarray

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.encoder.base.**BaseBinaryEncoder** (*args, **kwargs)
Bases: *gnes.encoder.base.BaseEncoder*

encode (data, *args, **kwargs)
Return type bytes

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.encoder.base.**BaseEncoder** (*args, **kwargs)
Bases: *gnes.base.TrainableBase*

encode (data, *args, **kwargs)
Return type Any

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.encoder.base.**BaseImageEncoder** (*args, **kwargs)
Bases: *gnes.encoder.base.BaseEncoder*

encode (img, *args, **kwargs)
Return type ndarray

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.encoder.base.**BaseNumericEncoder** (*args, **kwargs)
Bases: *gnes.encoder.base.BaseEncoder*

Note that all NumericEncoder can not be used as the first encoder of the pipeline

encode (data, *args, **kwargs)
Return type ndarray

train (*args, **kwargs)
Train the model, need to be overridden

```

class gnes.encoder.base.BaseTextEncoder (*args, **kwargs)
    Bases: gnes.encoder.base.BaseEncoder

    encode (text, *args, **kwargs)

        Return type Union[Tuple, ndarray]

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.encoder.base.BaseVideoEncoder (*args, **kwargs)
    Bases: gnes.encoder.base.BaseEncoder

    encode (data, *args, **kwargs)

        Return type Union[ndarray, List[ndarray]]

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.encoder.base.PipelineEncoder (*args, **kwargs)
    Bases: gnes.base.CompositionalTrainableBase

    encode (data, *args, **kwargs)

        Return type Any

    train (data, *args, **kwargs)
        Train the model, need to be overridden

```

Module contents

gnes.flow package

Submodules

gnes.flow.base module

```

class gnes.flow.base.BaseIndexFlow (*args, **kwargs)
    Bases: gnes.flow.Flow

```

BaseIndexFlow defines a common service pipeline when indexing.

It can not be directly used as all services are using the base module by default. You have to use `set()` to change the `yaml_path` of each service.

```

train (bytes_gen=None, **kwargs)
    Do training on the current flow

```

It will start a `CLIClient` and call `train()`.

Example,

```

with f.build(backend='thread') as flow:
    flow.train(txt_file='aa.txt')
    flow.train(image_zip_file='aa.zip', batch_size=64)
    flow.train(video_zip_file='aa.zip')
    ...

```

This will call the pre-built reader to read files into an iterator of bytes and feed to the flow.

One may also build a reader/generator on your own.

Example,

```
def my_reader():
    for _ in range(10):
        yield b'abcdefg' # each yield generates a document for training

with f.build(backend='thread') as flow:
    flow.train(bytes_gen=my_reader())
```

Parameters

- **bytes_gen** (Optional[Iterator[bytes]]) – An iterator of bytes. If not given, then you have to specify it in *kwargs*.
- **kwargs** – accepts all keyword arguments of *gnes client* CLI

```
class gnes.flow.base.BaseQueryFlow(*args, **kwargs)
```

Bases: *gnes.flow.Flow*

BaseIndexFlow defines a common service pipeline when indexing.

It can not be directly used as all services are using the base module by default. You have to use `set()` to change the *yaml_path* of each service.

```
train(bytes_gen=None, **kwargs)
```

Do training on the current flow

It will start a *CLIClient* and call *train()*.

Example,

```
with f.build(backend='thread') as flow:
    flow.train(txt_file='aa.txt')
    flow.train(image_zip_file='aa.zip', batch_size=64)
    flow.train(video_zip_file='aa.zip')
    ...
```

This will call the pre-built reader to read files into an iterator of bytes and feed to the flow.

One may also build a reader/generator on your own.

Example,

```
def my_reader():
    for _ in range(10):
        yield b'abcdefg' # each yield generates a document for training

with f.build(backend='thread') as flow:
    flow.train(bytes_gen=my_reader())
```

Parameters

- **bytes_gen** (Optional[Iterator[bytes]]) – An iterator of bytes. If not given, then you have to specify it in *kwargs*.
- **kwargs** – accepts all keyword arguments of *gnes client* CLI

gnes.flow.helper module

class gnes.flow.helper.**BuildLevel**

Bases: *gnes.service.base.BetterEnum*

An enumeration.

EMPTY = 0

GRAPH = 1

RUNTIME = 2

exception gnes.flow.helper.**FlowBuildLevelMismatch**

Bases: *ValueError*

Exception when required level is higher than the current build level

exception gnes.flow.helper.**FlowIncompleteError**

Bases: *ValueError*

Exception when the flow missing some important component to run

exception gnes.flow.helper.**FlowMissingNode**

Bases: *ValueError*

Exception when the topology is ambiguous

exception gnes.flow.helper.**FlowTopologyError**

Bases: *ValueError*

Exception when the topology is ambiguous

class gnes.flow.helper.**Service**

Bases: *gnes.service.base.BetterEnum*

An enumeration.

Encoder = 1

Frontend = 0

Indexer = 3

Preprocessor = 4

Router = 2

gnes.flow.helper.**build_required**(*required_level*)

Module contents

class gnes.flow.**Flow**(*with_frontend=True, is_trained=True, *args, **kwargs*)

Bases: *gnes.base.TrainableBase*

GNES Flow: an intuitive way to build workflow for GNES.

You can use *add()* then *build()* to customize your own workflow. For example:

```

from gnes.flow import Flow

f = (Flow(check_version=False, route_table=True)
     .add_preprocessor(yaml_path='BasePreprocessor'))

```

(continues on next page)

(continued from previous page)

```

    .add_encoder(yaml_path='BaseEncoder')
    .add_router(yaml_path='BaseRouter'))

with f.build(backend='thread') as flow:
    flow.index()
    ...

```

You can also use `add('Encoder', ...)` or `add(Service.Encoder, ...)` to add service to the flow. The generic `add()` provides a convenient way to build the flow.

As shown above, it is recommend to use flow in the context manner as showed above, as it manages all opened sockets/processes/threads automatically when exit from the context.

Note the different copy behaviors in `add()` and `build()`: `add()` always copy the flow by default, whereas `build()` modify the flow in place. You can change this behavior by specifying th argument `copy_flow=False`.

Create a new Flow object.

Parameters

- **with_frontend** (bool) – adding frontend service to the flow
- **is_trained** (bool) – indicating whether this flow is trained or not. if set to False then `index()` and `query()` can not be called before `train()`
- **kwargs** – keyword-value arguments that will be shared by all services

Frontend = 0

add (*service*, *name=None*, *recv_from=None*, *send_to=None*, *copy_flow=True*, ***kwargs*)

Add a service to the current flow object and return the new modified flow object. The attribute of the service can be later changed with `set()` or deleted with `remove()`

Note there are shortcut versions of this method. Recommend to use `add_encoder()`, `add_preprocessor()`, `add_router()`, `add_indexer()` whenever possible.

Parameters

- **service** (Union[Service, str]) – a ‘Service’ enum or string, possible choices: Encoder, Router, Preprocessor, Indexer, Frontend
- **name** (Optional[str]) – the name identifier of the service, can be used in ‘recv_from’, ‘send_to’, `set()` and `remove()`.
- **recv_from** (Union[str, Tuple[str], List[str], Service, None]) – the name of the service(s) that this service receives data from. One can also use ‘Service.Frontend’ to indicate the connection with the frontend.
- **send_to** (Union[str, Tuple[str], List[str], Service, None]) – the name of the service(s) that this service sends data to. One can also use ‘Service.Frontend’ to indicate the connection with the frontend.
- **copy_flow** (bool) – when set to true, then always copy the current flow and do the modification on top of it then return, otherwise, do in-line modification
- **kwargs** – other keyword-value arguments that the service CLI supports

Return type *Flow*

Returns a (new) flow object with modification

add_encoder (**args*, ***kwargs*)

Add an encoder to the current flow, a shortcut of `add(Service.Encoder)()`

Return type *Flow*

add_frontend (*args, **kwargs)

Add a frontend to the current flow, a shortcut of `add(Service.Frontend)()`. Usually you don't need to call this function explicitly, a flow object contains a frontend service by default. This function is useful when you build a flow without the frontend and want to customize the frontend later.

Return type *Flow*

add_indexer (*args, **kwargs)

Add an indexer to the current flow, a shortcut of `add(Service.Indexer)()`

Return type *Flow*

add_preprocessor (*args, **kwargs)

Add a preprocessor to the current flow, a shortcut of `add(Service.Preprocessor)()`

Return type *Flow*

add_router (*args, **kwargs)

Add a router to the current flow, a shortcut of `add(Service.Router)()`

Return type *Flow*

build (backend='process', copy_flow=False, *args, **kwargs)

Build the current flow and make it ready to use

Parameters

- **backend** (Optional[str]) – supported 'thread', 'process', 'swarm', 'k8s', 'shell', if None then only build graph only
- **copy_flow** (bool) – return the copy of the current flow

Return type *Flow*

Returns the current flow (by default)

close ()

Release the resources as model is destroyed

index (bytes_gen=None, **kwargs)

Do indexing on the current flow

Example,

```
with f.build(backend='thread') as flow:
    flow.index(txt_file='aa.txt')
    flow.index(image_zip_file='aa.zip', batch_size=64)
    flow.index(video_zip_file='aa.zip')
    ...
```

This will call the pre-built reader to read files into an iterator of bytes and feed to the flow.

One may also build a reader/generator on your own.

Example,

```
def my_reader():
    for _ in range(10):
        yield b'abcdefg' # each yield generates a document to index

with f.build(backend='thread') as flow:
    flow.index(bytes_gen=my_reader())
```

It will start a `CLIClient` and call `index()`.

Parameters

- **bytes_gen** (Optional[Iterator[bytes]]) – An iterator of bytes. If not given, then you have to specify it in *kwargs*.
- **kwargs** – accepts all keyword arguments of *gnes client CLI*

query (*bytes_gen=None, **kwargs*)

Do indexing on the current flow

It will start a `CLIClient` and call `query()`.

Example,

```
with f.build(backend='thread') as flow:
    flow.query(txt_file='aa.txt')
    flow.query(image_zip_file='aa.zip', batch_size=64)
    flow.query(video_zip_file='aa.zip')
    ...
```

This will call the pre-built reader to read files into an iterator of bytes and feed to the flow.

One may also build a reader/generator on your own.

Example,

```
def my_reader():
    for _ in range(10):
        yield b'abcdefg' # each yield generates a query for searching

with f.build(backend='thread') as flow:
    flow.query(bytes_gen=my_reader())
```

Parameters

- **bytes_gen** (Optional[Iterator[bytes]]) – An iterator of bytes. If not given, then you have to specify it in *kwargs*.
- **kwargs** – accepts all keyword arguments of *gnes client CLI*

remove (*name=None, copy_flow=True*)

Remove a service from the flow.

Parameters

- **name** (Optional[str]) – the name of the existing service
- **copy_flow** (bool) – when set to true, then always copy the current flow and do the modification on top of it then return, otherwise, do in-line modification

Return type *Flow*

Returns a (new) flow object with modification

set (*name, recv_from=None, send_to=None, copy_flow=True, clear_old_attr=False, as_last_service=False, **kwargs*)

Set the attribute of an existing service (added by `add()`) in the flow. For the attributes or kwargs that aren't given, they will remain unchanged as before.

Parameters

- **name** (str) – the name of the existing service

- **recv_from** (Union[str, Tuple[str], List[str], *Service*, None]) – the name of the service(s) that this service receives data from. One can also use ‘Service.Frontend’ to indicate the connection with the frontend.
- **send_to** (Union[str, Tuple[str], List[str], *Service*, None]) – the name of the service(s) that this service sends data to. One can also use ‘Service.Frontend’ to indicate the connection with the frontend.
- **copy_flow** (bool) – when set to true, then always copy the current flow and do the modification on top of it then return, otherwise, do in-line modification
- **clear_old_attr** (bool) – remove old attribute value before setting the new one
- **as_last_service** (bool) – whether setting the changed service as the last service in the graph
- **kwargs** – other keyword-value arguments that the service CLI supports

Return type *Flow*

Returns a (new) flow object with modification

set_last_service (*name*, *copy_flow=True*)

Set a service as the last service in the flow, useful when modifying the flow.

Parameters

- **name** (str) – the name of the existing service
- **copy_flow** (bool) – when set to true, then always copy the current flow and do the modification on top of it then return, otherwise, do in-line modification

Return type *Flow*

Returns a (new) flow object with modification

to_jpg (*path='flow.jpg'*, ***kwargs*)

Rendering the current flow as a jpg image, this will call *to_mermaid()* and it needs internet connection

Parameters

- **path** (str) – the file path of the image
- **kwargs** – keyword arguments of *to_mermaid()*

Return type None

Returns

to_k8s_yaml ()

Return type str

to_mermaid (*left_right=True*)

Output the mermaid graph for visualization

Parameters **left_right** (bool) – render the flow in left-to-right manner, otherwise top-down manner.

Return type str

Returns a mermaid-formatted string

to_python_code (*indent=4*)

Generate the python code of this flow

Parameters **indent** (int) – the number of whitespaces of indent

Return type `str`

Returns the generated python code

`to_shell_script()`

Return type `str`

`to_swarm_yaml(image='gnes/gnes:latest-alpine')`

Generate the docker swarm YAML compose file

Parameters `image` (`str`) – the default GNES docker image

Return type `str`

Returns the generated YAML compose file

`to_url(**kwargs)`

Rendering the current flow as a url points to a SVG, it needs internet connection

Parameters `kwargs` – keyword arguments of `to_mermaid()`

Return type `str`

Returns the url points to a SVG

`train(bytes_gen=None, **kwargs)`

Do training on the current flow

It will start a `CLIClient` and call `train()`.

Example,

```
with f.build(backend='thread') as flow:
    flow.train(txt_file='aa.txt')
    flow.train(image_zip_file='aa.zip', batch_size=64)
    flow.train(video_zip_file='aa.zip')
    ...
```

This will call the pre-built reader to read files into an iterator of bytes and feed to the flow.

One may also build a reader/generator on your own.

Example,

```
def my_reader():
    for _ in range(10):
        yield b'abcdfeg' # each yield generates a document for training

with f.build(backend='thread') as flow:
    flow.train(bytes_gen=my_reader())
```

Parameters

- `bytes_gen` (Optional[Iterator[bytes]]) – An iterator of bytes. If not given, then you have to specify it in `kwargs`.
- `kwargs` – accepts all keyword arguments of `gnes client CLI`

gnes.indexer package

Subpackages

gnes.indexer.chunk package

Subpackages

gnes.indexer.chunk.bindexer package

Submodules

gnes.indexer.chunk.bindexer.bindexer module

Module contents

gnes.indexer.chunk.hbindexer package

Submodules

gnes.indexer.chunk.hbindexer.hbindexer module

Module contents

Submodules

gnes.indexer.chunk.annoy module

```
class gnes.indexer.chunk.annoy.AnnoyIndexer (num_dim, data_path, metric='angular',
                                             n_trees=10, *args, **kwargs)
```

Bases: *gnes.indexer.base.BaseChunkIndexer*

Initialize an AnnoyIndexer

Parameters

- **num_dim** (*int*) – when set to -1, then num_dim is auto decided on first .add()
- **data_path** (*str*) – index data file managed by the annoy indexer
- **metric** (*str*) –
- **n_trees** (*int*) –
- **args** –
- **kwargs** –

```
add (keys, vectors, weights, *args, **kwargs)
```

adding new chunks and their vector representations

Parameters

- **keys** (*List[Tuple[int, Any]]*) – list of (doc_id, offset) tuple
- **vectors** (*ndarray*) – vector representations
- **weights** (*List[float]*) – weight of the chunks

```
post_init ()
```

Declare class attributes/members that can not be serialized in standard way

query (*keys, top_k, *args, **kwargs*)

Return type List[List[Tuple]]

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.indexer.chunk.faiss module

class gnes.indexer.chunk.faiss.**FaissIndexer** (*num_dim, index_key, data_path, *args, **kwargs*)

Bases: *gnes.indexer.base.BaseChunkIndexer*

Initialize an FaissIndexer

Parameters

- **num_dim** (int) – when set to -1, then num_dim is auto decided on first .add()
- **data_path** (str) – index data file managed by the faiss indexer

add (*keys, vectors, weights, *args, **kwargs*)

adding new chunks and their vector representations

Parameters

- **keys** (List[Tuple[int, Any]]) – list of (doc_id, offset) tuple
- **vectors** (ndarray) – vector representations
- **weights** (List[float]) – weight of the chunks

post_init ()

Declare class attributes/members that can not be serialized in standard way

query (*keys, top_k, *args, **kwargs*)

Return type List[List[Tuple]]

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.indexer.chunk.helper module

class gnes.indexer.chunk.helper.**DictKeyIndexer** (**args, **kwargs*)

Bases: *gnes.indexer.base.BaseChunkIndexerHelper*

add (*keys, weights, *args, **kwargs*)

adding new chunks and their vector representations

Parameters

- **keys** (List[Tuple[int, int]]) – list of (doc_id, offset) tuple
- **vectors** – vector representations
- **weights** (List[float]) – weight of the chunks

Return type int

query (*keys, *args, **kwargs*)

Return type List[Tuple[int, int, float]]

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.indexer.chunk.helper.**ListKeyIndexer** (*args, **kwargs)
Bases: *gnes.indexer.base.BaseChunkIndexerHelper*

add (keys, weights, *args, **kwargs)
adding new chunks and their vector representations

Parameters

- **keys** (List[Tuple[int, int]]) – list of (doc_id, offset) tuple
- **vectors** – vector representations
- **weights** (List[float]) – weight of the chunks

Return type int

query (keys, *args, **kwargs)

Return type List[Tuple[int, int, float]]

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.indexer.chunk.helper.**ListNumpyKeyIndexer** (*args, **kwargs)
Bases: *gnes.indexer.chunk.helper.ListKeyIndexer*

add (*args, **kwargs)
adding new chunks and their vector representations

Parameters

- **keys** – list of (doc_id, offset) tuple
- **vectors** – vector representations
- **weights** – weight of the chunks

Return type int

query (keys, *args, **kwargs)

Return type List[Tuple[int, int, float]]

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.indexer.chunk.helper.**NumpyKeyIndexer** (buffer_size=10000, col_size=3, *args, **kwargs)
Bases: *gnes.indexer.base.BaseChunkIndexerHelper*

add (keys, weights, *args, **kwargs)
adding new chunks and their vector representations

Parameters

- **keys** (List[Tuple[int, int]]) – list of (doc_id, offset) tuple
- **vectors** – vector representations
- **weights** (List[float]) – weight of the chunks

Return type int

capacity

query (keys, *args, **kwargs)

Return type List[Tuple[int, int, float]]

train (*args, **kwargs)
Train the model, need to be overridden

gnes.indexer.chunk.numpy module

class gnes.indexer.chunk.numpy.**NumpyIndexer** (is_binary=False, *args, **kwargs)

Bases: *gnes.indexer.base.BaseChunkIndexer*

An exhaustive search indexer using numpy The distance is computed as L1 distance normalized by the number of dimension

add (keys, vectors, weights, *args, **kwargs)
adding new chunks and their vector representations

Parameters

- **keys** (List[Tuple[int, Any]]) – list of (doc_id, offset) tuple
- **vectors** (ndarray) – vector representations
- **weights** (List[float]) – weight of the chunks

query (keys, top_k, *args, **kwargs)

Return type List[List[Tuple]]

train (*args, **kwargs)
Train the model, need to be overridden

Module contents

gnes.indexer.doc package

Submodules

gnes.indexer.doc.dict module

class gnes.indexer.doc.dict.**DictIndexer** (*args, **kwargs)

Bases: *gnes.indexer.base.BaseDocIndexer*

add (keys, docs, *args, **kwargs)
adding new docs and their protobuf representation

Parameters

- **keys** (List[int]) – list of doc_id
- **docs** (List[Document]) – list of protobuf Document objects

query (keys, *args, **kwargs)

Return type List[Document]

train (*args, **kwargs)
Train the model, need to be overridden

gnes.indexer.doc.filesys module

class `gnes.indexer.doc.filesys.DirectoryIndexer` (*data_path*, *keep_na_doc=True*,
file_suffix='gif', *args, **kwargs)

Bases: `gnes.indexer.base.BaseDocIndexer`

add (*keys*, *docs*, *args, **kwargs)

write GIFs of each document into disk folder structure: /data_path/doc_id/0.gif, 1.gif...

Parameters

- **keys** (`List[int]`) – list of doc id
- **docs** (`List[Document]`) – list of docs

query (*keys*, *args, **kwargs)

Find the doc according to the keys

Parameters **keys** (`List[int]`) – list of doc id

Return type `List[Document]`

Returns list of documents whose chunks field contain all the GIFs of this doc(one GIF per chunk)

train (*args, **kwargs)

Train the model, need to be overridden

gnes.indexer.doc.leveldb module

class `gnes.indexer.doc.leveldb.AsyncLVDBIndexer` (*data_path*, *keep_na_doc=True*,
drop_raw_bytes=False,
drop_chunk_blob=False, *args,
**kwargs)

Bases: `gnes.indexer.doc.leveldb.LVDBIndexer`

add (*keys*, *docs*, *args, **kwargs)

adding new docs and their protobuf representation

Parameters

- **keys** (`List[int]`) – list of doc_id
- **docs** (`List[Document]`) – list of protobuf Document objects

close ()

Release the resources as model is destroyed

post_init ()

Declare class attributes/members that can not be serialized in standard way

query (*args, **kwargs)

Return type `List[Any]`

train (*args, **kwargs)

Train the model, need to be overridden

class `gnes.indexer.doc.leveldb.LVDBIndexer` (*data_path*, *keep_na_doc=True*,
drop_raw_bytes=False,
drop_chunk_blob=False, *args, **kwargs)

Bases: `gnes.indexer.base.BaseDocIndexer`

add (*keys*, *docs*, **args*, ***kwargs*)
adding new docs and their protobuf representation

Parameters

- **keys** (`List[int]`) – list of `doc_id`
- **docs** (`List[Document]`) – list of protobuf `Document` objects

close ()
Release the resources as model is destroyed

post_init ()
Declare class attributes/members that can not be serialized in standard way

query (*keys*, **args*, ***kwargs*)
Return type `List[Document]`

train (**args*, ***kwargs*)
Train the model, need to be overridden

gnes.indexer.doc.rocksdb module

class `gnes.indexer.doc.rocksdb.RocksDBIndexer` (*data_path*, *drop_raw_data=False*,
drop_chunk_blob=False,
read_only=False, **args*, ***kwargs*)

Bases: `gnes.indexer.base.BaseDocIndexer`

add (*keys*, *docs*, **args*, ***kwargs*)
adding new docs and their protobuf representation

Parameters

- **keys** (`List[int]`) – list of `doc_id`
- **docs** (`List[Document]`) – list of protobuf `Document` objects

close ()
Release the resources as model is destroyed

post_init ()
Declare class attributes/members that can not be serialized in standard way

query (*keys*, **args*, ***kwargs*)
Return type `List[Document]`

scan (*reversed_scan=False*)

train (**args*, ***kwargs*)
Train the model, need to be overridden

Module contents

Submodules

gnes.indexer.base module

class `gnes.indexer.base.BaseChunkIndexer` (*helper_indexer=None*, **args*, ***kwargs*)
Bases: `gnes.indexer.base.BaseIndexer`

Storing chunks and their vector representations

add (*keys*, *vectors*, *weights*, **args*, ***kwargs*)
adding new chunks and their vector representations

Parameters

- **keys** (List[Tuple[int, int]]) – list of (doc_id, offset) tuple
- **vectors** (ndarray) – vector representations
- **weights** (List[float]) – weight of the chunks

num_chunks

num_chunks_in_doc (*doc_id*)

num_docs

query (*keys*, *top_k*, **args*, ***kwargs*)

Return type List[List[Tuple]]

query_and_score (*q_chunks*, *top_k*, **args*, ***kwargs*)

Return type List[ScoredResult]

train (**args*, ***kwargs*)

Train the model, need to be overridden

static update_counter (*func*)

static update_helper_indexer (*func*)

class `gnex.indexer.base.BaseChunkIndexerHelper` (*helper_indexer=None*, **args*,
***kwargs*)

Bases: `gnex.indexer.base.BaseChunkIndexer`

A helper class for storing chunk info, doc mapping, weights. This is especially useful when ChunkIndexer can not store these information by itself

add (*keys*, *weights*, **args*, ***kwargs*)
adding new chunks and their vector representations

Parameters

- **keys** (List[Tuple[int, int]]) – list of (doc_id, offset) tuple
- **vectors** – vector representations
- **weights** (List[float]) – weight of the chunks

Return type int

query (*keys*, **args*, ***kwargs*)

Return type List[Tuple[int, int, float]]

train (**args*, ***kwargs*)

Train the model, need to be overridden

class `gnex.indexer.base.BaseDocIndexer` (*normalize_fn=None*, *score_fn=None*,
is_big_score_similar=False, **args*, ***kwargs*)

Bases: `gnex.indexer.base.BaseIndexer`

Storing documents and contents

Base indexer, a valid indexer must implement `add()` and `query()` methods

add (*keys*, *docs*, **args*, ***kwargs*)
adding new docs and their protobuf representation

Parameters

- **keys** (`List[int]`) – list of `doc_id`
- **docs** (`List[Document]`) – list of protobuf `Document` objects

query (*keys*, **args*, ***kwargs*)

Return type `List[Document]`

query_and_score (*docs*, **args*, ***kwargs*)

Return type `List[ScoredResult]`

train (**args*, ***kwargs*)

Train the model, need to be overridden

static update_counter (*func*)

class `gn.es.indexer.base.BaseIndexer` (*normalize_fn=None*, *score_fn=None*,
is_big_score_similar=False, **args*, ***kwargs*)

Bases: `gn.es.base.TrainableBase`

Base indexer, a valid indexer must implement `add()` and `query()` methods

add (*keys*, *docs*, *weights*, **args*, ***kwargs*)

num_chunks

num_docs

query (*keys*, **args*, ***kwargs*)

Return type `List[Any]`

query_and_score (*q_chunks*, *top_k*)

Return type `List[ScoredResult]`

train (**args*, ***kwargs*)

Train the model, need to be overridden

class `gn.es.indexer.base.JointIndexer` (**args*, ***kwargs*)

Bases: `gn.es.base.CompositionalTrainableBase`

add (*keys*, *docs*, **args*, ***kwargs*)

Return type `None`

components

query (*keys*, *top_k*, **args*, ***kwargs*)

Return type `List[List[Tuple]]`

train (**args*, ***kwargs*)

Train the model, need to be overridden

Module contents

gnes.preprocessor package

Subpackages

gnes.preprocessor.audio package

Subpackages

gnes.preprocessor.audio.vggish_example_helper package

Submodules

gnes.preprocessor.audio.vggish_example_helper.mel_features module

`gnes.preprocessor.audio.vggish_example_helper.mel_features.frame` (*data*, *window_length*, *hop_length*)

Convert array into a sequence of successive possibly overlapping frames.

An n-dimensional array of shape (num_samples, ...) is converted into an (n+1)-D array of shape (num_frames, window_length, ...), where each frame starts hop_length points after the preceding one.

This is accomplished using stride_tricks, so the original data is not copied. However, there is no zero-padding, so any incomplete frames at the end are not included.

Args: data: np.array of dimension N >= 1. window_length: Number of samples in each frame. hop_length: Advance (in samples) between each window.

Returns: (N+1)-D np.array with as many rows as there are complete frames that can be extracted.

`gnes.preprocessor.audio.vggish_example_helper.mel_features.hertz_to_mel` (*frequencies_hertz*)

Convert frequencies to mel scale using HTK formula.

Args: frequencies_hertz: Scalar or np.array of frequencies in hertz.

Returns: Object of same size as frequencies_hertz containing corresponding values on the mel scale.

`gnes.preprocessor.audio.vggish_example_helper.mel_features.log_mel_spectrogram` (*data*, *audio_sample_rate*, *log_offset=0.0*, *window_length_secs*, *hop_length_secs*, ***kwargs*)

Convert waveform to a log magnitude mel-frequency spectrogram.

Args: data: 1D np.array of waveform data. audio_sample_rate: The sampling rate of data. log_offset: Add this to values when taking log to avoid -Infs. window_length_secs: Duration of each window to analyze. hop_length_secs: Advance between successive analysis windows. ****kwargs:** Additional arguments to pass to spectrogram_to_mel_matrix.

Returns: 2D np.array of (num_frames, num_mel_bins) consisting of log mel filterbank magnitudes for successive frames.

`gn.es.preprocessor.audio.vggish_example_helper.mel_features.periodic_hann(window_length)`
Calculate a “periodic” Hann window.

The classic Hann window is defined as a raised cosine that starts and ends on zero, and where every value appears twice, except the middle point for an odd-length window. Matlab calls this a “symmetric” window and `np.hanning()` returns it. However, for Fourier analysis, this actually represents just over one cycle of a period $N-1$ cosine, and thus is not compactly expressed on a length- N Fourier basis. Instead, it’s better to use a raised cosine that ends just before the final zero value - i.e. a complete cycle of a period- N cosine. Matlab calls this a “periodic” window. This routine calculates it.

Args: `window_length`: The number of points in the returned window.

Returns: A 1D `np.array` containing the periodic hann window.

`gn.es.preprocessor.audio.vggish_example_helper.mel_features.spectrogram_to_mel_matrix(num_mel_bins, num_spectrogram_bins, audio_sample_rate, lower_edge_hertz, upper_edge_hertz)`

Return a matrix that can post-multiply spectrogram rows to make mel.

Returns a `np.array` matrix `A` that can be used to post-multiply a matrix `S` of spectrogram values (STFT magnitudes) arranged as frames x bins to generate a “mel spectrogram” `M` of frames x `num_mel_bins`. $M = SA$.

The classic HTK algorithm exploits the complementarity of adjacent mel bands to multiply each FFT bin by only one mel weight, then add it, with positive and negative signs, to the two adjacent mel bands to which that bin contributes. Here, by expressing this operation as a matrix multiply, we go from `num_fft` multiplies per frame (plus around $2 * \text{num_fft}$ adds) to around num_fft^2 multiplies and adds. However, because these are all presumably accomplished in a single call to `np.dot()`, it’s not clear which approach is faster in Python. The matrix multiplication has the attraction of being more general and flexible, and much easier to read.

Args:

`num_mel_bins`: How many bands in the resulting mel spectrum. This is the number of columns in the output matrix.

`num_spectrogram_bins`: How many bins there are in the source spectrogram data, which is understood to be $\text{fft_size}/2 + 1$, i.e. the spectrogram only contains the nonredundant FFT bins.

`audio_sample_rate`: Samples per second of the audio at the input to the spectrogram. We need this to figure out the actual frequencies for each spectrogram bin, which dictates how they are mapped into mel.

`lower_edge_hertz`: Lower bound on the frequencies to be included in the mel spectrum. This corresponds to the lower edge of the lowest triangular band.

`upper_edge_hertz`: The desired top edge of the highest frequency band.

Returns: An `np.array` with shape `(num_spectrogram_bins, num_mel_bins)`.

Raises: `ValueError`: if frequency edges are incorrectly ordered or out of range.

`gn.es.preprocessor.audio.vggish_example_helper.mel_features.stft_magnitude(signal, fft_length, hop_length=None, window_length=None)`
Calculate the short-time Fourier transform magnitude.

Args: `signal`: 1D `np.array` of the input time-domain signal. `fft_length`: Size of the FFT to apply. `hop_length`: Advance (in samples) between each frame passed to FFT. `window_length`: Length of each block of samples to pass to FFT.

Returns: 2D `np.array` where each row contains the magnitudes of the `fft_length/2+1` unique values of the FFT for the corresponding frame of input samples.

Module contents

Submodules

`gnes.preprocessor.audio.audio_vanilla` module

`gnes.preprocessor.audio.vggish_example` module

```
class gnes.preprocessor.audio.vggish_example.VggishPreprocessor (num_frames=96,
                                                                num_bands=64,
                                                                sam-
                                                                ple_rate=16000,
                                                                log_offset=0.01,
                                                                exam-
                                                                ple_window_seconds=0.96,
                                                                exam-
                                                                ple_hop_seconds=0.96,
                                                                stft_window_length_seconds=0.025,
                                                                stft_hop_length_seconds=0.01,
                                                                mel_min_hz=125,
                                                                mel_max_hz=7500,
                                                                *args,
                                                                **kwargs)
```

Bases: `gnes.preprocessor.base.BaseAudioPreprocessor`

`apply` (*doc*)

Return type None

`train` (**args, **kwargs*)

Train the model, need to be overridden

`waveform_to_examples` (*data, sample_rate*)

Converts audio waveform into an array of examples for VGGish.

Args:

data: `np.array` of either one dimension (mono) or two dimensions (multi-channel, with the outer dimension representing channels). Each sample is generally expected to lie in the range [-1.0, +1.0], although this is not required.

`sample_rate`: Sample rate of data.

Returns: 3-D `np.array` of shape [`num_examples`, `num_frames`, `num_bands`] which represents a sequence of examples, each of which contains a patch of log mel spectrogram, covering `num_frames` frames of audio and `num_bands` mel frequency bands, where the frame length is `vggish_params.STFT_HOP_LENGTH_SECONDS`.

Module contents

gnes.preprocessor.image package

Submodules

gnes.preprocessor.image.resize module

```
class gnes.preprocessor.image.resize.ResizeChunkPreprocessor (target_width=224,  
target_height=224,  
*args, **kwargs)
```

Bases: *gnes.preprocessor.image.resize.SizedPreprocessor*

apply (*doc*)

Return type None

train (**args, **kwargs*)

Train the model, need to be overridden

```
class gnes.preprocessor.image.resize.SizedPreprocessor (target_width=224, tar-  
get_height=224, *args,  
**kwargs)
```

Bases: *gnes.preprocessor.base.BaseImagePreprocessor*

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.preprocessor.image.segmentation module

```
class gnes.preprocessor.image.segmentation.SegmentPreprocessor (model_name,  
model_dir,  
*args,  
**kwargs)
```

Bases: *gnes.preprocessor.image.resize.SizedPreprocessor*

apply (*doc*)

post_init ()

Declare class attributes/members that can not be serialized in standard way

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.preprocessor.image.sliding_window module

```
class gnes.preprocessor.image.sliding_window.VanillaSlidingPreprocessor (window_size=64,  
stride_height=64,  
stride_wide=64,  
*args,  
**kwargs)
```

Bases: *gnes.preprocessor.image.sliding_window._SlidingPreprocessor*

train (**args, **kwargs*)

Train the model, need to be overridden

```
class gnes.preprocessor.image.sliding_window.WeightedSlidingPreprocessor (window_size=64,  
stride_height=64,  
stride_wide=64,  
*args,  
**kwargs)
```

Bases: `gnes.preprocessor.image.sliding_window._SlidingPreprocessor`

```
train (*args, **kwargs)
```

Train the model, need to be overridden

Module contents

`gnes.preprocessor.io_utils` package

Submodules

`gnes.preprocessor.io_utils.audio` module

`gnes.preprocessor.io_utils.ffmpeg` module

```
gnes.preprocessor.io_utils.ffmpeg.compile_args (input_fn='pipe:', output_fn='pipe:',  
video_filters=[], audio_filters=[], in-  
put_options={}, output_options={},  
overwrite_output=True)
```

Wrapper for various `FFmpeg` related applications (`ffmpeg`, `ffprobe`).

```
gnes.preprocessor.io_utils.ffmpeg.extract_frame_size (ffmpeg_parse_info)
```

```
gnes.preprocessor.io_utils.ffmpeg.get_media_meta (input_fn='pipe:', input_data=None,  
input_options={})
```

```
gnes.preprocessor.io_utils.ffmpeg.parse_media_details (infos)
```

```
gnes.preprocessor.io_utils.ffmpeg.probe (input_fn)
```

`gnes.preprocessor.io_utils.gif` module

```
gnes.preprocessor.io_utils.gif.capture_frames (input_fn='pipe:', input_data=None,  
fps=None, pix_fmt='rgb24', vframes=-1)
```

Return type `ndarray`

```
gnes.preprocessor.io_utils.gif.encode_video (images, frame_rate, pix_fmt='rgb24')
```

`gnes.preprocessor.io_utils.helper` module

```
gnes.preprocessor.io_utils.helper.kwargs_to_cmd_args (kwargs)
```

```
gnes.preprocessor.io_utils.helper.run_command (cmd_args, input=None, pipe_stdin=True,  
pipe_stdout=False, pipe_stderr=False,  
quiet=False)
```

```
gnes.preprocessor.io_utils.helper.run_command_async(cmd_args, pipe_stdin=True,
                                                    pipe_stdout=False,
                                                    pipe_stderr=False,
                                                    quiet=False)
```

```
gnes.preprocessor.io_utils.helper.wait(process)
```

gnes.preprocessor.io_utils.video module

```
gnes.preprocessor.io_utils.video.capture_frames(input_fn='pipe:', input_data=None,
                                                  pix_fmt='rgb24', fps=-1, scale=None,
                                                  start_time=None, end_time=None,
                                                  vframes=-1, **kwargs)
```

Return type List[ndarray]

```
gnes.preprocessor.io_utils.video.encode_video(images, pix_fmt='rgb24', frame_rate=15,
                                                output_fn='pipe:', vcodec='libx264', for-
                                                mat='mp4', **kwargs)
```

```
gnes.preprocessor.io_utils.video.scale_video(input_fn='pipe:', output_fn='pipe:',
                                                input_data=None, start_time=None,
                                                end_time=None, scale=None,
                                                frame_rate=15, crf=16, vcodec='libx264',
                                                format='mp4', pix_fmt='yuv420p',
                                                **kwargs)
```

gnes.preprocessor.io_utils.webp module

```
gnes.preprocessor.io_utils.webp.encode_video(images, frame_rate, pix_fmt='rgb24')
```

Module contents

gnes.preprocessor.text package

Submodules

gnes.preprocessor.text.split module

```
class gnes.preprocessor.text.split.SentSplitPreprocessor(min_sent_len=1,
                                                         max_sent_len=256,
                                                         delimitator='!?',
                                                         is_json=False, *args,
                                                         **kwargs)
```

Bases: *gnes.preprocessor.base.BaseTextPreprocessor*

```
apply(doc)
```

Return type None

```
train(*args, **kwargs)
```

Train the model, need to be overridden

Module contents

gnes.preprocessor.video package

Submodules

gnes.preprocessor.video.ffmpeg module

```
class gnes.preprocessor.video.ffmpeg.FFmpegPreprocessor (frame_size='192:168',
                                                    frame_rate=10,
                                                    frame_num=-1,      du-
                                                    plicate_rm=True,
                                                    use_phash_weight=False,
                                                    phash_thresh=5,      *args,
                                                    **kwargs)
```

Bases: *gnes.preprocessor.base.BaseVideoPreprocessor*

apply (*doc*)

Return type None

duplicate_rm_hash (*images*)

Return type List[ndarray]

static pic_weight (*images*)

Return type List[float]

train (**args, **kwargs*)

Train the model, need to be overridden

```
class gnes.preprocessor.video.ffmpeg.FFmpegVideoSegmentor (frame_size='192:168',
                                                            frame_rate=10,
                                                            frame_num=-1,      seg-
                                                            ment_method='cut_by_frame',
                                                            segment_interval=-1,
                                                            segment_num=3,
                                                            max_frames_per_doc=-
                                                            1,
                                                            use_image_input=False,
                                                            splitter='__split__',
                                                            *args, **kwargs)
```

Bases: *gnes.preprocessor.base.BaseVideoPreprocessor*

apply (*doc*)

Return type None

train (**args, **kwargs*)

Train the model, need to be overridden

```
class gnes.preprocessor.video.ffmpeg.GifChunkPreprocessor (uniform_doc_weight=True,
                                                            *args, **kwargs)
```

Bases: *gnes.preprocessor.base.RawChunkPreprocessor, gnes.preprocessor.base.BaseVideoPreprocessor*

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.preprocessor.video.frame_select module

```
class gnes.preprocessor.video.frame_select.FrameSelectPreprocessor (sframes=1,  
*args,  
**kwargs)
```

Bases: *gnes.preprocessor.base.BaseVideoPreprocessor*

apply (*doc*)

Return type None

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.preprocessor.video.shot_detector module

```
class gnes.preprocessor.video.shot_detector.ShotDetectorPreprocessor (descriptor='block_hsv_histogram'  
dis-  
tance_metric='bhattacharya',  
de-  
tect_method='threshold',  
frame_size=None,  
frame_rate=10,  
vframes=-  
1,  
sframes=-  
1,  
drop_raw_data=False,  
*args,  
**kwargs)
```

Bases: *gnes.preprocessor.base.BaseVideoPreprocessor*

apply (*doc*)

Return type None

detect_shots (*frames*)

Return type List[List[ndarray]]

store_args_kwargs = True

train (**args, **kwargs*)

Train the model, need to be overridden

gnes.preprocessor.video.video_decoder module

```
class gnes.preprocessor.video.video_decoder.VideoDecoderPreprocessor (frame_rate=10,  
frame_size=None,  
vframes=-  
1,  
drop_raw_data=False,  
chunk_splitter=None,  
*args,  
**kwargs)
```

Bases: *gnes.preprocessor.base.BaseVideoPreprocessor*

apply (*doc*)

Return type None

store_args_kwargs = True

train (**args*, ***kwargs*)

Train the model, need to be overridden

gnes.preprocessor.video.video_encoder module

```
class gnes.preprocessor.video.video_encoder.VideoEncoderPreprocessor (frame_rate=10,  
pix_fmt='rgb24',  
video_format='mp4',  
*args,  
**kwargs)
```

Bases: *gnes.preprocessor.base.BaseVideoPreprocessor*

apply (*doc*)

Return type None

train (**args*, ***kwargs*)

Train the model, need to be overridden

Module contents

Submodules

gnes.preprocessor.base module

```
class gnes.preprocessor.base.BaseAudioPreprocessor (uniform_doc_weight=True, *args,  
**kwargs)
```

Bases: *gnes.preprocessor.base.BasePreprocessor*

doc_type = 4

train (**args*, ***kwargs*)

Train the model, need to be overridden

```
class gnes.preprocessor.base.BaseImagePreprocessor (uniform_doc_weight=True, *args,  
**kwargs)
```

Bases: *gnes.preprocessor.base.BasePreprocessor*

doc_type = 2

train (**args*, ***kwargs*)

Train the model, need to be overridden

```
class gnes.preprocessor.base.BasePreprocessor (uniform_doc_weight=True, *args,  
**kwargs)
```

Bases: *gnes.base.TrainableBase*

apply (*doc*)

Return type None

doc_type = 0

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.preprocessor.base.**BaseTextPreprocessor** (uniform_doc_weight=True, *args, **kwargs)

Bases: *gnes.preprocessor.base.BasePreprocessor*

doc_type = 1

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.preprocessor.base.**BaseVideoPreprocessor** (uniform_doc_weight=True, *args, **kwargs)

Bases: *gnes.preprocessor.base.BasePreprocessor*

doc_type = 3

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.preprocessor.base.**PipelinePreprocessor** (*args, **kwargs)

Bases: *gnes.base.CompositionalTrainableBase*

apply (doc)

Return type None

train (data, *args, **kwargs)
Train the model, need to be overridden

class gnes.preprocessor.base.**RawChunkPreprocessor** (uniform_doc_weight=True, *args, **kwargs)

Bases: *gnes.preprocessor.base.BasePreprocessor*

apply (doc)

Return type None

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.preprocessor.base.**UnaryPreprocessor** (doc_type, *args, **kwargs)

Bases: *gnes.preprocessor.base.BasePreprocessor*

apply (doc)

is_trained = True

raw_to_chunk (chunk, raw_bytes)

train (*args, **kwargs)
Train the model, need to be overridden

gnes.preprocessor.helper module

gnes.preprocessor.helper.**block_descriptor** (image, descriptor_fn, num_blocks=3)

Return type ndarray

gnes.preprocessor.helper.**canny_edge** (image, **kwargs)

Return type ndarray

`gnes.preprocessor.helper.check_motion` (*prev_dists*, *cur_dist*, *motion_threshold=0.75*)
Returns a boolean value to decide if the peak is due to a motion

`gnes.preprocessor.helper.compare_descriptor` (*descriptor1*, *descriptor2*, *metric='chisqr'*)
Return type float

`gnes.preprocessor.helper.compare_ecr` (*descriptors*, ***kwargs*)
Return type List[float]

`gnes.preprocessor.helper.compute_descriptor` (*image*, *method='rgb_histogram'*, ***kwargs*)
Return type <built-in function array>

`gnes.preprocessor.helper.detect_peak_boundary` (*distances*, *method='kmeans'*, ***kwargs*)
Return type List[int]

`gnes.preprocessor.helper.get_all_subarea` (*img*)

`gnes.preprocessor.helper.get_audio` (*buffer_data*, *sample_rate*, *interval*, *duration*)
Return type List[ndarray]

`gnes.preprocessor.helper.get_gif` (*images*, *fps=10*)

`gnes.preprocessor.helper.get_video_length` (*video_path*)

`gnes.preprocessor.helper.get_video_length_from_raw` (*buffer_data*)

`gnes.preprocessor.helper.hsv_histogram` (*image*)
Return type ndarray

`gnes.preprocessor.helper.kmeans_algo` (*distances*, ***kwargs*)
Return type List[int]

`gnes.preprocessor.helper.motion_algo` (*distances*, ***kwargs*)
Return type List[int]

`gnes.preprocessor.helper.phash_descriptor` (*image*)

`gnes.preprocessor.helper.pyramid_descriptor` (*image*, *descriptor_fn*, *max_level=2*)
Return type ndarray

`gnes.preprocessor.helper.rgb_histogram` (*image*)
Return type ndarray

`gnes.preprocessor.helper.split_mp4_random` (*video_path*, *avg_length*, *max_clip_second=10*)

`gnes.preprocessor.helper.split_video_frames` (*buffer_data*, *splitter='__split__'*)

`gnes.preprocessor.helper.thre_algo` (*distances*, ***kwargs*)
Return type List[int]

`gnes.preprocessor.helper.torch_transform` (*img*)

Module contents

gnes.proto package

Submodules

gnes.proto.gnes_pb2 module

gnes.proto.gnes_pb2_grpc module

class gnes.proto.gnes_pb2_grpc.**GnesRPCServicer**

Bases: object

Call (*request, context*)

Index (*request, context*)

Query (*request, context*)

StreamCall (*request_iterator, context*)

Train (*request, context*)

option (rpc_core.method_no_deadline) = true; option (rpc_core.service_default_deadline_ms) = 5000;

class gnes.proto.gnes_pb2_grpc.**GnesRPCStub** (*channel*)

Bases: object

Constructor.

Args: channel: A grpc.Channel.

gnes.proto.gnes_pb2_grpc.**add_GnesRPCServicer_to_server** (*servicer, server*)

Module contents

class gnes.proto.**RequestGenerator**

Bases: object

static index (*data, batch_size=0, doc_type=1, doc_id_start=0, request_id_start=0, random_doc_id=False, *args, **kwargs*)

static query (*query, top_k, doc_type=1, request_id_start=0, *args, **kwargs*)

static train (*data, batch_size=0, doc_type=1, doc_id_start=0, request_id_start=0, random_doc_id=False, *args, **kwargs*)

gnes.proto.**send_message** (*sock, msg, timeout=-1, squeeze_pb=False, **kwargs*)

Return type None

gnes.proto.**recv_message** (*sock, timeout=-1, check_version=False, **kwargs*)

Return type Optional[Message]

gnes.proto.**blob2array** (*blob*)

Convert a blob proto to an array.

Return type ndarray

gnes.proto.**array2blob** (*x*)

Converts a N-dimensional array to blob proto.

Return type `NdArray`

`gnes.proto.add_route` (*evlp, name, identity*)

`gnes.proto.add_version` (*evlp*)

gnes.router package

Submodules

gnes.router.base module

class `gnes.router.base.BaseEmbedReduceRouter` (**args, **kwargs*)

Bases: `gnes.router.base.BaseReduceRouter`

apply (*msg, accum_msgs, *args, **kwargs*)

reduce embeddings from encoders (means, concat ...)

Parameters

- **msg** (`Message`) – the current message
- **accum_msgs** (`List[Message]`) – accumulated messages

Return type `None`

reduce_embedding (*accum_msgs, msg_type, chunk_idx, doc_idx*)

train (**args, **kwargs*)

Train the model, need to be overridden

class `gnes.router.base.BaseMapRouter` (**args, **kwargs*)

Bases: `gnes.router.base.BaseRouter`

apply (*msg, *args, **kwargs*)

Modify the incoming message

Parameters **msg** (`Message`) – incoming message

Return type `Generator[+T_co, -T_contra, +V_co]`

train (**args, **kwargs*)

Train the model, need to be overridden

class `gnes.router.base.BaseReduceRouter` (**args, **kwargs*)

Bases: `gnes.router.base.BaseRouter`

apply (*msg, accum_msgs, *args, **kwargs*)

Modify the current message based on accumulated messages

Parameters

- **msg** (`Message`) – the current message
- **accum_msgs** (`List[Message]`) – accumulated messages

Return type `None`

train (**args, **kwargs*)

Train the model, need to be overridden

```
class gnes.router.base.BaseRouter (*args, **kwargs)
```

Bases: *gnes.base.TrainableBase*

Base class for the router. Inherit from this class to create a new router.

Router forwards messages between services. Essentially, it receives a 'gnes_pb2.Message' and call *apply()* method on it.

```
apply (msg, *args, **kwargs)
```

Modify the incoming message

Parameters *msg* (Message) – incoming message

```
train (*args, **kwargs)
```

Train the model, need to be overridden

```
class gnes.router.base.BaseTopkReduceRouter (reduce_op='sum', *args, **kwargs)
```

Bases: *gnes.router.base.BaseReduceRouter*

```
apply (msg, accum_msgs, *args, **kwargs)
```

Modify the current message based on accumulated messages

Parameters

- **msg** (Message) – the current message
- **accum_msgs** (List[Message]) – accumulated messages

```
get_key (x)
```

Return type str

```
post_init ()
```

Declare class attributes/members that can not be serialized in standard way

```
set_key (x, k)
```

Return type None

```
train (*args, **kwargs)
```

Train the model, need to be overridden

```
class gnes.router.base.PipelineRouter (*args, **kwargs)
```

Bases: *gnes.base.CompositionalTrainableBase*

```
apply (*args, **kwargs)
```

Return type None

```
train (*args, **kwargs)
```

Train the model, need to be overridden

gnes.router.map module

```
class gnes.router.map.BlockRouter (sleep_sec=5, *args, **kwargs)
```

Bases: *gnes.router.base.BaseMapRouter*

Wait for 'sleep_sec' seconds and forward messages, useful for benchmark

```
apply (msg, *args, **kwargs)
```

Modify the incoming message

Parameters *msg* (Message) – incoming message

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.router.map.DocBatchRouter (*args, **kwargs)
Bases: *gnes.router.base.BaseMapRouter*

apply (msg, *args, **kwargs)
Modify the incoming message

Parameters msg (Message) – incoming message

Return type Generator[+T_co, -T_contra, +V_co]

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.router.map.PublishRouter (num_part, *args, **kwargs)
Bases: *gnes.router.base.BaseMapRouter*

Copy a message ‘num_part’ time and forward it, useful for PUB-SUB sockets. ‘num_part’ is an indicator for downstream sync-barrier, e.g. a ReduceRouter

apply (msg, *args, **kwargs)
Modify the incoming message

Parameters msg (Message) – incoming message

Return type Generator[+T_co, -T_contra, +V_co]

train (*args, **kwargs)
Train the model, need to be overridden

gnes.router.reduce module

class gnes.router.reduce.AvgEmbedRouter (*args, **kwargs)
Bases: *gnes.router.base.BaseEmbedReduceRouter*

Gather all embeddings from multiple encoders and do average on a specific axis. In default, average will happen on the first axis. chunk_idx, doc_idx denote index in for loop used in BaseEmbedReduceRouter

reduce_embedding (accum_msgs, msg_type, chunk_idx, doc_idx)

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.router.reduce.Chunk2DocTopkReducer (reduce_op='sum', *args, **kwargs)
Bases: *gnes.router.base.BaseTopkReduceRouter*

Gather all chunks by their doc_id, result in a topk doc list. This is almost always useful, as the final result should be group by doc_id not chunk

get_key (x)

Return type str

set_key (x, k)

train (*args, **kwargs)
Train the model, need to be overridden

class gnes.router.reduce.ChunkTopkReducer (reduce_op='sum', *args, **kwargs)
Bases: *gnes.router.base.BaseTopkReduceRouter*

Gather all chunks by their chunk_id from all shards, aka doc_id-offset, result in a topk chunk list

get_key (*x*)

Return type `str`

set_key (*x*, *k*)

train (**args*, ***kwargs*)

Train the model, need to be overridden

class `gn.es.router.reduce.ConcatEmbedRouter` (**args*, ***kwargs*)

Bases: `gn.es.router.base.BaseEmbedReduceRouter`

Gather all embeddings from multiple encoders and concat them on a specific axis. In default, concat will happen on the last axis. `chunk_idx`, `doc_idx` denote index in for loop used in `BaseEmbedReduceRouter`

reduce_embedding (*accum_msgs*, *msg_type*, *chunk_idx*, *doc_idx*)

train (**args*, ***kwargs*)

Train the model, need to be overridden

class `gn.es.router.reduce.DocFillReducer` (**args*, ***kwargs*)

Bases: `gn.es.router.base.BaseReduceRouter`

Gather all documents raw content from multiple shards. This is only useful when you have - multiple doc-indexer and docs are spreaded over multiple shards. - require full-doc retrieval with the original content, not just an doc id Ideally, only each doc can only belong to one shard.

apply (*msg*, *accum_msgs*, **args*, ***kwargs*)

Modify the current message based on accumulated messages

Parameters

- **msg** (`gn.es.pb2.Message`) – the current message
- **accum_msgs** (`List[Message]`) – accumulated messages

train (**args*, ***kwargs*)

Train the model, need to be overridden

class `gn.es.router.reduce.DocTopkReducer` (*reduce_op='sum'*, **args*, ***kwargs*)

Bases: `gn.es.router.base.BaseTopkReduceRouter`

Gather all docs by their `doc_id`, result in a topk doc list

get_key (*x*)

Return type `str`

set_key (*x*, *k*)

train (**args*, ***kwargs*)

Train the model, need to be overridden

Module contents

`gn.es.score_fn` package

Submodules

gnes.score_fn.base module

```

class gnes.score_fn.base.BaseScoreFn (context=None, *args, **kwargs)
    Bases: gnes.base.TrainableBase

    Base score function. A score function must implement __call__ method

    new_score (*, operands=(), **kwargs)

    train (*args, **kwargs)
        Train the model, need to be overridden

    warn_unnamed = False

class gnes.score_fn.base.CombinedScoreFn (score_mode='multiply', *args, **kwargs)
    Bases: gnes.score_fn.base.BaseScoreFn

    Combine multiple scores into one score, defaults to 'multiply'

        Parameters score_mode (str) – specifies how the computed scores are combined

    post_init ()
        Declare class attributes/members that can not be serialized in standard way

    supported_ops

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.score_fn.base.ModifierScoreFn (modifier='none', factor=1.0, factor_name='GivenConstant', *args, **kwargs)
    Bases: gnes.score_fn.base.BaseScoreFn

    Modifier to apply to the value score = modifier(factor * value)

    post_init ()
        Declare class attributes/members that can not be serialized in standard way

    supported_ops

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.score_fn.base.ScoreOps
    Bases: object

    abs = <gnes.score_fn.base.ModifierScoreFn object>
    avg = <gnes.score_fn.base.CombinedScoreFn object>
    ln = <gnes.score_fn.base.ModifierScoreFn object>
    ln1p = <gnes.score_fn.base.ModifierScoreFn object>
    ln2p = <gnes.score_fn.base.ModifierScoreFn object>
    log = <gnes.score_fn.base.ModifierScoreFn object>
    log1p = <gnes.score_fn.base.ModifierScoreFn object>
    log2p = <gnes.score_fn.base.ModifierScoreFn object>
    max = <gnes.score_fn.base.CombinedScoreFn object>
    min = <gnes.score_fn.base.CombinedScoreFn object>
    multiply = <gnes.score_fn.base.CombinedScoreFn object>

```

```
none = <gnes.score_fn.base.ModifierScoreFn object>
reciprocal = <gnes.score_fn.base.ModifierScoreFn object>
reciprocallp = <gnes.score_fn.base.ModifierScoreFn object>
sqrt = <gnes.score_fn.base.ModifierScoreFn object>
square = <gnes.score_fn.base.ModifierScoreFn object>
sum = <gnes.score_fn.base.CombinedScoreFn object>
gnes.score_fn.base.get_unary_score (value, **kwargs)
```

gnes.score_fn.chunk module

```
class gnes.score_fn.chunk.BM25ChunkScoreFn (threshold=0.8, *args, **kwargs)
    Bases: gnes.score_fn.base.CombinedScoreFn
    score = relevance * idf(q_chunk) * tf(q_chunk) * (k1 + 1) / (tf(q_chunk) + k1 * (1 - b + b * (chunk_in_doc / avg_chunk_in_doc)))
    in bm25 algorithm: idf(q_chunk) = log(1 + (doc_count - f(q_chunk) + 0.5) / (f(q_chunk) + 0.5)),
    where f(q_chunk) is number of docs that contains q_chunk. In our system, this denotes number of docs appearing in query results.
    In elastic search, b = 0.75, k1 = 1.2
    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.score_fn.chunk.CoordChunkScoreFn (score_mode='multiply', *args, **kwargs)
    Bases: gnes.score_fn.base.CombinedScoreFn
    score = relevance * query_coordination query_coordination: #chunks return / #chunks in this doc(query doc)
    Parameters score_mode (str) – specifies how the computed scores are combined
    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.score_fn.chunk.TFIDFChunkScoreFn (threshold=0.8, *args, **kwargs)
    Bases: gnes.score_fn.base.CombinedScoreFn
    score = relevance * tf(q_chunk) * (idf(q_chunk)**2) tf(q_chunk) is calculated based on the relevance of query result. tf(q_chunk) = number of queried chunks where relevance >= threshold idf(q_chunk) = log(total_chunks / tf(q_chunk) + 1)
    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.score_fn.chunk.WeightedChunkOffsetScoreFn (score_mode='multiply', *args, **kwargs)
    Bases: gnes.score_fn.base.CombinedScoreFn
    score = d_chunk.weight * relevance * offset_divergence * q_chunk.weight offset_divergence is calculated based on doc_type:
    TEXT && VIDEO && AUDIO: offset is 1-D IMAGE: offset is 2-D
    Parameters score_mode (str) – specifies how the computed scores are combined
```

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

```
class gnes.score_fn.chunk.WeightedChunkScoreFn (score_mode='multiply', *args,
                                                **kwargs)
```

Bases: *gnes.score_fn.base.CombinedScoreFn*

score = d_chunk.weight * relevance * q_chunk.weight

Parameters **score_mode** (str) – specifies how the computed scores are combined

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

gnes.score_fn.doc module

```
class gnes.score_fn.doc.CoordDocScoreFn (score_mode='multiply', *args, **kwargs)
```

Bases: *gnes.score_fn.base.CombinedScoreFn*

score = score * query_coordination query_coordination: #chunks recalled / #chunks in this doc

Parameters **score_mode** (str) – specifies how the computed scores are combined

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

```
class gnes.score_fn.doc.WeightedDocScoreFn (score_mode='multiply', *args, **kwargs)
```

Bases: *gnes.score_fn.base.CombinedScoreFn*

Parameters **score_mode** (str) – specifies how the computed scores are combined

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

gnes.score_fn.normalize module

```
class gnes.score_fn.normalize.Normalizer1
```

Bases: *gnes.score_fn.base.ModifierScoreFn*

Do normalizing: score = 1 / (1 + sqrt(score))

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

```
class gnes.score_fn.normalize.Normalizer2 (num_dim)
```

Bases: *gnes.score_fn.base.ModifierScoreFn*

Do normalizing: score = 1 / (1 + score / num_dim)

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

```
class gnes.score_fn.normalize.Normalizer3 (num_dim)
```

Bases: *gnes.score_fn.normalize.Normalizer2*

Do normalizing: score = 1 / (1 + sqrt(score) / num_dim)

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

```
class gnes.score_fn.normalize.Normalizer4 (num_bytes)  
    Bases: gnes.score_fn.base.ModifierScoreFn
```

Do normalizing: $\text{score} = 1 - \text{score} / \text{num_bytes}$

```
train (*args, **kwargs)  
    Train the model, need to be overridden
```

```
class gnes.score_fn.normalize.Normalizer5  
    Bases: gnes.score_fn.base.ModifierScoreFn
```

Do normalizing: $\text{score} = 1 / (1 + \text{sqrt}(\text{abs}(\text{score})))$

```
train (*args, **kwargs)  
    Train the model, need to be overridden
```

Module contents

gnes.service package

Submodules

gnes.service.base module

```
class gnes.service.base.BaseService (args)  
    Bases: object
```

```
close ()
```

```
default_host = '0.0.0.0'
```

```
dump (respect_dump_interval=True)
```

```
handler = <gnes.service.base.MessageHandler object>
```

```
load_model (base_class, yaml_path=None)
```

Return type ~T

```
post_init ()
```

```
run ()
```

```
status
```

```
class gnes.service.base.BetterEnum  
    Bases: enum.Enum
```

An enumeration.

```
from_string = <bound method BetterEnum.from_string of <enum 'BetterEnum'>>
```

```
exception gnes.service.base.BlockMessage  
    Bases: Exception
```

```
exception gnes.service.base.ComponentNotLoad  
    Bases: Exception
```

```
class gnes.service.base.ConcurrentService  
    Bases: type
```

```
exception gnes.service.base.EventLoopEnd  
    Bases: Exception
```

```

class gnes.service.base.MessageHandler (mh=None)
    Bases: object

    call_hooks (msg, hook_type, *args, **kwargs)
        All post handler hooks are called after the handler is done but before sending out the message to the
        next service. All pre handler hooks are called after the service received a message and before calling the
        message handler

    call_routes (msg)

    call_routes_send_back (msg, out_sock)

    register (msg_type)

    register_hook (hook_type, only_when_verbose=False)
        Register a function as a pre/post hook

        Parameters

- only_when_verbose (bool) – only call the hook when verbose is true
- hook_type (Union[str, Tuple[str]]) – possible values ‘pre’ or ‘post’ or (‘pre’, ‘post’)

class gnes.service.base.ParallelType
    Bases: gnes.service.base.BetterEnum

    An enumeration.

    PUB_BLOCK = 2

    PUB_NONBLOCK = 3

    PUSH_BLOCK = 0

    PUSH_NONBLOCK = 1

    is_block

    is_push

class gnes.service.base.ReduceOp
    Bases: gnes.service.base.BetterEnum

    An enumeration.

    ALWAYS_ONE = 1

    CONCAT = 0

exception gnes.service.base.ServiceError
    Bases: Exception

class gnes.service.base.ServiceManager (service_cls, args)
    Bases: object

    join ()

class gnes.service.base.SocketType
    Bases: gnes.service.base.BetterEnum

    An enumeration.

    PAIR_BIND = 8

    PAIR_CONNECT = 9

    PUB_BIND = 6

```

```
PUB_CONNECT = 7
PULL_BIND = 0
PULL_CONNECT = 1
PUSH_BIND = 2
PUSH_CONNECT = 3
SUB_BIND = 4
SUB_CONNECT = 5

is_bind
paired
```

```
gnes.service.base.build_socket (ctx, host, port, socket_type, identity=None, use_ipc=False)
```

```
    Return type Tuple[Socket, str]
```

```
gnes.service.base.get_random_ipc ()
```

```
    Return type str
```

```
gnes.service.base.send_ctrl_message (address, msg, timeout)
```

gnes.service.encoder module

```
class gnes.service.encoder.EncoderService (args)
    Bases: gnes.service.base.BaseService
    embed_chunks_in_docs (docs, do_encoding=True, is_input_list=True)
    handler = <gnes.service.base.MessageHandler object>
    post_init ()
```

gnes.service.frontend module

```
class gnes.service.frontend.FrontendService (args)
    Bases: object
    join ()
    stop ()
```

gnes.service.grpc module

```
class gnes.service.grpc.GRPCService (args)
    Bases: gnes.service.base.BaseService
    close ()
    handler = <gnes.service.base.MessageHandler object>
    post_init ()
```


gnes.service.indexer module

```

class gnes.service.indexer.IndexerService(args)
    Bases: gnes.service.base.BaseService

    handler = <gnes.service.base.MessageHandler object>
    post_init()

```

gnes.service.preprocessor module

```

class gnes.service.preprocessor.PreprocessorService(args)
    Bases: gnes.service.base.BaseService

    handler = <gnes.service.base.MessageHandler object>
    post_init()

```

gnes.service.router module

```

class gnes.service.router.RouterService(args)
    Bases: gnes.service.base.BaseService

    handler = <gnes.service.base.MessageHandler object>
    post_init()

```

Module contents

1.2.2 Submodules

gnes.component module

gnes.helper module

```

gnes.helper.get_sys_info()
gnes.helper.get_optimal_sample_size(x)
gnes.helper.get_perm(L, m)
gnes.helper.time_profile(func)
gnes.helper.set_logger(context, verbose=False)
gnes.helper.batch_iterator(data, batch_size, axis=0)
    Return type Iterator[Any]
gnes.helper.batching(func=None, *, batch_size=None, num_batch=None, iter_axis=0, con-
    cat_axis=0, chunk_dim=-1)
gnes.helper.load_contrib_module()
gnes.helper.parse_arg(v)
gnes.helper.profiling(func)

```

```
class gnes.helper.FileLock (lock_file='LOCK')
```

Bases: object

Implements the Posix based file locking (Linux, Ubuntu, MacOS, etc.)

```
acquire()
```

```
is_locked
```

```
release()
```

```
gnes.helper.train_required (func)
```

```
gnes.helper.get_first_available_gpu()
```

```
class gnes.helper.PathImporter
```

Bases: object

```
static add_modules (*paths)
```

```
gnes.helper.progressbar (i, prefix="", suffix="", count=100, size=60)
```

Example:

```
for i in range(10000): progressbar(i, prefix="computing: ", count=100, size=60)
```

The resulted output is: computing: [#####.] 199/200
 99/100 computing: [#####.] 199/200
 computing: [#####.] 299/300
 computing: [#####.] 399/400
 computing: [#####.] 499/500
 computing: [#####.] 599/600
 computing: [#####.] 699/700
 computing: [#####.] 799/800 com-
 puting: [#####.] 899/900 computing:
 [#####.....] 950/1000

```
class gnes.helper.Singleton (cls)
```

Bases: object

Make your class singleton

gnes.uuid module

1.2.3 Module contents

1.3 Troubleshooting

1.3.1 Check if docker swarm/stack runs successfully

```
docker service ls
```

ID	NAME	MODE	REPLICAS
↪	IMAGE	PORTS	
j7b533zxmzg5	gnes-swarm-2654_encoder	replicated	0/1
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master		
0vlxu4acg1ph	gnes-swarm-2654_income-proxy	replicated	0/1
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master	*:4962->4962/tcp	
equqrhsn7pky	gnes-swarm-2654_indexer	replicated	0/3
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master		

(continues on next page)

(continued from previous page)

nd7euo7mcpa9	gnes-swarm-2654_middleman-proxy	replicated	0/1	↵
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master			
ssdlk9gzmggw	gnes-swarm-2654_outgoing-proxy	replicated	0/1	↵
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master	*:4963->4963/tcp		
xgxeetyhos6t	my-gnes_encoder	replicated	1/1	↵
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f			
zny37400p225	my-gnes_income-proxy	replicated	1/1	↵
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f	*:8598->8598/tcp		
taqqg6qwrxlw	my-gnes_indexer	replicated	3/3	↵
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f			
j96gnyy8ysbn	my-gnes_middleman-proxy	replicated	1/1	↵
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f			
e28spnuksjw8	my-gnes_outgoing-proxy	replicated	1/1	↵
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f	*:8599->8599/tcp		

In the above example, we started two swarms, i.e. `gnes-swarm-2654` and `my-gnes`. Unfortunately, `gnes-swarm-2654` fails to start and is not running at all. But how can one tell that?

Note the column `REPLICAS`, which indicates the number of running service (versus the number of required services). `gnes-swarm-2654` gives 0/0 for all services. This suggests the swarm fails to start. The next step is to investigate the reason.

1.3.2 Investigate the reason of a failed service

One can not print out all logs of a docker swarm. Instead, one can inspect service by service, e.g.

```
docker service ps gnes-swarm-2654_encoder --format "{{json .Error}}" --no-trunc
```

```
"\"invalid mount config for type \"bind\": bind source path does not exist: /data/han/
↪test-shell/output_data\""
\"invalid mount config for type \"bind\": bind source path does not exist: /data/han/
↪test-shell/output_data\""
\"invalid mount config for type \"bind\": bind source path does not exist: /data/han/
↪test-shell/output_data\""
\"invalid mount config for type \"bind\": bind source path does not exist: /data/han/
↪test-shell/output_data\""
```

Now the reason is clear, `output_data` does not exist when starting the swarm. But why there are duplicated lines there? This is because docker swarm did three retries before giving up on starting this service, where each time it met the same problem. Thus four duplicated lines in total.

1.3.3 Delete a failed service

Now that the reason is clear, we can delete the failed service and release the resources.

```
docker stack rm gnes-swarm-2654
```

```
Removing service gnes-swarm-2654_encoder
Removing service gnes-swarm-2654_income-proxy
Removing service gnes-swarm-2654_indexer
Removing service gnes-swarm-2654_middleman-proxy
Removing service gnes-swarm-2654_outgoing-proxy
Removing network gnes-swarm-2654_gnes-net
```

1.3.4 Locate internal errors by looking at logs

Sometime the service fails to start but `docker service ps` gives no error,

```
docker service ps gnes-swarm-4254_encoder --format "{{json .Error}}" --no-trunc
```

```
""
```

Or it shows an error that is not explanatory.

```
"\"task: non-zero exit (2)\""
```

Often in this case, the service fails to start *not* due to the docker config, but due to the GNES internal error. To see that,

```
docker service logs gnes-swarm-4254_income-proxy
```

```
gnes-swarm-4254_income-proxy.1.yj5v8n4dhfgv@VM-0-3-ubuntu | [--
↪proxy_type {BS,Dict,MapProxyService,Message,MessageHandler,ProxyService,
↪ReduceProxyService,defaultdict}}
gnes-swarm-4254_income-proxy.1.yj5v8n4dhfgv@VM-0-3-ubuntu | [--
↪batch_size BATCH_SIZE] [--num_part NUM_PART]
gnes-swarm-4254_income-proxy.1.kmgk21qo6m0n@VM-0-3-ubuntu | [--
↪proxy_type {BS,Dict,MapProxyService,Message,MessageHandler,ProxyService,
↪ReduceProxyService,defaultdict}}
gnes-swarm-4254_income-proxy.1.w04d552cuj93@VM-0-3-ubuntu | gnes proxy: error:
↪argument --batch_size: invalid int value: ''
gnes-swarm-4254_income-proxy.1.kmgk21qo6m0n@VM-0-3-ubuntu | [--
↪batch_size BATCH_SIZE] [--num_part NUM_PART]
```

One can now clearly see that the error comes from an incorrectly given `--batch_size`, which throws from GNES CLI.

1.4 Protobuf Implementation

The file `gnes/proto/gnes.proto` defines the protobuf used in GNES. It is the core message protocol used in communicating between services. It also defines the interface of a gRPC service.

`gnes_pb2.py` and `gnes_pb2_grpc.py` are python interfaces automatically generated by protobuf tools.

For developers who want to change the protobuf definition, one needs to first edit `gnes/proto/gnes.proto` and then regenerate the python codes (i.e. `gnes_pb2.py` and `gnes_pb2_grpc.py`).

1.4.1 Generating `gnes_pb2.py` and `gnes_pb2_grpc.py`

Take MacOS as an example,

1. Download `protoc-$VERSION-$PLATFORM.zip` from [the official site](#) and decompress it.
2. Copy the binary file and include to your system path:

```
cp ~/Downloads/protoc-3.7.1-osx-x86_64/bin/protoc /usr/local/bin/
cp -r ~/Downloads/protoc-3.7.1-osx-x86_64/include/* /usr/local/include/
```

1. Install gRPC tools dependencies: `brew install automake autoconf libtool`

2. Install gRPC and `grpc_python_plugin` from the source:

```
git clone https://github.com/grpc/grpc.git
git submodule update --init
make grpc_python_plugin
```

1. This will compile the `grpc-python-plugin` and build it to, e.g., `/Documents/grpc/bins/opt/grpc_python_plugin`
2. Generate the python codes:

```
SRC_DIR=gnες/proto/
PLUGIN_PATH=/Documents/grpc/bins/opt/grpc_python_plugin

protoc -I $SRC_DIR --python_out=$SRC_DIR --grpc_python_out=$SRC_DIR --plugin=protoc-
↳gen-grpc_python=${PLUGIN_PATH} ${SRC_DIR}gnες.proto
```

1. Fixing the import in `gnες_pb2_grpc.py`. For some reason (probably a bug of gRPC?), the generated code of import is not correct in `gnες_pb2_grpc.py`, you have to change it to the following:

```
# Generated by the gRPC Python protocol compiler plugin. DO NOT EDIT!
import grpc

from . import gnes_pb2 as gnes__pb2
```

1.5 Environment Variables

There are couple of environment variables that GNES respect during runtime.

1.5.1 GNES_PROFILING

Set to any non-empty string to turn on service-level time profiling for GNES.

Default is disabled.

1.5.2 GNES_PROFILING_MEM

Set to any non-empty string to turn on service-level memory profiling for GNES. Warning, memory profiling could hurt the efficiency significantly.

Default is disabled.

1.5.3 GNES_WARN_UNNAMED_COMPONENT

Set to 0 to turn off the warning like this object is not named ("name" is not found under "gnες_config" in YAML config), i will call it "BaseRouter-51ce94cc". naming the object is important as it provides an unique identifier when serializing/deserializing this object.

Set to 1 to enable it.

Default is enabled.

1.5.4 GNES_VCS_VERSION

Git version of GNES. This is used when `--check_version` is turned on. For GNES official docker image, `GNES_VCS_VERSION` is automatically set to the git version during the building procedure.

Default is the git head version when building docker image. Otherwise it is not set.

1.5.5 GNES_CONTROL_PORT

Control port of the microservice. Useful when doing health check via `gnes healthcheck`.

Default is not set. A random port will be used.

1.5.6 GNES_CONTRIB_MODULE

(deprecated) Paths of the third party components. See examples in GNES hub for latest usage.

1.5.7 GNES_IPC SOCK_TMP

Temp directory for ipc sockets, not used on Windows.

1.6 Using GNES with Docker Swarm

1.6.1 Build your first GNES app on local machine

Let's start with a typical indexing procedure by writing a YAML config (see the left column of the table):

Now let's see what the YAML config says. First impression, it is pretty intuitive. It defines a pipeline workflow consists of preprocessing, encoding and indexing, where the output of the former component is the input of the next. This pipeline is a typical workflow of *index* or *query* runtime. Under each component, we also associate it with a YAML config specifying how it should work. Right now they are not important for understanding the big picture, nonetheless curious readers can checkout how each YAML looks like by expanding the items below.

```
!SentSplitPreprocessor
parameters:
  start_doc_id: 0
  random_doc_id: True
  delimiter: "[.!?]+"
gnes_config:
  is_trained: true
```

```
!PipelineEncoder
components:
- !GPT2Encoder
  parameters:
    model_dir: $GPT2_CI_MODEL
    pooling_stragy: REDUCE_MEAN
  gnes_config:
    is_trained: true
- !PCALocalEncoder
  parameters:
```

(continues on next page)

(continued from previous page)

```

    output_dim: 32
    num_locals: 8
    gnes_config:
      batch_size: 2048
  - !PQEncoder
    parameters:
      cluster_per_byte: 8
      num_bytes: 8
    gnes_config:
      work_dir: ./
    name: gpt2bin-pipe

```

```

!BIndexer
parameters:
  num_bytes: 8
  data_path: /out_data/idx.binary
gnest_config:
  work_dir: ./
  name: bindexer

```

On the right side of the above table, you can see how the actual data flow looks like. There is an additional component gRPCFrontend automatically added to the workflow, it allows you to feed the data and fetch the result via gRPC protocol through port 5566.

Now it's time to run! **GNES board** can automatically generate a starting script/config based on the YAML config you give, saving troubles of writing them on your own.

You can also start a GNES board locally. Simply run `docker run -d -p 0.0.0.0:80:8080/tcp gnes/gnes compose --serve`

As a cloud-native application, GNES requires an **orchestration engine** to coordinate all micro-services. We support Kubernetes, Docker Swarm and shell-based multi-process. Let's see what the generated script looks like in this case.

```

#!/usr/bin/env bash
set -e

trap 'kill $(jobs -p)' EXIT

printf "starting service gRPCFrontend with 0 replicas...\n"
gnest frontend --grpc_port 5566 --port_out 49668 --socket_out PUSH_BIND --port_in_
↪60654 --socket_in PULL_CONNECT &
printf "starting service Preprocessor with 0 replicas...\n"
gnest preprocess --yaml_path text-prep.yml --port_in 49668 --socket_in PULL_CONNECT --
↪port_out 61911 --socket_out PUSH_BIND &
printf "starting service Encoder with 0 replicas...\n"
gnest encode --yaml_path gpt2.yml --port_in 61911 --socket_in PULL_CONNECT --port_out_
↪49947 --socket_out PUSH_BIND &
printf "starting service Indexer with 0 replicas...\n"
gnest index --yaml_path b-indexer.yml --port_in 49947 --socket_in PULL_CONNECT --port_
↪out 60654 --socket_out PUSH_BIND &

wait

```

```

version: '3.4'
services:
  gRPCFrontend00:

```

(continues on next page)

(continued from previous page)

```

image: gnes/gnes-full:latest
command: frontend --grpc_port 5566 --port_out 49668 --socket_out PUSH_BIND --port_
↪in
    60654 --socket_in PULL_CONNECT --host_in Indexer30
ports:
- 5566:5566
Preprocessor10:
image: gnes/gnes-full:latest
command: preprocess --port_in 49668 --socket_in PULL_CONNECT
    --port_out 61911 --socket_out PUSH_BIND --yaml_path /Preprocessor10_yaml --host_
↪in
    gRPCFrontend00
configs:
- Preprocessor10_yaml
Encoder20:
image: gnes/gnes-full:latest
command: encode --port_in 61911 --socket_in PULL_CONNECT
    --port_out 49947 --socket_out PUSH_BIND --yaml_path /Encoder20_yaml --host_in
    Preprocessor10
configs:
- Encoder20_yaml
Indexer30:
image: gnes/gnes-full:latest
command: index --port_in 49947 --socket_in PULL_CONNECT
    --port_out 60654 --socket_out PUSH_BIND --yaml_path /Indexer30_yaml --host_in
    Encoder20
configs:
- Indexer30_yaml
volumes: {}
networks:
gnes-net:
driver: overlay
attachable: true
configs:
Preprocessor10_yaml:
file: text-prep.yml
Encoder20_yaml:
file: gpt2.yml
Indexer30_yaml:
file: b-indexer.yml

```

For the sake of simplicity, we will just use the generated shell-script to start GNES. Create a new file say `run.sh`, copy the content to it and run it via `$ bash ./run.sh`. You should see the output as follows:

This suggests the GNES app is ready and waiting for the incoming data. You may now feed data to it through the `gRPCFrontend`. Depending on your language (Python, C, Java, Go, HTTP, Shell, etc.) and the content form (image, video, text, etc), the data feeding part can be slightly different.

To stop a running GNES, you can simply do control + c.

1.6.2 Scale your GNES app to the cloud

Now let's juice it up a bit. To be honest, building a single-machine process-based pipeline is not impressive anyway. The true power of GNES is that you can scale any component at any time you want. Encoding is slow? Adding more machines. Preprocessing takes too long? More machines. Index file is too large? Adding shards, aka. more machines!

In this example, we compose a more complicated GNES workflow for images. This workflow consists of multiple

preprocessors, encoders and two types of indexers. In particular, we introduce two types of indexers: one for storing the encoded binary vectors, the other for storing the original images, i.e. full-text index. These two types of indexers work in parallel. Check out the YAML file on the left side of table for more details, note how `replicas` is defined for each component.

You may realize that besides the `gRPCFrontend`, multiple `Router` have been added to the workflow. Routers serve as a message broker between microservices, determining how and where the message is received and sent. In the last pipeline example, the data flow is too simple so there is no need for adding any router. In this example routers are necessary for connecting multiple preprocessors and encoders, otherwise preprocessors wouldn't know where to send the message. GNES Board automatically adds router to the workflow when necessary based on the type of two consecutive layers. It may also add stacked routers, as you can see between encoder and indexer in the right graph.

Again, the detailed YAML config of each component is not important for understanding the big picture, hence we omit it for now.

This time we will run GNES via DockerSwarm. To do that simply copy the generated DockerSwarm YAML config to a file say `my-gnes.yml`, and then do

```
docker stack deploy --compose-file my-gnes.yml gnes-531
```

Note that `gnes-531` is your GNES stack name, keep that name in mind. If you forget about that name, you can always use `docker stack ls` to find out. To tell whether the whole stack is running successfully or not, you can use `docker service ls -f name=gnes-531`. The number of replicas 1/1 or 4/4 suggests everything is fine.

Generally, a complete and successful Docker Swarm starting process should look like the following:

When the GNES stack is ready and waiting for the incoming data, you may now feed data to it through the `gRPCFrontend`. Depending on your language (Python, C, Java, Go, HTTP, Shell, etc.) and the content form (image, video, text, etc), the data feeding part can be slightly different.

To stop a running GNES stack, you can use `docker stack rm gnes-531`.

1.6.3 Customize GNES to your need

With the help of GNES Board, you can easily compose a GNES app for different purposes. The table below summarizes some common compositions with the corresponding workflow visualizations. Note, we hide the component-wise YAML config (i.e. `yaml_path`) for the sake of clarity.

1.7 Contributing to GNES

Thanks for your interest in contributing! GNES always welcome the contribution from the open-source community, individual committers and other partners. Without you, GNES can't be successful.

1.7.1 Making Your First Commit

The beginning is always the hardest. But fear not, even if you find a typo, a missing docstring or unit test, you can simply correct them by making a commit to GNES. Here are the steps:

1. Create a new branch, say `fix-gnes-typo-1`
2. Fix/improve the codebase
3. Commit the changes. Note the **commit message must follow *the naming style***, say `fix(readme): improve the readability and move sections`

4. Make a pull request. Note the **commit message must follow *the naming style***. It can simply be one of your commit messages, just copy paste it, e.g. `fix(readme): improve the readability and move sections`
5. Submit your pull request and wait for all checks passed (usually 10 minutes)
 - Coding style
 - Commit and PR styles check
 - All unit tests
6. Request reviews from one of the developers from our core team.
7. Get a LGTM and PR gets merged.

Well done! Once a PR gets merged, here are the things happened next:

- all Docker images tagged with `-latest` will be automatically updated in an hour. You may check the [its building status at here](#)
- on every Friday when a new release is published, PyPi packages and all Docker images tagged with `-stable` will be updated accordingly.
- your contribution and commits will be included in [our weekly release note](#).

1.7.2 Table of Content

- *Commit Message Naming*
- *Merging Process*
- *Release Process*
 - *Major and minor version increments*
- *Testing Locally*
- *Interesting Points*

1.7.3 Commit Message Naming

To help everyone with understanding the commit history of GNES, we employ `commitlint` in the CI pipeline to enforce the commit styles. Specifically, our convention is:

```
type(scope?): subject
```

where `type` is one of the following:

- build
- ci
- chore
- docs
- feat
- fix
- perf
- refactor

- revert
- style
- test

scope is optional, represents the module your commit working on.

subject explains the commit.

As an example, a commit that implements a new encoder should be phrased as:

```
feat(encoder): add new inceptionV3 as image encoder
```

1.7.4 Merging Process

A pull request has to meet the following conditions to be merged into master:

- Coding style check (PEP8, via Codacy)
- Commit style check (in CI pipeline via Drone.io)
- Unit tests (via Drone.io)
- Review and approval from a GNES team member.

After the merging is triggered, the build will be delivered to the followings:

- **Docker Hub:** gnes:latest will be updated.
- **Tencent Container Service:** gnes:latest will be updated.
- **ReadTheDoc:** latest will be updated.
- **Benchmark:** speed test will be updated.

Note that merging into master does not mean an official releasing. For the releasing process, please refer to the next section.

1.7.5 Release Process

A new release is scheduled on every Friday (triggered and approved by [Han Xiao](#)) summarizing all new commits since the last release. The release will increment the third (revision) part of the version number, i.e. from 0.0.24 to 0.0.25.

After a release is triggered, the build will be delivered to the followings:

- **Docker Hub:** a new image with the release version tag will be created, gnes:latest will be updated.
- **Tencent Container Service:** a new image with the release version tag will be created, gnes:latest will be updated.
- **PyPi Package:** a new version of Python package is uploaded to Pypi, allowing one to `pip install -U gnes`
- **ReadTheDoc:** a new version of the document will be built, latest will be updated and the old version will be archived
- **Benchmark:** speed test will be updated.

Meanwhile, a new pull request containing the updated [CHANGELOG](#) and the new version number will be made automatically, pending for review and merge.

Major and minor version increments

- MAJOR version when GNES make incompatible API changes;
- MINOR version when GNES add functionality in a backwards-compatible manner.

The decision of incrementing major and minor version, i.e. from 0.0.0 to 0.1.0 or from 1.0.0 to 2.0.0, is made by the GNES team.

1.7.6 Testing Locally

The best way to test GNES is using a Docker container, in which you don't have to worry about the dependencies.

We provide a public Docker image `gnes/ci-base`, which contains the required dependencies and some pretrained models used in our continuous integration pipeline.

You can [find the image at here](#) or pull the image via:

```
docker pull gnes/ci-base
```

To test GNES inside this image, you may run

```
docker run --network=host --rm --entrypoint "/bin/bash" -it gnes/ci-base

# now you are inside the 'gnes/ci-base' container
# first sync your local modification, then
pip install -e .[all]
python -m unittest tests/*.py
```

1.7.7 Interesting Points

Currently there are three major directions of contribution:

- **Porting state-of-the-art models to GNES.** This includes new preprocessing algorithms, new DNN networks for encoding, and new high-performance index. Believe me, it is super easy to wrap an algorithm and use it in GNES. Checkout this example.
- **Adding tutorial and learning experience.** What is good and what can be improved? If you apply GNES in your domain, whether it's about NLP or CV, whether it's a blog post or a Reddit/Twitter thread, we are always eager to hear your thoughts.
- **Completing the user experience of other programming languages.** GNES offers a generic interface with gRPC and protobuf, therefore it is easy to add an interface for other languages, e.g. Java, C, Go.

1.8 Release Note (v0.0.46)

Release time: 2019-10-17 18:14:45

We'd like to thank all contributors for this new release! In particular, Han Xiao, felix, raccoonliukai, hanhxiao, Jem,

1.8.1 New Features

- [\[02941b6c\]](#) - **parser**: add `ctrl_with_ipc` to cli (*hanhxiao*)

1.8.2 Bug fixes

- [dbf1a5e7] - **release**: add some hints to the release script (*Han Xiao*)
- [a641d5c7] - **shot-detector**: rename shot detector (*felix*)
- [d48eb53a] - **preprocessor**: add max_shot_num for shotdetect (*raccoonliukai*)
- [9331ef58] - **flow**: use recommend flow api to reduce confusion (*hanhxiao*)
- [660f8f99] - **flow**: make base flow as class not object (*hanhxiao*)
- [3062c43c] - **ci**: remove cffi from gnes docker image (*hanhxiao*)
- [38147fe8] - **helper**: fix gpuutil exception (*hanhxiao*)
- [15d9b4fe] - **ci**: fix cffi version to 1.12.3 (*hanhxiao*)
- [cbac2de4] - **ci**: fix cffi to 1.12.3 (*hanhxiao*)
- [24e41bec] - **preprocessor**: add numpy transform (*Jem*)
- [6008f7d1] - **service**: revert socket log (*Han Xiao*)
- [474deddf] - **control-sock**: build control socket at the beginning (*felix*)
- [707d9e96] - **service-logging**: show socket creating (*felix*)
- [01531f74] - **stream-call**: hungry mode to receive responses (*felix*)

1.8.3 Code Refactoring

- [f16c672e] - **client**: make query method as generator (*hanhxiao*)

1.8.4 Documentation

- [ec03351d] - **flow**: add flow to readme as main api (*Han Xiao*)

1.8.5 Other Improvements

- [31c05416] - **flow**: mark flow as a breaking change (*Han Xiao*)
- [4efd414e] - **changelog**: update change log to v0.0.45 (*hanhxiao*)
- [14fa5a18] - ... (*felix*)

1.9 Breaking Changes (v0.0.45 -> v0.0.46)

The new GNES Flow API introduced since v0.0.46 has become the main API of GNES. It provides a pythonic and intuitive way of building pipelines in GNES, enabling run/debug on a local machine. It also supports graph visualization, swarm/k8s config export, etc. More information about [GNES Flow](#) can be found at here.

As a consequence, the `composer` module as well as `gnes compose` CLI and GNES board web UI will be removed in the next releases.

GNES board will be redesigned using the GNES Flow API. We highly [welcome your contribution on this thread!](#)

1.10 Release Note (v0.0.45)

Release time: 2019-10-15 14:01:07

We'd like to thank all contributors for this new release! In particular, hanhxiao, felix, Han Xiao,

1.10.1 New Features

- [166698ce] - **flow**: add index and query flow as common flow (*hanhxiao*)
- [8a60c261] - **flow**: flow can not export docker swarm config (*hanhxiao*)
- [80cb530e] - **flow**: add flow to python generator (*hanhxiao*)
- [f6536c87] - **flow**: add eq operator to the flow to enable comparison (*hanhxiao*)
- [9ca757b4] - **flow**: add set remove and set_last (*hanhxiao*)
- [3c3c54b5] - **webp-encoder**: support webp encoder (*felix*)
- [b94490f1] - **flow**: allow add service to be str (*hanhxiao*)
- [4055ad8e] - **flow**: add support to replicas plot (*hanhxiao*)
- [7265f76c] - **grpc**: add proxy argument to cli (*hanhxiao*)
- [3901078c] - **inception_v4_encoder**: add inception v4 encoder for video (*felix*)

1.10.2 Bug fixes

- [8911314b] - **style**: double quote to single quote (*Han Xiao*)
- [228a2b19] - **flow**: fix unit test assert in flow (*hanhxiao*)
- [7d2c681e] - **flow**: add warning to jpg downloader (*hanhxiao*)
- [fce94d94] - **service**: fix ServiceManager replicas router (*hanhxiao*)
- [2705c287] - **video-decoder**: none chunk splitter (*felix*)

1.10.3 Code Refactoring

- [48d1828c] - **flow**: reorganize file structure for flow (*hanhxiao*)
- [c8f1df5b] - **flow**: flow is now a trainable base (*hanhxiao*)

1.10.4 Other Improvements

- [5121d65a] - revert unittest (*felix*)
- [8ada2538] - **changelog**: update change log to v0.0.44 (*hanhxiao*)

1.11 Release Note (v0.0.44)

Release time: 2019-10-11 15:27:37

We'd like to thank all contributors for this new release! In particular, hanhxiao, felix, Jem,

1.11.1 New Features

- [2fb0f4f9] - **flow**: add dump to jpg (*hanhxiao*)
- [552fcdfe] - **indexer-cli**: add as_response switcher to indexer cli (*hanhxiao*)
- [c8cedd04] - **service**: remove async dump for better stability (*hanhxiao*)
- [1739c7b6] - **flow**: add client to flow (*hanhxiao*)
- [43b9d014] - **flow**: add context manager to flow (*hanhxiao*)
- [ae0d4056] - **flow**: first version of gnes flow (*hanhxiao*)

1.11.2 Bug fixes

- [c23ea61f] - **frontend**: fix frontend blocking behavior (*hanhxiao*)
- [c880c9b0] - **service**: make service handler thread-safe (*hanhxiao*)
- [a3da0582] - **flow**: fix flow unit test (*hanhxiao*)
- [6d118404] - **ffmpeg**: threads=1 (*felix*)
- [bca5b5b7] - **base**: fix env expansion in gnes_config (*hanhxiao*)
- [72f4a044] - **indexer**: fix empty chunk and dump_interval (*hanhxiao*)
- [9b79cdf5] - **memory-leak**: try to fix memory leak danger (*felix*)
- [16097f3f] - **video-decoder**: fix name (*felix*)
- [199a71a6] - **frontend**: remove duplicate receive (*hanhxiao*)
- [73dae6bd] - **service**: minor fix on the dump_interval (*hanhxiao*)
- [6f401905] - **client**: fix bugs for client (*Jem*)
- [c5af9308] - **parser**: use str instead of textio stream to prevent serializer err (*hanhxiao*)
- [6a368335] - **cli**: show more detailed version info in cli (*hanhxiao*)

1.11.3 Documentation

- [d60a24a9] - fix docs format (*hanhxiao*)

1.11.4 Other Improvements

- [3cc4b041] - fix unit test (*felix*)
- [57519198] - **changelog**: update change log to v0.0.43 (*hanhxiao*)

1.12 Release Note (v0.0.43)

Release time: 2019-09-30 17:37:58

We'd like to thank all contributors for this new release! In particular, felix, hanhxiao, raccoonliukai,

1.12.1 New Features

- [bbf1ed8e] - **frontend**: add max pending request to frontend (*hanhxiao*)
- [946df39b] - **indexer**: delay the num_dim spec on first add (*hanhxiao*)
- [fdc38d57] - **cd**: trigger benchmark in push and tag (*hanhxiao*)
- [f8b9e00e] - **cd**: smaller num document for benchmarking (*hanhxiao*)
- [3974a1ba] - **cd**: adding benchmark to cd pipeline (*hanhxiao*)
- [086a73cb] - **proto**: add vcs version to pb (*hanhxiao*)
- [285d9dde] - **docker**: add vcs version as env var (*hanhxiao*)
- [a3e22db3] - **service**: add healthcheck for arbitrary service (*hanhxiao*)
- [dedb8ba2] - **proto**: add ready status for healthcheck (*hanhxiao*)

1.12.2 Bug fixes

- [47add702] - **ffmpeg-threads**: threads=0 (*felix*)
- [9365ddb9] - **video-decoder**: minor revision video-decoder chunk splitter (*felix*)
- [be09bb09] - **cd**: fix duplicate step name in cd (*hanhxiao*)
- [ccf4efc8] - **cd**: fix trigger in cd pipeline (*hanhxiao*)
- [ca73b702] - **shotdetect**: support get arguments from yaml (*raccoonliukai*)
- [17aa78da] - **shotdetect**: fix bug with thre_algo after histcmp (*raccoonliukai*)
- [a6d1484e] - **shotdetect**: fix bug with thre_algo (*raccoonliukai*)
- [b69591de] - **shotdetect**: fix shot boundary (*raccoonliukai*)
- [f6c263a7] - **ffmpeg**: use -threads = 1 for ffmpeg (*felix*)
- [beafdb3a] - **docker**: fix vcs ref url and add build date as env (*hanhxiao*)
- [0367334a] - **service**: fix error message (*hanhxiao*)
- [434bc8db] - **service**: styling (*hanhxiao*)

1.12.3 Code Refactoring

- [e588c946] - **encoder**: update the init func for flair (*hanhxiao*)
- [1b85375c] - **client**: remove benchmark client (*hanhxiao*)
- [535085ef] - **video-decoder**: update video decoder signature and add more options (*felix*)
- [aa0f994e] - **cli**: refactor parser for healthcheck (*hanhxiao*)

1.12.4 Documentation

- [9ad7448f] - add env vars explain (*hanhxiao*)

1.12.5 Other Improvements

- [e180a9f0] - fix error (*felix*)
- [074eec0d] - **changelog**: update change log to v0.0.42 (*hanhxiao*)

1.13 Release Note (v0.0.42)

Release time: 2019-09-26 10:40:48

We'd like to thank all contributors for this new release! In particular, hanhxiao, felix, raccoonliukai, Jem,

1.13.1 New Features

- [8bef90dd] - **cd**: docker images now push to github during merge and tag (*hanhxiao*)
- [0ea566ff] - **service**: send ndarray separately (*hanhxiao*)
- [09199d82] - **preprocessor**: add frame selector (*Jem*)
- [a2f10589] - **parser**: add raw_bytes_in_sep to cli (*hanhxiao*)
- [10788951] - **proto**: speedup send/recv by separating raw_bytes from pb (*hanhxiao*)
- [2326fe97] - **preprocessor**: add preprocessor for mp4 and gif decode (*raccoonliukai*)
- [803afb34] - **snowflake-uuid**: add snowflake uuid generator (*felix*)
- [fe7025f5] - **frontend**: dump route in the frontend (*hanhxiao*)
- [8fbb0945] - **router**: add a block router for benchmarking (*hanhxiao*)

1.13.2 Bug fixes

- [43145019] - **unittest**: fix unit test for send recv (*hanhxiao*)
- [b6f2cda] - **service**: fix send/recv for better compatability (*hanhxiao*)
- [8c6f2558] - fix route table sum time (*raccoonliukai*)
- [8a0beec8] - **service**: send single long message rather than multiple (*hanhxiao*)
- [3b1f963c] - **preprocessor**: add solution for raw_video (*raccoonliukai*)
- [1b4a04fe] - **preprocessor**: add videodecode in init (*raccoonliukai*)
- [7108460a] - **memory_leak**: try to fix memory leak (*felix*)
- [82951d95] - **frontend**: use poll for better efficiency (*hanhxiao*)
- [2f539b7a] - **snowflake**: fix error shift (*felix*)
- [84e67792] - **frontend**: fix progressbar and route table (*hanhxiao*)
- [9a65e4fe] - **frontend**: flush dump (*hanhxiao*)
- [2e326af5] - catch exception in hook function (*hanhxiao*)
- [402867cc] - fix route table total time (*hanhxiao*)
- [30976179] - **docker**: decoupling prerequest and gnes install (*hanhxiao*)

- [c5347a5b] - **service**: make route_table as option for all services (*hanhxiao*)
- [45a078d9] - **docker**: reduce the size of built image (*hanhxiao*)

1.13.3 Documentation

- [95493b47] - **readme**: add github package to readme (*hanhxiao*)

1.13.4 Other Improvements

- [31f53bc2] - fix unit test (*felix*)
- [89945ef8] - **changelog**: update change log to v0.0.41 (*hanhxiao*)

1.14 Release Note (v0.0.41)

Release time: 2019-09-20 20:51:39

We'd like to thank all contributors for this new release! In particular, felix, raccoonliukai, hanhxiao, Jem,

1.14.1 New Features

- [e255bd48] - **shot-detector**: limit number of frames in shots (*felix*)
- [6e87afa4] - **traffic-controller**: network traffic controller in frontend (*felix*)
- [6833a27c] - **preprocessor**: add sframes for shots frame number (*raccoonliukai*)
- [ea89d8cb] - **stream-call**: only 1000 pending tasks (*felix*)
- [ca53c65f] - **video-encoder**: encode video from list of images (*felix*)

1.14.2 Bug fixes

- [6aa0c3ca] - **ffmpeg-video**: fix bug for scaling videos to stdout (*felix*)
- [780aad0d] - **subprocess**: close stdout and stderr to avoid memory leak (*felix*)
- [205962fb] - **socket**: raise socket rec/send message exception (*felix*)
- [64acb4cd] - **preprocessor**: fix bug when num_frames < 4 in shotdetect (*raccoonliukai*)
- [05db02f7] - **stream-client**: request queue size is limited by 1000 (*felix*)
- [4f389449] - **socket-buffer**: set hwm and buffer limit for zmq socket (*felix*)
- [092379e1] - **preprocessor**: fix type of index in shotdetect (*raccoonliukai*)
- [9023afcd] - **test**: add test to cover three runtimes (*hanhxiao*)
- [65ffff1a9] - **cli**: fix progressbar (*hanhxiao*)
- [8828535c] - **proto**: fix version check in recv message (*hanhxiao*)

1.14.3 Code Refactoring

- [d7c65727] - **scorefn**: move query coord into doc indexer (*Jem*)
- [b4007ba6] - **frontend**: realtime response to reduce network overload (*felix*)
- [de5b3368] - **shot-detector**: merge code from hub (*felix*)

1.14.4 Other Improvements

- [6acff20b] - clean unused variable (*felix*)
- [373df7da] - **changelog**: update change log to v0.0.40 (*hanhxiao*)

1.15 Release Note (v0.0.40)

Release time: 2019-09-12 19:54:34

We'd like to thank all contributors for this new release! In particular, Han Xiao, Jem, hanhxiao, felix,

1.15.1 New Features

- [b5e10bcd] - **scorefn**: add offset tfidf bm25 qureycoord (*Jem*)
- [20e7db6b] - **cli**: show default value of each argument (*hanhxiao*)
- [c6960b52] - **progressbar**: add dynamic progress bar (*felix*)
- [6c8017c1] - **frontend**: new route time table (*hanhxiao*)
- [81639268] - **frontend**: add route time table (*hanhxiao*)

1.15.2 Bug fixes

- [8704331e] - **proto**: fix version check in recv message (*hanhxiao*)
- [563a48c7] - **cli**: fix cli client required (*hanhxiao*)
- [3db34449] - **proto**: fix merge route logic (*hanhxiao*)
- [c31f21db] - **parser**: fix default dump interval to 5 (*hanhxiao*)
- [00c25f39] - **parser**: remove limite on message size (*hanhxiao*)
- [f89b4363] - **parser**: set dump_interval to -1 (*hanhxiao*)

1.15.3 Code Refactoring

- [a7a0ec8b] - **grpc-client**: add response handler (*felix*)
- [ce43be80] - **proto**: remove num_replicas from routes (*hanhxiao*)
- [d007fd9b] - **indexer**: remove unused code in leveldbindexer (*hanhxiao*)

1.15.4 Documentation

- [787d0f14] - **readme**: add poem search demo (*hanhxiao*)

1.15.5 Unit Test and CICD

- [3af95227] - **cli**: add unittest for cli (*Jem*)

1.15.6 Other Improvements

- [0b22029a] - **indexer**: fix styles in indexer (*Han Xiao*)
- [edba197a] - clean and format codes (*felix*)
- [2a781aee] - **license**: remove aiohttp from barebone GNES license (*hanhxiao*)
- [cc72cf2b] - **docker**: revert alpine docker to reduce size (*hanhxiao*)
- [9f58fb35] - **changelog**: update change log to v0.0.39 (*hanhxiao*)

1.16 Release Note (v0.0.39)

Release time: 2019-09-11 17:22:11

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, raccoonliukai, Larry Yan,

1.16.1 New Features

- [fdb53a09] - **client**: add status logging (*hanhxiao*)
- [621c06b3] - **client**: abstract query_callback (*hanhxiao*)
- [2c2a4d83] - **indexer**: add indexer info (*Jem*)
- [cfe97aab] - **proto**: add version info to proto (*hanhxiao*)
- [f1a187c1] - **proto**: versioning protobuf and validating in recv_message (*hanhxiao*)

1.16.2 Bug fixes

- [31c796d3] - **client**: fix weights in helper indexer (*hanhxiao*)
- [21c3a8a9] - **client**: rename stub to _stub (*hanhxiao*)
- [235d901a] - **parser**: add max_concurrency to client (*hanhxiao*)
- [c988b327] - **client**: fix sync client (*hanhxiao*)
- [54a252e5] - **indexer**: add helper indexer to registration (*hanhxiao*)
- [a1aed8f4] - **client**: use StreamingClient as the parent class of CLIClient (*raccoonliukai*)
- [a5999828] - **preprocessor**: add vframe(frame_num) for video and gif frames capture (*raccoonliukai*)
- [8357754a] - **encoder**: fix PCAEncoder mean from fp64 to fp32 (*raccoonliukai*)
- [654a5ba4] - **encoder**: fix vlad to speed up centroids calculation (*Larry Yan*)

- [814b2ee6] - **encoder**: fix vald encocer (*Larry Yan*)
- [ffc822b3] - **encoder**: fix vlad unittest (*Larry Yan*)
- [ddf13ff1] - **encoder**: fix bug in vlad encoder (*Larry Yan*)
- [1ba4e11c] - **encoder**: fix vald encoder and add unittest (*Larry Yan*)
- [f8e18d06] - **encoder**: fix vald in numeric encoder (*Larry Yan*)
- [fbfa1e47] - **transformer**: add model eval (*Jem*)

1.16.3 Code Refactoring

- [421e21f7] - **indexer**: add counter and move key_only to chunk helper (*hanhxiao*)
- [2a3f1210] - **router**: refactor base embed reduce router (*Jem*)

1.16.4 Documentation

- [d4f59211] - update component overview (*hanhxiao*)

1.16.5 Other Improvements

- [ded79bee] - **changelog**: update change log to v0.0.38 (*hanhxiao*)

1.17 Release Note (v0.0.38)

Release time: 2019-09-06 17:25:48

We'd like to thank all contributors for this new release! In particular, hanhxiao,

1.17.1 Other Improvements

- [71a3073a] - bump version (*hanhxiao*)
- [9666a462] - **changelog**: update change log to v0.0.37 (*hanhxiao*)

1.18 Release Note (v0.0.37)

Release time: 2019-09-06 16:46:20

We'd like to thank all contributors for this new release! In particular, hanhxiao, felix, Jem, raccoonliu, raccoonliukai, Han Xiao, Larry Yan,

1.18.1 New Features

- [105a0abf] - **encoder**: add debug hook (*hanhxiao*)
- [0f04877f] - **service**: add pre and post hooks to baseservice (*hanhxiao*)
- [92860848] - **reducer**: add concat reducer (*Jem*)
- [2e6e80db] - **encoder**: add PCAEncoder support in gnes buster image (*raccoonliukai*)
- [16fa80bd] - **tests**: add unittest for PCAEncoder (*raccoonliukai*)
- [5a745b1e] - **tests**: add unittest for EncoderService and IndexerService (*raccoonliukai*)
- [a0fec684] - **service**: logging elapsed time and body type change (*hanhxiao*)
- [57cc95ff] - **encoder**: add quantizer (*Jem*)
- [00e6280d] - **score_fn**: use numpy for score fn (*hanhxiao*)
- [201c27e7] - **cli**: add `--sorted_response` as cli argument (*hanhxiao*)
- [81b21093] - **index**: move sort logic to base (*hanhxiao*)
- [a2d55dda] - **index**: move sort logic out to base (*hanhxiao*)
- [674a9da2] - **encoder**: add lab video model (*Jem*)
- [50a944b6] - **encoder**: add yt8m feature extractor (*Jem*)
- [f908f381] - **score_fn**: make score_fn as a TrainableBase (*hanhxiao*)
- [14c7e522] - **score_fn**: make score_fn dumpable (*hanhxiao*)
- [0b78798d] - **score_fn**: add score_fn as a new module (*hanhxiao*)
- [da56544f] - **encoder**: add PCAEncoder for incremental pca training (*raccoonliukai*)
- [97bb6de2] - **lab encoder**: add vggish for audio (*Jem*)
- [8cdcb7e8] - **chunk scorer**: add offset divergence (*Jem*)

1.18.2 Bug fixes

- [d404b8a7] - **tests**: use lowercase for true (*raccoonliu*)
- [bb9bbe9d] - **tests**: modify EncoderService unittest (*raccoonliukai*)
- [cd53a24b] - **indexer**: fix numpy indexer (*hanhxiao*)
- [d70e877e] - **shot-detector**: fix case of only one shot in video (*felix*)
- [e631d396] - **service**: indexer service return empty when no chunk (*hanhxiao*)
- [67b211da] - **encoder**: remove image resize from TFInceptionEncoder (*raccoonliukai*)
- [40849abc] - **indexer**: fix `is_sorted` in response flush away the request (*hanhxiao*)
- [ab819387] - **ffmpeg**: use tempfile as input instead of pipe (*felix*)
- [a8d2acfd] - **service**: `is_input_list` is false when query (*Jem*)
- [ba21c4e7] - **service**: fix bug for doc type in encoder (*Larry Yan*)
- [a4658250] - **scorer**: fix np float conversion (*hanhxiao*)
- [2d6c70fc] - **indexer**: fix `vec np.concat` (*hanhxiao*)

- [2ba135db] - **indexer**: fix empty chunks indexing (*hanhxiao*)
- [40dd1d5a] - **encoder**: fix `embed_chunks_in_docs` function (*hanhxiao*)
- [d94329b3] - **preprocess**: fix offset in sentence splitter (*hanhxiao*)

1.18.3 Code Refactoring

- [a8e87d9f] - **service**: minimize event loop, move handling to handler (*hanhxiao*)
- [06aab813] - **grpc-client**: implement async client via multi-threaded (*felix*)
- [35fa3ba4] - **pb**: remove unused field (*hanhxiao*)
- [6bbfc993] - **score_fn**: rename score functions (*hanhxiao*)
- [e9feaa61] - **score_fn**: use `post_init` instead of property (*hanhxiao*)
- [f406f8f0] - **score_fn**: move `normalize_fn` and `score_fn` to the init (*hanhxiao*)

1.18.4 Other Improvements

- [c988c7dd] - remove sklearn from base dep (*felix*)
- [2fd8dab4] - minor fix on the styling (*Han Xiao*)
- [066d6e99] - **pb2**: do not reformat pb2 generated python file (*hanhxiao*)
- [1bbc4358] - **service**: revert encoder service (*hanhxiao*)
- [b97b718f] - **changelog**: update change log to v0.0.36 (*hanhxiao*)

1.19 Release Note (v0.0.36)

Release time: 2019-08-30 17:32:23

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, felix,

1.19.1 New Features

- [07534f89] - **score**: improve score explain for better interpretability (*hanhxiao*)
- [32d815d7] - **filesystem**: keep doc meta info (*felix*)
- [92fc3d8c] - **scale_video**: keep ratio of video frame (*felix*)

1.19.2 Bug fixes

- [f1402f50] - **cli**: fix cli chanel close (*hanhxiao*)
- [b140cca9] - **service**: fix exception when no chunks (*hanhxiao*)
- [cee99a63] - **logger**: change the color semantic for loglevel (*hanhxiao*)
- [4efea726] - **service**: raise except when empty chunk (*hanhxiao*)
- [31bffeb7] - **preprocessor**: add `min_len` to split preprocessor (*hanhxiao*)

- [7b16354a] - **style**: fix style issues (*hanhxiao*)
- [c6183960] - **service**: fix training logic in encoderservice (*hanhxiao*)
- [5828d20a] - **preprocessor**: fix SentSplitPreprocessor (*hanhxiao*)
- [522c5a4e] - **preprocessor**: rename SentSplitPreprocessor (*hanhxiao*)
- [030d6c66] - **setup**: fix path in setup script (*hanhxiao*)
- [3818c9a3] - **test**: fix router tests (*hanhxiao*)
- [9d03441e] - **proto**: regenerate pb2 (*hanhxiao*)
- [f49f9a5b] - **indexer**: fix parsing in DictIndexer (*hanhxiao*)
- [0215c6bf] - **ffmpeg**: fix issue for start and duration argument position (*felix*)
- [a735a719] - **service**: log error in base service (*hanhxiao*)
- [3263e96c] - **service**: move py_import from service manager to base service (*hanhxiao*)
- [990c879d] - **client**: fix client progress bar, http (*hanhxiao*)
- [d02cd757] - **router**: respect num_part when set (*hanhxiao*)
- [a76a4604] - **ffmpeg-video**: fix bug for scaling videos to stdout (*felix*)

1.19.3 Code Refactoring

- [42e7c13b] - **indexer**: separate score logic and index logic (*hanhxiao*)
- [0c6f4851] - **preprocessor**: use io utils in audio and gif (*Jem*)
- [bae75b8c] - **router**: separate router and scoring logics (*hanhxiao*)
- [c3ebb93a] - **proto**: refactor offset nd (*Jem*)
- [e3bbbd9b] - **shot_detector**: update ffmpeg api (*felix*)
- [10cef54e] - **ffmpeg**: refactor ffmpeg again (*felix*)

1.19.4 Unit Test and CICD

- [1e9ef35c] - **pipeline**: test pipeline load from yaml (*hanhxiao*)
- [5b7c9f19] - **pipeline**: add unit test for pipeline encoder (*hanhxiao*)
- [bef2bf9c] - **indexer**: add unit test for dict indexer as service (*hanhxiao*)
- [620cf3bd] - **ffmpeg**: add unittest for ffmpeg api (*felix*)

1.19.5 Other Improvements

- [c83448b5] - **license**: add license header to frontend (*hanhxiao*)
- [04deea3a] - **license**: remove irrelevant packages from license (*hanhxiao*)
- [10f4bedb] - fix error (*felix*)
- [4aa997c3] - **changelog**: update change log to v0.0.35 (*hanhxiao*)

1.20 Release Note (v0.0.35)

Release time: 2019-08-26 18:15:02

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem,

1.20.1 New Features

- [b4444cc0] - **encoder**: separate pooling as an indep. encoder (*hanhxiao*)
- [ce0e65ae] - **helper**: batching decorator supports tuple (*hanhxiao*)
- [a584c7e5] - **helper**: add `as_numpy_array` decorator (*hanhxiao*)

1.20.2 Bug fixes

- [ff7926d8] - **encoder**: fix eager execution (*hanhxiao*)
- [27a7ca8c] - **cli**: add a small jitter to prevent div zero (*hanhxiao*)

1.20.3 Code Refactoring

- [39561000] - **base**: add `on_gpu` to replace `use_cuda` (*hanhxiao*)
- [928574cd] - **encoder**: replace `gpt` and `elmo` with `transformer` (*hanhxiao*)
- [52538276] - **encoder**: no for loop in torch encoder now (*Jem*)
- [7493af97] - **preprocessor**: add `init`, change signature (*Jem*)

1.20.4 Other Improvements

- [20249afb] - **changelog**: update change log to v0.0.34 (*hanhxiao*)

1.21 Release Note (v0.0.34)

Release time: 2019-08-23 19:00:27

We'd like to thank all contributors for this new release! In particular, hanhxiao,

1.21.1 Other Improvements

- [79a8effd] - **changelog**: update change log to v0.0.33 (*hanhxiao*)

1.22 Release Note (v0.0.34)

Release time: 2019-08-23 18:44:34

We'd like to thank all contributors for this new release! In particular, hanhxiao,

1.22.1 Other Improvements

- [79a8effd] - **changelog**: update change log to v0.0.33 (*hanhxiao*)

1.23 Release Note (v0.0.33)

Release time: 2019-08-23 18:34:28

We'd like to thank all contributors for this new release! In particular, Jem, hanhxiao, felix, raccoonliukai,

1.23.1 New Features

- [9d488e3f] - **client**: add progress bar and speed metric to cli (*hanhxiao*)
- [829d148c] - **scale_video**: scale video use ffmpeg (*felix*)
- [bc2e441d] - **compose**: add minimum http server without flask dep (*hanhxiao*)
- [d420f348] - **video_preprocessor**: add edge detect for shotdetect (*raccoonliukai*)

1.23.2 Bug fixes

- [6cfbda9d] - **preprocessor**: move dependency into function (*Jem*)
- [0e88b77a] - **frontend**: fix request_id zero is none (*hanhxiao*)
- [ca28ecb9] - **video_preprocessor**: use rgb as standard color (*raccoonliukai*)
- [5b5feb0b] - **video_preprocessor**: use dict update (*raccoonliukai*)
- [47721b1c] - **video_preprocessor**: remove custom canny threshold (*raccoonliukai*)
- [16aaa777] - **video_preprocessor**: modify inaccurate names (*raccoonliukai*)
- [dfb54b62] - **video_preprocessor**: Remove incorrect comments (*raccoonliukai*)

1.23.3 Code Refactoring

- [3d63fac6] - **proto**: request_id is now an integer (*hanhxiao*)
- [4497d765] - **shotdetector**: use updated ffmpeg api to capture frames from videos (*felix*)
- [dbc06a85] - **ffmpeg**: refactor ffmpeg to read frames, vides and gif (*felix*)
- [a7b12cb6] - **preprocessor**: add gif chunk prep (*Jem*)
- [559a9971] - **compose**: unify flask and http handler (*hanhxiao*)

1.23.4 Documentation

- [a2801d5c] - link gnes hub tutorial to readme (*hanhxiao*)

1.23.5 Other Improvements

- [02f70a03] - fix bug (*felix*)
- [c970bec3] - **changelog**: update change log to v0.0.32 (*hanhxiao*)

1.24 Release Note (v0.0.32)

Release time: 2019-08-21 17:23:13

We'd like to thank all contributors for this new release! In particular, hanhxiao, Han Xiao, Jem,

1.24.1 New Features

- [38567b00] - **indexer**: add preprocessor and lvdb for storing gif (*Jem*)
- [35465e85] - **base**: later import module now override the earlier ones (*hanhxiao*)

1.24.2 Bug fixes

- [5c2b60a4] - remove `target_image_size` (*hanhxiao*)
- [944b8c09] - **ci**: fix unit tests for modules (*hanhxiao*)

1.24.3 Code Refactoring

- [5f1ca000] - fixing the imports of all base module (*hanhxiao*)
- [0d1bd4e2] - **preprocessor**: remove unnecessary init (*Han Xiao*)

1.24.4 Unit Test and CICD

- [3820db6a] - **encoder**: rename BasePytorchEncoder to TorchvisionEncoder (*hanhxiao*)

1.24.5 Other Improvements

- [3147a1d5] - **changelog**: update change log to v0.0.31 (*hanhxiao*)

1.25 Release Note (v0.0.31)

Release time: 2019-08-20 14:01:04

We'd like to thank all contributors for this new release! In particular, hanhxiao,

1.25.1 New Features

- [f7beae7b] - **cli**: add `py_path` in parser to load external modules (*hanhxiao*)

1.25.2 Other Improvements

- [ec1eb787] - **release**: fix duplicate release notes (*hanhxiao*)
- [447756d5] - **changelog**: update change log to v0.0.30 (*hanhxiao*)

1.26 Release Note (v0 . 0 . 30)

Release time: 2019-08-19 14:13:03

We'd like to thank all contributors for this new release! In particular, hanhxiao,

1.26.1 New Features

- [7b5cc86a] - **contrib**: no need to give module name in advance (*hanhxiao*)

1.26.2 Bug fixes

- [5f69c781] - **contrib**: allowing dump for contributed module (*hanhxiao*)

1.26.3 Other Improvements

- [565ef569] - **changelog**: update change log to v0.0.29 (*hanhxiao*)

1.27 Release Note (v0 . 0 . 29)

Release time: 2019-08-16 15:40:31

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem,

1.27.1 Bug fixes

- [2f905168] - **setup**: fix encoding problem in setup.py (*hanhxiao*)
- [469bc51d] - **ci**: fix cd pipeline (*hanhxiao*)

1.27.2 Code Refactoring

- [66d020bd] - **base**: component renamed to components (*hanhxiao*)
- [3a2b85b6] - **proto**: assign doc id in request generator (*Jem*)

1.27.3 Documentation

- [b854c697] - **readme**: fix description on images (*hanhxiao*)

1.27.4 Unit Test and CICD

- [f1658c92] - **docker**: docker image tag-alpine as default tag (*hanhxiao*)
- [8885512f] - **docker**: clean up the space after docker build (*hanhxiao*)

1.27.5 Other Improvements

- [00ca6919] - **changelog**: update change log to v0.0.28 (*hanhxiao*)

1.28 Release Note (v0.0.28)

Release time: 2019-08-14 20:54:26

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, raccoonliukai, Larry Yan,

1.28.1 New Features

- [0133905c] - **client**: add a client for benchmarking and testing (*hanhxiao*)
- [732f2e64] - **encoder**: add pytorch transformers support in text encoder (*raccoonliukai*)
- [6aab48c8] - **docker**: add buster image with minimum dependencies (*hanhxiao*)
- [da1bbc0d] - **docker**: add alpine image with minimum dependencies (*hanhxiao*)

1.28.2 Bug fixes

- [315bd16a] - **doc sum router**: use meta info instead of doc id to do doc sum (*Jem*)
- [c9e92722] - **encoder**: use offline model in ci-base for pytorch transformer (*raccoonliukai*)
- [d7b42d39] - **setup**: remove unused dependencies (*hanhxiao*)
- [5b8acf7c] - **test**: fix routes assert in tests (*hanhxiao*)
- [5fedf6df] - **encoder**: fix unused variable (*raccoonliukai*)
- [df616463] - **cli**: remove unnecessary argument (*hanhxiao*)
- [fd76aa79] - **request_generator**: send index request in index mode (*Jem*)
- [64163cb1] - **batching**: enable to process three dimension output in batching (*Jem*)
- [415456d6] - **preprocessor**: fix bug (*Larry Yan*)
- [c150ad59] - **preprocessor**: modify ffmpeg video pre add video cutting method (*Larry Yan*)
- [b0f22d04] - **audio preprocessor**: filter audio with zero length (*Jem*)
- [d1cfa539] - **preprocessor**: modify ffmpeg video preprocessor (*Larry Yan*)

1.28.3 Documentation

- [e11a920e] - **readme**: add image explain to readme (*hanhxiao*)

1.28.4 Unit Test and CICD

- [a8700801] - **drone**: add self hosted drone (*hanhxiao*)
- [079d0a1a] - **docker**: move docker-build to a more controllable cd process (*hanhxiao*)

1.28.5 Other Improvements

- [5257259f] - add kai liu to core maintainers (*hanhxiao*)
- [8d318204] - **changelog**: update change log to v0.0.27 (*hanhxiao*)

1.29 Release Note (v0.0.27)

Release time: 2019-08-09 19:51:57

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, Larry Yan, raccoonliu, Han Xiao, raccoonliukai,

1.29.1 New Features

- [55126f2b] - **grpc**: add a general purpose grpc service (*hanhxiao*)
- [23c6e68a] - **reduce router**: add chunk and doc reduce routers for audio (*Jem*)
- [6d3d2b4c] - **cli**: use ServiceManager as default service runner (*hanhxiao*)
- [ccfd474a] - **service**: add ServiceManager and enable parallel services in one container (*hanhxiao*)
- [63f9173f] - **service**: enabling the choose of thread or process as the backend (*hanhxiao*)
- [2647b848] - **audio**: add preprocess and mfcc encoder for audio (*Jem*)
- [208e1937] - **audio**: add preprocess and mfcc encoder for audio, update protobuf (*Jem*)
- [77a2ea42] - **parser**: improve yaml_path parsing (*hanhxiao*)
- [762535ca] - **vlad**: add vlad and enable multiple chunks and frames (*Jem*)
- [64e948d4] - **encoder**: add onnxruntime for image encoder (*raccoonliukai*)
- [f03e6fc2] - **encoder**: add onnxruntime support for image encoder (*raccoonliukai*)

1.29.2 Bug fixes

- [5ae46d61] - **composer**: rename grpcfrontend to frontend (*hanhxiao*)
- [4cb83383] - **audio**: restrict max length for mfcc encoding (*Jem*)
- [e516646f] - **grpc**: add max_message_size to the argparser (*hanhxiao*)
- [0493e6fc] - **encoder**: fix netvlad (*Larry Yan*)
- [e773aa33] - **service manager**: fix nonetype for service manager (*Jem*)
- [d5d15d7f] - **compose**: fix a bug in doc_reduce_test (*hanhxiao*)
- [6856cb0a] - **compose**: copy args on every request (*hanhxiao*)
- [f80e8c03] - **cli**: set default num_part is None (*hanhxiao*)

- [7031fe20] - **preprocessor**: add random sampling to ffmpeg (*Larry Yan*)
- [fd37e6d9] - **encoder**: fix bug caused by batching in inception_mixture (*Larry Yan*)
- [2191b27b] - **composer**: fix yaml generation (*hanhxiao*)
- [e5fefcee] - **encoder**: fix batching in encoder (*hanhxiao*)
- [e35e3b3c] - **composer**: fix composer router generation logic (*hanhxiao*)
- [7300e055] - **preprocessor**: quantity improvement (*Larry Yan*)
- [47efaba4] - **unittest**: fix unittest of video preprocessor 2 (*Larry Yan*)
- [a6efb4af] - **unittest**: fix unittest of video preprocessor (*Larry Yan*)
- [dd1216bb] - **unittest**: fix unittest for video processor (*Larry Yan*)
- [8e6dc4c6] - **encoder**: add func for preprocessor (*Larry Yan*)
- [2b21dc5a] - **encoder**: fix unused import and variable (*raccoonliu*)
- [fd576915] - **test**: fix import (*Han Xiao*)
- [a0fdad36] - **test**: fix broken code (*Han Xiao*)
- [8ca07a74] - **test**: fix img_process_for_test (*Han Xiao*)
- [7c16fb8b] - **preprocessor**: fix bug in ffmpeg.py and add more func to helper (*Larry Yan*)
- [e6a37119] - **preprocessor**: fix bug in params in ffmpeg (*Larry Yan*)
- [f8d2abe5] - **preprocessor**: fix bug in ffmpeg (*Larry Yan*)
- [67610f86] - **preprocessor**: add more method for cutting video (*Larry Yan*)

1.29.3 Code Refactoring

- [8516096d] - **grpc**: moving zmqclient to client module (*hanhxiao*)
- [5e3409e1] - **grpc**: hide private class inside gRPCfrontend (*hanhxiao*)
- [6407cc8d] - **yaml**: remove useless default yaml config (*hanhxiao*)
- [c1e406ae] - **onnx**: move batch_size to class attribute (*Han Xiao*)

1.29.4 Unit Test and CICD

- [5503dbe7] - skip joint indexer test as it is not even used (*hanhxiao*)
- [8ab101ca] - add mergify for auto merging (*hanhxiao*)
- [203d1697] - **chore**: exclude chore job from ci pipeline (*hanhxiao*)
- [24f9fd1c] - fix yaml_path missing in the test (*hanhxiao*)
- [23a83a40] - simplify yaml naming (*hanhxiao*)

1.29.5 Other Improvements

- [e8e3b9b9] - **changelog**: update change log to v0.0.26 (*hanhxiao*)

1.30 Release Note (v0.0.26)

Release time: 2019-08-02 18:18:45

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, Larry Yan,

1.30.1 New Features

- [d0b2ef0b] - **composer**: more interaction for gnes board (*hanhxiao*)
- [9c33dc66] - **router**: allow consecutive mapping and reducing ops (*hanhxiao*)

1.30.2 Bug fixes

- [fc5026da] - **board**: improve gnes board 500 message (*hanhxiao*)
- [823bdeda] - **test**: fix grpc gentle shutdown (*hanhxiao*)
- [f6a801f7] - **test**: fix preprocessor building for image test (*hanhxiao*)
- [50fdc041] - **base**: fix ref to CompositionalTrainableBase (*hanhxiao*)
- [54a931c7] - **test**: fix test images by removing mac stuff (*hanhxiao*)
- [14cdfabe] - **sliding window**: fix the boundary (*Jem*)
- [46b5c94e] - **encoder**: fix name for video encoder (*Larry Yan*)
- [15eb50b4] - **encoder**: fix params in basevideo encoder (*Larry Yan*)
- [5b0fe7c6] - **preprocessor**: fix FFmpegVideoSegmentor (*Larry Yan*)
- [d6a46fa6] - **encoder**: fix import path for mixture encoder (*Larry Yan*)
- [17779676] - **encoder**: fix mixture encoder (*Larry Yan*)
- [95f03c56] - **encoder**: fix bug in video mixture encoder (*Larry Yan*)
- [3fd1c06] - **encoder**: fix mixture (*Larry Yan*)
- [67991533] - **encoder**: add netvlad and netfv register class (*Larry Yan*)
- [92500f0f] - **encoder**: add netvlad and netfv (*Larry Yan*)

1.30.3 Code Refactoring

- [c430ef64] - **base**: better batch_size control (*hanhxiao*)
- [58217d8c] - **base**: moving is_trained to class attribute (*hanhxiao*)
- [7126d496] - **preprocessor**: separate resize logic from the unary preprocessor (*hanhxiao*)
- [52f87c7f] - **base**: make pipelineencoder more general and allow pipelinepreprocessor (*hanhxiao*)

1.30.4 Documentation

- [3ab3723e] - **tutorial**: fix image and code layout (*hanhxiao*)

1.30.5 Other Improvements

- [635ba37f] - **changelog**: update change log to v0.0.25 (*hanhxiao*)

1.31 Release Note (v0.0.25)

Release time: 2019-07-26 19:45:21

We'd like to thank all contributors for this new release! In particular, hanhxiao, felix, Larry Yan, Jem, Han Xiao, Felix,

1.31.1 New Features

- [66aec9c9] - **grpc**: add StreamCall and decouple send and receive (*hanhxiao*)
- [5697441b] - **indexer**: consider offset relevance at query time (*Jem*)
- [04c9c745] - **image preprocessor**: calculate offsetnd for each chunk (*Jem*)
- [b34a765a] - **compose**: add interactive mode of GNES board using Flask (*hanhxiao*)
- [5876c15e] - **base**: support loading external modules from py and yaml (*hanhxiao*)

1.31.2 Bug fixes

- [a20672d3] - **preprocessor**: add logging in helper module (*felix*)
- [f9500c1f] - **protobuffer**: add doc_type as func argument in RequestGenerator (*felix*)
- [1c3bb01a] - **service**: fix bug in doc_type name in indexer service (*Larry Yan*)
- [d834f578] - **service**: add doc type to req generator (*Larry Yan*)
- [80e234e1] - **service**: fix bug in req Generator add doc_type (*Larry Yan*)
- [5743e258] - **indexer**: fix bug in indexer service (*Larry Yan*)
- [11dde2bf] - **encoder**: fix bug in tf inception (*Larry Yan*)
- [ded92c57] - **indexer**: fix bug for indexer service dealing with empty doc (*Larry Yan*)
- [1dff06f1] - **encoder**: fix bug for encoder service dealing with empty doc (*Larry Yan*)
- [7e43d5a2] - **preprocessor**: fix ffmpeg to deal with broken image (*Larry Yan*)
- [83ebaced] - **preprocessor**: move import imagehash to inside (*hanhxiao*)
- [7c669a70] - **test**: rename the yaml test file (*hanhxiao*)
- [2cc26342] - **compose**: change textarea font to monospace (*hanhxiao*)
- [e644e391] - **encoder**: fix gpu limitation in inception (*Larry Yan*)
- [89d8b70c] - **grpc**: fix bug in RequestGenerator query (*Larry Yan*)
- [c52c2cc6] - **base**: fix gnes_config mixed in kwargs (*hanhxiao*)
- [68c15fac] - **base**: fix redundant warning in pipeline encoder (*hanhxiao*)
- [aadeeefb] - **composer**: fix composer state machine (*hanhxiao*)
- [c0bffe6c] - **indexer**: normalize weight (*Jem*)

- [2c696483] - **indexer**: fix weight in indexer call (*Larry Yan*)
- [139a02d9] - **compose**: fix compose bug of pub-sub rule, duplicate yaml_path (*hanhxiao*)
- [649ed131] - **encoder**: add normalize option in cvae encoder (*Larry Yan*)
- [eb487799] - **encoder**: fix tf scope error in cvae encoder (*Larry Yan*)
- [ab6c88cc] - **encoder**: fix error in cvae encoder (*Larry Yan*)
- [a4b883ac] - **indexer**: add drop raw bytes option to leveldb (*Larry Yan*)
- [4b52bcba] - **grpc**: fix grpc plugin path (*Larry Yan*)
- [d3fbbcac] - **weighting**: add simple normalization to chunk search (*Jem*)
- [08a9a4e3] - **grpc**: fix grpc service (*Larry Yan*)
- [6e6bbf83] - **grpc**: add auto-gen grpc code (*Larry Yan*)
- [b89d8fa2] - **grpc**: add stream index and train in proto (*Larry Yan*)
- [15cd7e58] - **base**: fix dump and load on compositional encoder (*hanhxiao*)
- [bab48919] - **encoder**: fix tf inception (*Larry Yan*)
- [973672ef] - **encoder**: fix bug for encoder bin load (*Larry Yan*)
- [1bef3971] - **setup**: fix setup script (*hanhxiao*)
- [67fb5766] - **compose**: fix argparser (*hanhxiao*)
- [63c4515f] - **compose**: accept parser argument only (*hanhxiao*)
- [887d89cc] - **release**: ask BOT_URL before releasing (*hanhxiao*)

1.31.3 Code Refactoring

- [9973f600] - **preprocessor**: rename singleton to unary (*hanhxiao*)
- [a1a2b020] - **proto**: refactor request stream call (*hanhxiao*)

1.31.4 Documentation

- [c853e3da] - **tutorial**: fix svg size (*hanhxiao*)
- [04cccdcd] - **tutorial**: fix svg path (*hanhxiao*)
- [8927cd4f] - **tutorial**: add yaml explain (*hanhxiao*)
- [5b52ce4c] - fix doc path (*hanhxiao*)
- [45751e1f] - **readme**: add quick start for readme (*hanhxiao*)
- [73891ecc] - **readme**: add install guide to readme and contribution guide (*hanhxiao*)

1.31.5 Unit Test and CI/CD

- [6ff3079b] - **unittest**: skip all os environ test (*hanhxiao*)
- [816fa043] - **unittest**: skip blocked test (*hanhxiao*)
- [79a9c106] - **unittest**: run test in verbose mode (*hanhxiao*)

- [83276f90] - **torchvision**: install torchvision dependency to enable tests (*hanhxiao*)
- [499682ce] - **base**: add unit test for load a dumped pipeline from yaml (*hanhxiao*)
- [26a7ad18] - **composer**: add unit test for flask (*hanhxiao*)
- [87ec1fd2] - **base**: move module delete to teardown (*hanhxiao*)
- [479b183d] - **compose**: skip unit test (*hanhxiao*)

1.31.6 Other Improvements

- [c30f39cc] - ... (*felix*)
- [2d5654c0] - **license**: add license (*hanhxiao*)
- [d3347910] - reformat code and optimize import (*hanhxiao*)
- [71491ffb] - **changelog**: update change log to v0.0.24 (*hanhxiao*)

1.32 Release Note (v0.0.24)

Release time: 2019-07-19 18:18:46

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, Larry Yan, felix,

1.32.1 New Features

- [9f6c0524] - **fasterrcnn**: add the original image to chunk list (*Jem*)
- [abb0841c] - **encoder**: add convolution variational autoencoder (*Larry Yan*)

1.32.2 Bug fixes

- [1b526832] - **base**: fix dump yaml kwargs (*hanhxiao*)
- [086f3cea] - **base**: fix ump instance (*hanhxiao*)
- [12dfde42] - **base**: move name setting to trainable base (*hanhxiao*)
- [16f1a497] - **base**: move set config to metaclass (*hanhxiao*)
- [b97acd6c] - **base**: fix duplicate warning (*hanhxiao*)
- [991e4425] - **base**: fix duplicate load and init from yaml (*hanhxiao*)
- [69a486e5] - **compose**: fix import (*hanhxiao*)
- [4977aa3c] - **vector indexer**: reorder relevance and chunk weight (*Jem*)
- [2448411d] - **encoder**: modify CVAE (*Larry Yan*)
- [b4bf0bf8] - **indexer**: add path check for dir and file (*hanhxiao*)
- [92f36c33] - **fasterrcnn**: handle imgs with 0 chunk (*Jem*)
- [a1329913] - **fasterrcnn**: fix bug for gpu (*Jem*)
- [38eca0ce] - **grpc**: change grpc client message size limit (*felix*)
- [3836020a] - **preprocessor**: fix preprocessor service handler function name error (*felix*)

- [599a3c3d] - **compose**: fix composer logic (*hanhxiao*)
- [7f3b2fb5] - **release**: fix git tag version (*hanhxiao*)

1.32.3 Code Refactoring

- [9bbb3c05] - **compose**: move compose template to resources (*hanhxiao*)
- [a4e153d7] - **base**: remove dump path and reorganize work dir (*hanhxiao*)

1.32.4 Unit Test and CICD

- [e088ea9c] - **drone**: turn off profiling in ci (*hanhxiao*)
- [33a570b9] - **drone**: remove pylint for faster ci (*hanhxiao*)
- [51eafac7] - **indexer**: fix data path in unit test (*hanhxiao*)

1.32.5 Other Improvements

- [43ef4108] - **git**: add tmp to ignore (*hanhxiao*)
- [44b1a0c9] - fix unittest (*felix*)
- [984a9a2d] - **changelog**: update change log to v0.0.23 (*hanhxiao*)

1.33 Release Note (v0.0.23)

Release time: 2019-07-17 18:28:08

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, felix, Larry Yan, Han Xiao,

1.33.1 New Features

- [cb4d9cf2] - **release**: add auto release and keep change log (*hanhxiao*)
- [c667d874] - **image_preprocessor**: add fasterRCNN (*Jem*)
- [a6c2975b] - **composer**: improve the gnes board with cards (*hanhxiao*)
- [6ec4233d] - **composer**: add swarm and bash generator (*hanhxiao*)
- [08aa30f4] - **composer**: add shell script generator (*hanhxiao*)
- [033a4b9c] - **composer**: add composer and mermaid renderer (*hanhxiao*)

1.33.2 Bug fixes

- [2b7c3f18] - **compose**: resolve unclosed file warning (*hanhxiao*)
- [8030feb2] - **compose**: fix router logic in compose (*hanhxiao*)
- [736f6053] - **gnesboard**: fix cdn (*hanhxiao*)
- [fb07ff02] - **doc_reducer_router**: fix reduce error (*felix*)

- [a7236308] - **image encoder**: define use_cuda variable via args (*felix*)
- [cba5e190] - **image_encoder**: enable batching encoding (*felix*)
- [3423ec83] - **composer**: add compose api to api.py (*hanhxiao*)
- [70ba3fca] - **composer**: in bash mode always run job in background (*hanhxiao*)
- [054981ce] - **composer**: fix gnes board naming (*hanhxiao*)
- [743ec3b0] - **composer**: fix unit test and add tear down (*hanhxiao*)
- [64aef413] - **composer**: fix styling according to codacy (*hanhxiao*)
- [dca4b03b] - **service**: fix bug grpc (*Larry Yan*)
- [09e68da2] - **service**: fix grpc server size limit (*Larry Yan*)
- [3da8da19] - **encoder**: rm un-used import in inception (*Larry Yan*)
- [8780a4da] - bugs for integrated test (*Jem*)
- [38fff782] - **preprocessor**: move cv2 dep to pic_weight (*Han Xiao*)
- [37155bba] - **preprocessor-video**: move sklearn dep to apply (*Han Xiao*)
- [1f6a06a2] - **encoder**: rm tf inception unittest (*Larry Yan*)
- [eaffbbff] - **encoder**: register tf inception in **init** (*Larry Yan*)
- [d0099b79] - **encoder**: add necessary code from tf (*Larry Yan*)
- [b480774a] - **encoder**: add inception tf (*Larry Yan*)

1.33.3 Documentation

- [54276c6a] - **readme**: improve readme image and structure (*hanhxiao*)

1.33.4 Unit Test and CICD

- [1dcfdfa7] - **docker-image**: optimize docker file (*felix*)
- [bda562d1] - **drone**: auto release with cron job (*hanhxiao*)

1.33.5 Other Improvements

- [0c737a94] - **release**: revert back master check (*hanhxiao*)
- [7b04697c] - **changelog**: revert the change log to empty (*hanhxiao*)
- [d02f320d] - revert docker file (*felix*)

CHAPTER 2

Tutorials

Warning: Tutorial is still under construction. Stay tuned! Meanwhile, we sincerely welcome you to contribute your own learning experience / case study with GNES!

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

g

gnes, 76
gnes.base, 19
gnes.cli, 21
gnes.cli.api, 20
gnes.cli.parser, 21
gnes.client, 23
gnes.client.base, 21
gnes.client.cli, 22
gnes.client.http, 22
gnes.client.stream, 23
gnes.component, 75
gnes.composer, 24
gnes.composer.base, 23
gnes.composer.flask, 24
gnes.composer.http, 24
gnes.encoder, 37
gnes.encoder.audio, 26
gnes.encoder.audio.mfcc, 25
gnes.encoder.audio.vggish, 26
gnes.encoder.audio.vggish_cores, 25
gnes.encoder.audio.vggish_cores.vggish_params, 25
gnes.encoder.audio.vggish_cores.vggish_postprocess, 25
gnes.encoder.base, 36
gnes.encoder.image, 28
gnes.encoder.image.cvae, 27
gnes.encoder.image.cvae_cores, 26
gnes.encoder.image.inception, 27
gnes.encoder.image.inception_cores, 26
gnes.encoder.image.onnx, 27
gnes.encoder.image.torchvision, 28
gnes.encoder.numeric, 31
gnes.encoder.numeric.hash, 28
gnes.encoder.numeric.pca, 28
gnes.encoder.numeric.pooling, 29
gnes.encoder.numeric.pq, 29
gnes.encoder.numeric.quantizer, 30
gnes.encoder.numeric.standarder, 30
gnes.encoder.numeric.tf_pq, 30
gnes.encoder.numeric.vlad, 31
gnes.encoder.text, 33
gnes.encoder.text.bert, 31
gnes.encoder.text.char, 32
gnes.encoder.text.flair, 32
gnes.encoder.text.w2v, 32
gnes.encoder.video, 36
gnes.encoder.video.incep_mixture, 34
gnes.encoder.video.inception, 34
gnes.encoder.video.mixture_core, 33
gnes.encoder.video.yt8m_feature_extractor, 35
gnes.encoder.video.yt8m_feature_extractor_cores, 33
gnes.encoder.video.yt8m_model, 35
gnes.flow, 39
gnes.flow.base, 37
gnes.flow.helper, 39
gnes.helper, 75
gnes.indexer, 53
gnes.indexer.base, 50
gnes.indexer.chunk, 48
gnes.indexer.chunk.annoy, 45
gnes.indexer.chunk.faiss, 46
gnes.indexer.chunk.helper, 46
gnes.indexer.chunk.numpy, 48
gnes.indexer.doc, 50
gnes.indexer.doc.dict, 48
gnes.indexer.doc.filesys, 49
gnes.indexer.doc.leveldb, 49
gnes.indexer.doc.rocksdb, 50
gnes.preprocessor, 64
gnes.preprocessor.audio, 56
gnes.preprocessor.audio.vggish_example, 55
gnes.preprocessor.audio.vggish_example_helper, 55

- gnes.preprocessor.audio.vggish_example_helper.mel_features,
53
- gnes.preprocessor.base, 61
- gnes.preprocessor.helper, 62
- gnes.preprocessor.image, 57
- gnes.preprocessor.image.resize, 56
- gnes.preprocessor.image.segmentation,
56
- gnes.preprocessor.image.sliding_window,
56
- gnes.preprocessor.io_utils, 58
- gnes.preprocessor.io_utils.ffmpeg, 57
- gnes.preprocessor.io_utils.gif, 57
- gnes.preprocessor.io_utils.helper, 57
- gnes.preprocessor.io_utils.video, 58
- gnes.preprocessor.io_utils.webp, 58
- gnes.preprocessor.text, 59
- gnes.preprocessor.text.split, 58
- gnes.preprocessor.video, 61
- gnes.preprocessor.video.ffmpeg, 59
- gnes.preprocessor.video.frame_select,
60
- gnes.preprocessor.video.shot_detector,
60
- gnes.preprocessor.video.video_decoder,
60
- gnes.preprocessor.video.video_encoder,
61
- gnes.proto, 64
- gnes.proto.gnes_pb2, 64
- gnes.proto.gnes_pb2_grpc, 64
- gnes.router, 68
- gnes.router.base, 65
- gnes.router.map, 66
- gnes.router.reduce, 67
- gnes.score_fn, 72
- gnes.score_fn.base, 69
- gnes.score_fn.chunk, 70
- gnes.score_fn.doc, 71
- gnes.score_fn.normalize, 71
- gnes.service, 75
- gnes.service.base, 72
- gnes.service.encoder, 74
- gnes.service.frontend, 74
- gnes.service.grpc, 74
- gnes.service.indexer, 75
- gnes.service.preprocessor, 75
- gnes.service.router, 75
- gnes.uuid, 76

A

- abs (*gnes.score_fn.base.ScoreOps* attribute), 69
- acquire() (*gnes.helper.FileLock* method), 76
- ActionNoYes (class in *gnes.cli.parser*), 21
- add() (*gnes.flow.Flow* method), 40
- add() (*gnes.indexer.base.BaseChunkIndexer* method), 51
- add() (*gnes.indexer.base.BaseChunkIndexerHelper* method), 51
- add() (*gnes.indexer.base.BaseDocIndexer* method), 52
- add() (*gnes.indexer.base.BaseIndexer* method), 52
- add() (*gnes.indexer.base.JointIndexer* method), 52
- add() (*gnes.indexer.chunk.annoy.AnnoyIndexer* method), 45
- add() (*gnes.indexer.chunk.faiss.FaissIndexer* method), 46
- add() (*gnes.indexer.chunk.helper.DictKeyIndexer* method), 46
- add() (*gnes.indexer.chunk.helper.ListKeyIndexer* method), 47
- add() (*gnes.indexer.chunk.helper.ListNumpyKeyIndexer* method), 47
- add() (*gnes.indexer.chunk.helper.NumpyKeyIndexer* method), 47
- add() (*gnes.indexer.chunk.numpy.NumpyIndexer* method), 48
- add() (*gnes.indexer.doc.dict.DictIndexer* method), 48
- add() (*gnes.indexer.doc.filesys.DirectoryIndexer* method), 49
- add() (*gnes.indexer.doc.leveldb.AsyncLVDBIndexer* method), 49
- add() (*gnes.indexer.doc.leveldb.LVDBIndexer* method), 49
- add() (*gnes.indexer.doc.rocksdb.RocksDBIndexer* method), 50
- add_comp() (*gnes.composer.base.YamlComposer* method), 23
- add_encoder() (*gnes.flow.Flow* method), 40
- add_frontend() (*gnes.flow.Flow* method), 41
- add_GnesRPCServicer_to_server() (in module *gnes.proto.gnes_pb2_grpc*), 64
- add_indexer() (*gnes.flow.Flow* method), 41
- add_layer() (*gnes.composer.base.YamlComposer* method), 23
- add_modules() (*gnes.helper.PathImporter* static method), 76
- add_preprocessor() (*gnes.flow.Flow* method), 41
- add_route() (in module *gnes.proto*), 65
- add_router() (*gnes.flow.Flow* method), 41
- add_version() (in module *gnes.proto*), 65
- ALWAYS_ONE (*gnes.service.base.ReduceOp* attribute), 73
- AnnoyIndexer (class in *gnes.indexer.chunk.annoy*), 45
- append() (*gnes.composer.base.YamlComposer.Layer* method), 23
- apply() (*gnes.preprocessor.audio.vggish_example.VggishPreprocessor* method), 55
- apply() (*gnes.preprocessor.base.BasePreprocessor* method), 61
- apply() (*gnes.preprocessor.base.PipelinePreprocessor* method), 62
- apply() (*gnes.preprocessor.base.RawChunkPreprocessor* method), 62
- apply() (*gnes.preprocessor.base.UnaryPreprocessor* method), 62
- apply() (*gnes.preprocessor.image.resize.ResizeChunkPreprocessor* method), 56
- apply() (*gnes.preprocessor.image.segmentation.SegmentPreprocessor* method), 56
- apply() (*gnes.preprocessor.text.split.SentSplitPreprocessor* method), 58
- apply() (*gnes.preprocessor.video.ffmpeg.FFmpegPreprocessor* method), 59
- apply() (*gnes.preprocessor.video.ffmpeg.FFmpegVideoSegmentor* method), 59
- apply() (*gnes.preprocessor.video.frame_select.FrameSelectPreprocessor* method), 60
- apply() (*gnes.preprocessor.video.shot_detector.ShotDetectorPreprocessor* method), 60

- [apply \(\) \(gnes.preprocessor.video.video_decoder.VideoDecoderPreprocessor method\), 60](#)
[apply \(\) \(gnes.preprocessor.video.video_encoder.VideoEncoderPreprocessor method\), 61](#)
[apply \(\) \(gnes.router.base.BaseEmbedReduceRouter method\), 65](#)
[apply \(\) \(gnes.router.base.BaseMapRouter method\), 65](#)
[apply \(\) \(gnes.router.base.BaseReduceRouter method\), 65](#)
[apply \(\) \(gnes.router.base.BaseRouter method\), 66](#)
[apply \(\) \(gnes.router.base.BaseTopkReduceRouter method\), 66](#)
[apply \(\) \(gnes.router.base.PipelineRouter method\), 66](#)
[apply \(\) \(gnes.router.map.BlockRouter method\), 66](#)
[apply \(\) \(gnes.router.map.DocBatchRouter method\), 67](#)
[apply \(\) \(gnes.router.map.PublishRouter method\), 67](#)
[apply \(\) \(gnes.router.reduce.DocFillReducer method\), 68](#)
[array2blob \(\) \(in module gnes.proto\), 64](#)
[AsyncLVDBIndexer \(class in gnes.indexer.doc.leveldb\), 49](#)
[avg \(gnes.score_fn.base.ScoreOps attribute\), 69](#)
[AvgEmbedRouter \(class in gnes.router.reduce\), 67](#)
- ## B
- [BaseAudioEncoder \(class in gnes.encoder.base\), 36](#)
[BaseAudioPreprocessor \(class in gnes.preprocessor.base\), 61](#)
[BaseBinaryEncoder \(class in gnes.encoder.base\), 36](#)
[BaseChunkIndexer \(class in gnes.indexer.base\), 50](#)
[BaseChunkIndexerHelper \(class in gnes.indexer.base\), 51](#)
[BaseDocIndexer \(class in gnes.indexer.base\), 51](#)
[BaseEmbedReduceRouter \(class in gnes.router.base\), 65](#)
[BaseEncoder \(class in gnes.encoder.base\), 36](#)
[BaseImageEncoder \(class in gnes.encoder.base\), 36](#)
[BaseImagePreprocessor \(class in gnes.preprocessor.base\), 61](#)
[BaseIndexer \(class in gnes.indexer.base\), 52](#)
[BaseIndexFlow \(class in gnes.flow.base\), 37](#)
[BaseMapRouter \(class in gnes.router.base\), 65](#)
[BaseNumericEncoder \(class in gnes.encoder.base\), 36](#)
[BaseONNXImageEncoder \(class in gnes.encoder.image.onnx\), 27](#)
[BasePreprocessor \(class in gnes.preprocessor.base\), 61](#)
[BaseQueryFlow \(class in gnes.flow.base\), 38](#)
[BaseReduceRouter \(class in gnes.router.base\), 65](#)
[BaseRouter \(class in gnes.router.base\), 65](#)
[BaseScoreFn \(class in gnes.score_fn.base\), 69](#)
[BaseService \(class in gnes.service.base\), 72](#)
[BaseTextPreprocessor \(class in gnes.preprocessor.base\), 62](#)
[BaseTopkReduceRouter \(class in gnes.router.base\), 66](#)
[BaseVideoEncoder \(class in gnes.encoder.base\), 37](#)
[BaseVideoPreprocessor \(class in gnes.preprocessor.base\), 62](#)
[batch_iterator \(\) \(in module gnes.helper\), 75](#)
[batch_size \(gnes.encoder.audio.mfcc.MfccEncoder attribute\), 25](#)
[batch_size \(gnes.encoder.image.cvae.CVAEEncoder attribute\), 27](#)
[batch_size \(gnes.encoder.image.inception.TFInceptionEncoder attribute\), 27](#)
[batch_size \(gnes.encoder.image.onnx.BaseONNXImageEncoder attribute\), 27](#)
[batch_size \(gnes.encoder.image.torchvision.TorchvisionEncoder attribute\), 28](#)
[batch_size \(gnes.encoder.numeric.hash.HashEncoder attribute\), 28](#)
[batch_size \(gnes.encoder.numeric.pca.PCAEncoder attribute\), 28](#)
[batch_size \(gnes.encoder.numeric.pca.PCALocalEncoder attribute\), 29](#)
[batch_size \(gnes.encoder.numeric.pq.PQEncoder attribute\), 29](#)
[batch_size \(gnes.encoder.numeric.quantizer.QuantizerEncoder attribute\), 30](#)
[batch_size \(gnes.encoder.numeric.standarder.StandarderEncoder attribute\), 30](#)
[batch_size \(gnes.encoder.numeric.tf_pq.TFPQEncoder attribute\), 30](#)
[batch_size \(gnes.encoder.numeric.vlad.VladEncoder attribute\), 31](#)
[batch_size \(gnes.encoder.video.incep_mixture.IncepMixtureEncoder attribute\), 34](#)
[batch_size \(gnes.encoder.video.inception.InceptionVideoEncoder attribute\), 34](#)
[batch_size \(gnes.encoder.video.yt8m_feature_extractor.YouTube8MFeatureExtractor attribute\), 35](#)
[batch_size \(gnes.encoder.video.yt8m_model.YouTube8MEncoder attribute\), 35](#)
[batching \(\) \(in module gnes.helper\), 75](#)
[BertEncoder \(class in gnes.encoder.text.bert\), 31](#)
[BertEncoderServer \(class in gnes.encoder.text.bert\), 31](#)
[BertEncoderWithServer \(class in gnes.encoder.text.bert\), 32](#)
[BetterEnum \(class in gnes.service.base\), 72](#)
[blob2array \(\) \(in module gnes.proto\), 64](#)
[block_descriptor \(\) \(in module gnes.preprocessor.helper\), 62](#)
[BlockMessage, 72](#)

BlockRouter (class in *gnes.router.map*), 66
 BM25ChunkScoreFn (class in *gnes.score_fn.chunk*), 70
 build() (*gnes.flow.Flow* method), 41
 build_all() (*gnes.composer.base.YamlComposer* method), 23
 build_dockerswarm() (*gnes.composer.base.YamlComposer* static method), 23
 build_html() (*gnes.composer.base.YamlComposer* static method), 24
 build_kubernetes() (*gnes.composer.base.YamlComposer* static method), 24
 build_layers() (*gnes.composer.base.YamlComposer* method), 24
 build_mermaid() (*gnes.composer.base.YamlComposer* static method), 24
 build_required() (in module *gnes.flow.helper*), 39
 build_shell() (*gnes.composer.base.YamlComposer* static method), 24
 build_socket() (in module *gnes.service.base*), 74
 BuildLevel (class in *gnes.flow.helper*), 39
 bytes_generator (*gnes.client.cli.CLIClient* attribute), 22

C

call() (*gnes.client.base.GrpcClient* method), 22
 Call() (*gnes.proto.gnes_pb2_grpc.GnesRPCServicer* method), 64
 call_hooks() (*gnes.service.base.MessageHandler* method), 73
 call_routes() (*gnes.client.base.ResponseHandler* method), 22
 call_routes() (*gnes.service.base.MessageHandler* method), 73
 call_routes_send_back() (*gnes.service.base.MessageHandler* method), 73
 canny_edge() (in module *gnes.preprocessor.helper*), 62
 capacity (*gnes.indexer.chunk.helper.NumpyKeyIndexer* attribute), 47
 capture_frames() (in module *gnes.preprocessor.io_utils.gif*), 57
 capture_frames() (in module *gnes.preprocessor.io_utils.video*), 58
 CharEmbeddingEncoder (class in *gnes.encoder.text.char*), 32
 check_fields() (*gnes.composer.base.YamlComposer* method), 24
 check_motion() (in module *gnes.preprocessor.helper*), 62
 Chunk2DocTopkReducer (class in *gnes.router.reduce*), 67
 ChunkTopkReducer (class in *gnes.router.reduce*), 67
 CLIClient (class in *gnes.client.cli*), 22
 client() (in module *gnes.cli.api*), 20
 close() (*gnes.base.CompositionalTrainableBase* method), 20
 close() (*gnes.base.TrainableBase* method), 19
 close() (*gnes.client.base.GrpcClient* method), 22
 close() (*gnes.client.base.ZmqClient* method), 22
 close() (*gnes.client.stream.StreamingClient* method), 23
 close() (*gnes.client.stream.SyncClient* method), 23
 close() (*gnes.encoder.numeric.tf_pq.TFPQEncoder* method), 30
 close() (*gnes.encoder.text.bert.BertEncoder* method), 31
 close() (*gnes.encoder.text.bert.BertEncoderServer* method), 31
 close() (*gnes.flow.Flow* method), 41
 close() (*gnes.indexer.doc.leveldb.AsyncLVDBIndexer* method), 49
 close() (*gnes.indexer.doc.leveldb.LVDBIndexer* method), 50
 close() (*gnes.indexer.doc.rocksdb.RocksDBIndexer* method), 50
 close() (*gnes.service.base.BaseService* method), 72
 close() (*gnes.service.grpc.GRPCService* method), 74
 CombinedScoreFn (class in *gnes.score_fn.base*), 69
 comp2args (*gnes.composer.base.YamlComposer* attribute), 24
 comp2file (*gnes.composer.base.YamlComposer* attribute), 24
 compare_descriptor() (in module *gnes.preprocessor.helper*), 63
 compare_ecr() (in module *gnes.preprocessor.helper*), 63
 compile_args() (in module *gnes.preprocessor.io_utils.ffmpeg*), 57
 ComponentNotLoad, 72
 components (*gnes.base.CompositionalTrainableBase* attribute), 20
 components (*gnes.indexer.base.JointIndexer* attribute), 52
 compose() (in module *gnes.cli.api*), 20
 CompositionalTrainableBase (class in *gnes.base*), 20
 compute_descriptor() (in module *gnes.preprocessor.helper*), 63
 CONCAT (*gnes.service.base.ReduceOp* attribute), 73
 ConcatEmbedRouter (class in *gnes.router.reduce*), 68
 ConcurrentService (class in *gnes.service.base*), 72
 CoordChunkScoreFn (class in *gnes.score_fn.chunk*),

- 70
- CoordDocScoreFn (class in *gnes.score_fn.doc*), 71
- CVAEEncoder (class in *gnes.encoder.image.cvae*), 27
- ## D
- default_host (*gnes.service.base.BaseService* attribute), 72
- default_values (*gnes.composer.base.YamlComposer.Layer* attribute), 23
- detect_peak_boundary () (in module *gnes.preprocessor.helper*), 63
- detect_shots () (*gnes.preprocessor.video.shot_detector.ShotDetectorPreprocessor* method), 60
- DictIndexer (class in *gnes.indexer.doc.dict*), 48
- DictKeyIndexer (class in *gnes.indexer.chunk.helper*), 46
- DirectoryIndexer (class in *gnes.indexer.doc.filesys*), 49
- doc_type (*gnes.preprocessor.base.BaseAudioPreprocessor* attribute), 61
- doc_type (*gnes.preprocessor.base.BaseImagePreprocessor* attribute), 61
- doc_type (*gnes.preprocessor.base.BasePreprocessor* attribute), 61
- doc_type (*gnes.preprocessor.base.BaseTextPreprocessor* attribute), 62
- doc_type (*gnes.preprocessor.base.BaseVideoPreprocessor* attribute), 62
- DocBatchRouter (class in *gnes.router.map*), 67
- DocFillReducer (class in *gnes.router.reduce*), 68
- DocTopkReducer (class in *gnes.router.reduce*), 68
- dump () (*gnes.base.TrainableBase* method), 19
- dump () (*gnes.service.base.BaseService* method), 72
- dump_full_path (*gnes.base.TrainableBase* attribute), 19
- dump_yaml () (*gnes.base.TrainableBase* method), 19
- duplicate_rm_hash () (*gnes.preprocessor.video.ffmpeg.FFmpegPreprocessor* method), 59
- ## E
- embed_chunks_in_docs () (*gnes.service.encoder.EncoderService* method), 74
- EMPTY (*gnes.flow.helper.BuildLevel* attribute), 39
- encode () (*gnes.encoder.audio.mfcc.MfccEncoder* method), 25
- encode () (*gnes.encoder.audio.vggish.VggishEncoder* method), 26
- encode () (*gnes.encoder.base.BaseAudioEncoder* method), 36
- encode () (*gnes.encoder.base.BaseBinaryEncoder* method), 36
- encode () (*gnes.encoder.base.BaseEncoder* method), 36
- encode () (*gnes.encoder.base.BaseImageEncoder* method), 36
- encode () (*gnes.encoder.base.BaseNumericEncoder* method), 36
- encode () (*gnes.encoder.base.BaseTextEncoder* method), 37
- encode () (*gnes.encoder.base.BaseVideoEncoder* method), 37
- encode () (*gnes.encoder.base.PipelineEncoder* method), 37
- encode () (*gnes.encoder.image.cvae.CVAEEncoder* method), 27
- encode () (*gnes.encoder.image.inception.TFInceptionEncoder* method), 27
- encode () (*gnes.encoder.image.onnx.BaseONNXImageEncoder* method), 27
- encode () (*gnes.encoder.image.torchvision.TorchvisionEncoder* method), 28
- encode () (*gnes.encoder.numeric.hash.HashEncoder* method), 28
- encode () (*gnes.encoder.numeric.pca.PCAEncoder* method), 28
- encode () (*gnes.encoder.numeric.pca.PCALocalEncoder* method), 29
- encode () (*gnes.encoder.numeric.pooling.PoolingEncoder* method), 29
- encode () (*gnes.encoder.numeric.pq.PQEncoder* method), 29
- encode () (*gnes.encoder.numeric.quantizer.QuantizerEncoder* method), 30
- encode () (*gnes.encoder.numeric.standarder.StandarderEncoder* method), 30
- encode () (*gnes.encoder.numeric.tf_pq.TFPQEncoder* method), 30
- encode () (*gnes.encoder.numeric.vlad.VladEncoder* method), 31
- encode () (*gnes.encoder.text.bert.BertEncoder* method), 31
- encode () (*gnes.encoder.text.bert.BertEncoderWithServer* method), 32
- encode () (*gnes.encoder.text.char.CharEmbeddingEncoder* method), 32
- encode () (*gnes.encoder.text.flair.FlairEncoder* method), 32
- encode () (*gnes.encoder.text.w2v.Word2VecEncoder* method), 32
- encode () (*gnes.encoder.video.incep_mixture.IncepMixtureEncoder* method), 34
- encode () (*gnes.encoder.video.inception.InceptionVideoEncoder* method), 34
- encode () (*gnes.encoder.video.yt8m_feature_extractor.YouTube8MFeatureExtractor* method), 35

- [encode\(\) \(gnes.encoder.video.yt8m_model.YouTube8MEncoder.first_available_gpu\(\) \(in module method\), 35](#)
[encode\(\) \(in module gnes.cli.api\), 20](#)
[encode_video\(\) \(in module gnes.preprocessor.io_utils.gif\), 57](#)
[encode_video\(\) \(in module gnes.preprocessor.io_utils.video\), 58](#)
[encode_video\(\) \(in module gnes.preprocessor.io_utils.webp\), 58](#)
[Encoder \(gnes.flow.helper.Service attribute\), 39](#)
[EncoderService \(class in gnes.service.encoder\), 74](#)
[EventLoopEnd, 72](#)
[extract_frame_size\(\) \(in module gnes.preprocessor.io_utils.ffmpeg\), 57](#)
- ## F
- [faiss_index\(\) \(gnes.encoder.numeric.vlad.VladEncoder.method\), 31](#)
[FaissIndexer \(class in gnes.indexer.chunk.faiss\), 46](#)
[FFmpegPreprocessor \(class in gnes.preprocessor.video.ffmpeg\), 59](#)
[FFmpegVideoSegmentor \(class in gnes.preprocessor.video.ffmpeg\), 59](#)
[FileLock \(class in gnes.helper\), 75](#)
[FlairEncoder \(class in gnes.encoder.text.flair\), 32](#)
[Flow \(class in gnes.flow\), 39](#)
[FlowBuildLevelMismatch, 39](#)
[FlowIncompleteError, 39](#)
[FlowMissingNode, 39](#)
[FlowTopologyError, 39](#)
[frame\(\) \(in module gnes.preprocessor.audio.vggish_example_helper.features\), 53](#)
[FrameSelectPreprocessor \(class in gnes.preprocessor.video.frame_select\), 60](#)
[from_string \(gnes.service.base.BetterEnum attribute\), 72](#)
[from_yaml\(\) \(gnes.base.CompositionalTrainableBase class method\), 20](#)
[from_yaml\(\) \(gnes.base.TrainableBase class method\), 20](#)
[Frontend \(gnes.flow.Flow attribute\), 40](#)
[Frontend \(gnes.flow.helper.Service attribute\), 39](#)
[frontend\(\) \(in module gnes.cli.api\), 20](#)
[FrontendService \(class in gnes.service.frontend\), 74](#)
- ## G
- [get_all_subarea\(\) \(in module gnes.preprocessor.helper\), 63](#)
[get_audio\(\) \(in module gnes.preprocessor.helper\), 63](#)
[get_component_name \(gnes.composer.base.YamlComposer.Layer attribute\), 23](#)
[get_first_available_gpu\(\) \(in module gnes.helper\), 76](#)
[get_gif\(\) \(in module gnes.preprocessor.helper\), 63](#)
[get_key\(\) \(gnes.router.base.BaseTopkReduceRouter method\), 66](#)
[get_key\(\) \(gnes.router.reduce.Chunk2DocTopkReducer method\), 67](#)
[get_key\(\) \(gnes.router.reduce.ChunkTopkReducer method\), 67](#)
[get_key\(\) \(gnes.router.reduce.DocTopkReducer method\), 68](#)
[get_main_parser\(\) \(in module gnes.cli.parser\), 21](#)
[get_media_meta\(\) \(in module gnes.preprocessor.io_utils.ffmpeg\), 57](#)
[get_optimal_sample_size\(\) \(in module gnes.helper\), 75](#)
[get_perm\(\) \(in module gnes.helper\), 75](#)
[get_random_ipc\(\) \(in module gnes.service.base\), 74](#)
[get_sys_info\(\) \(in module gnes.helper\), 75](#)
[get_unary_score\(\) \(in module gnes.score_fn.base\), 70](#)
[get_value\(\) \(gnes.composer.base.YamlComposer.Layer static method\), 23](#)
[get_video_length\(\) \(in module gnes.preprocessor.helper\), 63](#)
[get_video_length_from_raw\(\) \(in module gnes.preprocessor.helper\), 63](#)
[GifChunkPreprocessor \(class in gnes.preprocessor.video.ffmpeg\), 59](#)
[gnes \(module\), 76](#)
[gnes.base \(module\), 19](#)
[gnes.cli \(module\), 21](#)
[gnes.cli.api \(module\), 20](#)
[gnes.cli.parser \(module\), 21](#)
[gnes.client \(module\), 23](#)
[gnes.client.base \(module\), 21](#)
[gnes.client.cli \(module\), 22](#)
[gnes.client.http \(module\), 22](#)
[gnes.client.stream \(module\), 23](#)
[gnes.component \(module\), 75](#)
[gnes.composer \(module\), 24](#)
[gnes.composer.base \(module\), 23](#)
[gnes.composer.flask \(module\), 24](#)
[gnes.composer.http \(module\), 24](#)
[gnes.encoder \(module\), 37](#)
[gnes.encoder.audio \(module\), 26](#)
[gnes.encoder.audio.mfcc \(module\), 25](#)
[gnes.encoder.audio.vggish \(module\), 26](#)
[gnes.encoder.audio.vggish_cores \(module\), 25](#)
[gnes.encoder.audio.vggish_cores.vggish_params \(module\), 25](#)
[gnes.encoder.audio.vggish_cores.vggish_postprocess](#)

(*module*), 25

`gnes.encoder.base` (*module*), 36

`gnes.encoder.image` (*module*), 28

`gnes.encoder.image.cvae` (*module*), 27

`gnes.encoder.image.cvae_cores` (*module*), 26

`gnes.encoder.image.inception` (*module*), 27

`gnes.encoder.image.inception_cores` (*module*), 26

`gnes.encoder.image.onnx` (*module*), 27

`gnes.encoder.image.torchvision` (*module*), 28

`gnes.encoder.numeric` (*module*), 31

`gnes.encoder.numeric.hash` (*module*), 28

`gnes.encoder.numeric.pca` (*module*), 28

`gnes.encoder.numeric.pooling` (*module*), 29

`gnes.encoder.numeric.pq` (*module*), 29

`gnes.encoder.numeric.quantizer` (*module*), 30

`gnes.encoder.numeric.standarder` (*module*), 30

`gnes.encoder.numeric.tf_pq` (*module*), 30

`gnes.encoder.numeric.vlad` (*module*), 31

`gnes.encoder.text` (*module*), 33

`gnes.encoder.text.bert` (*module*), 31

`gnes.encoder.text.char` (*module*), 32

`gnes.encoder.text.flair` (*module*), 32

`gnes.encoder.text.w2v` (*module*), 32

`gnes.encoder.video` (*module*), 36

`gnes.encoder.video.incep_mixture` (*module*), 34

`gnes.encoder.video.inception` (*module*), 34

`gnes.encoder.video.mixture_core` (*module*), 33

`gnes.encoder.video.yt8m_feature_extractor` (*module*), 35

`gnes.encoder.video.yt8m_feature_extractor_core` (*module*), 33

`gnes.encoder.video.yt8m_model` (*module*), 35

`gnes.flow` (*module*), 39

`gnes.flow.base` (*module*), 37

`gnes.flow.helper` (*module*), 39

`gnes.helper` (*module*), 75

`gnes.indexer` (*module*), 53

`gnes.indexer.base` (*module*), 50

`gnes.indexer.chunk` (*module*), 48

`gnes.indexer.chunk.annoy` (*module*), 45

`gnes.indexer.chunk.faiss` (*module*), 46

`gnes.indexer.chunk.helper` (*module*), 46

`gnes.indexer.chunk.numpy` (*module*), 48

`gnes.indexer.doc` (*module*), 50

`gnes.indexer.doc.dict` (*module*), 48

`gnes.indexer.doc.filesys` (*module*), 49

`gnes.indexer.doc.leveldb` (*module*), 49

`gnes.indexer.doc.rocksdb` (*module*), 50

`gnes.preprocessor` (*module*), 64

`gnes.preprocessor.audio` (*module*), 56

`gnes.preprocessor.audio.vggish_example` (*module*), 55

`gnes.preprocessor.audio.vggish_example_helper` (*module*), 55

`gnes.preprocessor.audio.vggish_example_helper.mel` (*module*), 53

`gnes.preprocessor.base` (*module*), 61

`gnes.preprocessor.helper` (*module*), 62

`gnes.preprocessor.image` (*module*), 57

`gnes.preprocessor.image.resize` (*module*), 56

`gnes.preprocessor.image.segmentation` (*module*), 56

`gnes.preprocessor.image.sliding_window` (*module*), 56

`gnes.preprocessor.io_utils` (*module*), 58

`gnes.preprocessor.io_utils.ffmpeg` (*module*), 57

`gnes.preprocessor.io_utils.gif` (*module*), 57

`gnes.preprocessor.io_utils.helper` (*module*), 57

`gnes.preprocessor.io_utils.video` (*module*), 58

`gnes.preprocessor.io_utils.webp` (*module*), 58

`gnes.preprocessor.text` (*module*), 59

`gnes.preprocessor.text.split` (*module*), 58

`gnes.preprocessor.video` (*module*), 61

`gnes.preprocessor.video.ffmpeg` (*module*), 59

`gnes.preprocessor.video.frame_select` (*module*), 60

`gnes.preprocessor.video.shot_detector` (*module*), 60

`gnes.preprocessor.video.video_decoder` (*module*), 60

`gnes.preprocessor.video.video_encoder` (*module*), 61

`gnes.proto` (*module*), 64

`gnes.proto.gnes_pb2` (*module*), 64

`gnes.proto.gnes_pb2_grpc` (*module*), 64

`gnes.router` (*module*), 68

`gnes.router.base` (*module*), 65

`gnes.router.map` (*module*), 66

`gnes.router.reduce` (*module*), 67

`gnes.score_fn` (*module*), 72

`gnes.score_fn.base` (*module*), 69

`gnes.score_fn.chunk` (*module*), 70

`gnes.score_fn.doc` (*module*), 71

`gnes.score_fn.normalize` (*module*), 71

`gnes.service` (*module*), 75

gnes.service.base (module), 72
 gnes.service.encoder (module), 74
 gnes.service.frontend (module), 74
 gnes.service.grpc (module), 74
 gnes.service.indexer (module), 75
 gnes.service.preprocessor (module), 75
 gnes.service.router (module), 75
 gnes.uuid (module), 76
 GnesRPCServicer (class in *gnes.proto.gnes_pb2_grpc*), 64
 GnesRPCStub (class in *gnes.proto.gnes_pb2_grpc*), 64
 GRAPH (*gnes.flow.helper.BuildLevel* attribute), 39
 grpc () (in module *gnes.cli.api*), 20
 GrpcClient (class in *gnes.client.base*), 21
 GRPCService (class in *gnes.service.grpc*), 74

H

handler (*gnes.client.base.GrpcClient* attribute), 22
 handler (*gnes.client.stream.StreamingClient* attribute), 23
 handler (*gnes.client.stream.SyncClient* attribute), 23
 handler (*gnes.service.base.BaseService* attribute), 72
 handler (*gnes.service.encoder.EncoderService* attribute), 74
 handler (*gnes.service.grpc.GRPCService* attribute), 74
 handler (*gnes.service.indexer.IndexerService* attribute), 75
 handler (*gnes.service.preprocessor.PreprocessorService* attribute), 75
 handler (*gnes.service.router.RouterService* attribute), 75
 hash () (*gnes.encoder.numeric.hash.HashEncoder* method), 28
 HashEncoder (class in *gnes.encoder.numeric.hash*), 28
 healthcheck () (in module *gnes.cli.api*), 20
 hertz_to_mel () (in module *gnes.preprocessor.audio.vggish_example_helper*), 53
 hsv_histogram () (in module *gnes.preprocessor.helper*), 63
 HttpClient (class in *gnes.client.http*), 22

I

IncepMixtureEncoder (class in *gnes.encoder.video.incep_mixture*), 34
 InceptionVideoEncoder (class in *gnes.encoder.video.inception*), 34
 index () (*gnes.client.cli.CLIClient* method), 22
 index () (*gnes.flow.Flow* method), 41
 Index () (*gnes.proto.gnes_pb2_grpc.GnesRPCServicer* method), 64
 index () (*gnes.proto.RequestGenerator* static method), 64
 index () (in module *gnes.cli.api*), 21

Indexer (*gnes.flow.helper.Service* attribute), 39
 IndexerService (class in *gnes.service.indexer*), 75
 is_bind (*gnes.service.base.SocketType* attribute), 74
 is_block (*gnes.service.base.ParallelType* attribute), 73
 is_heto_single_component (*gnes.composer.base.YamlComposer.Layer* attribute), 23
 is_homo_multi_component (*gnes.composer.base.YamlComposer.Layer* attribute), 23
 is_homogenous (*gnes.composer.base.YamlComposer.Layer* attribute), 23
 is_locked (*gnes.helper.FileLock* attribute), 76
 is_pipeline (*gnes.base.CompositionalTrainableBase* attribute), 20
 is_push (*gnes.service.base.ParallelType* attribute), 73
 is_single_component (*gnes.composer.base.YamlComposer.Layer* attribute), 23
 is_trained (*gnes.base.CompositionalTrainableBase* attribute), 20
 is_trained (*gnes.encoder.text.bert.BertEncoder* attribute), 31
 is_trained (*gnes.encoder.text.bert.BertEncoderServer* attribute), 31
 is_trained (*gnes.encoder.text.char.CharEmbeddingEncoder* attribute), 32
 is_trained (*gnes.encoder.text.flair.FlairEncoder* attribute), 32
 is_trained (*gnes.encoder.text.w2v.Word2VecEncoder* attribute), 33
 is_trained (*gnes.preprocessor.base.UnaryPreprocessor* attribute), 62

J

join () (*gnes.service.base.ServiceManager* method), 73
 join () (*gnes.service.frontend.FrontendService* method), 74
 JointIndexer (class in *gnes.indexer.base*), 52

K

kmeans_algo () (in module *gnes.preprocessor.helper*), 63
 kmeans_pred () (*gnes.encoder.numeric.vlad.VladEncoder* method), 31
 kmeans_train () (*gnes.encoder.numeric.vlad.VladEncoder* method), 31
 kwargs_to_cmd_args () (in module *gnes.preprocessor.io_utils.helper*), 57

L

ListKeyIndexer (class in *gnes.indexer.chunk.helper*), 47

ListNumpyKeyIndexer (class in *gnes.indexer.chunk.helper*), 47
 ln (*gnes.score_fn.base.ScoreOps* attribute), 69
 ln1p (*gnes.score_fn.base.ScoreOps* attribute), 69
 ln2p (*gnes.score_fn.base.ScoreOps* attribute), 69
 load() (*gnes.base.TrainableBase* static method), 20
 load_contrib_module() (in module *gnes.helper*), 75
 load_model() (*gnes.service.base.BaseService* method), 72
 load_yaml() (*gnes.base.TrainableBase* class method), 20
 log (*gnes.score_fn.base.ScoreOps* attribute), 69
 log1p (*gnes.score_fn.base.ScoreOps* attribute), 69
 log2p (*gnes.score_fn.base.ScoreOps* attribute), 69
 log_mel_spectrogram() (in module *gnes.preprocessor.audio.vggish_example_helper.mel_features*), 53
 LVDBIndexer (class in *gnes.indexer.doc.leveladb*), 49

M

main() (in module *gnes.cli*), 21
 masked_reduce_max() (*gnes.encoder.numeric.pooling.PoolingEncoder* method), 29
 masked_reduce_mean() (*gnes.encoder.numeric.pooling.PoolingEncoder* method), 29
 max (*gnes.score_fn.base.ScoreOps* attribute), 69
 MessageHandler (class in *gnes.service.base*), 72
 MfccEncoder (class in *gnes.encoder.audio.mfcc*), 25
 min (*gnes.score_fn.base.ScoreOps* attribute), 69
 minus_mask() (*gnes.encoder.numeric.pooling.PoolingEncoder* method), 29
 ModifierScoreFn (class in *gnes.score_fn.base*), 69
 motion_algo() (in module *gnes.preprocessor.helper*), 63
 mul_mask() (*gnes.encoder.numeric.pooling.PoolingEncoder* method), 29
 multiply (*gnes.score_fn.base.ScoreOps* attribute), 69

N

new_score() (*gnes.score_fn.base.BaseScoreFn* method), 69
 none (*gnes.score_fn.base.ScoreOps* attribute), 69
 Normalizer1 (class in *gnes.score_fn.normalize*), 71
 Normalizer2 (class in *gnes.score_fn.normalize*), 71
 Normalizer3 (class in *gnes.score_fn.normalize*), 71
 Normalizer4 (class in *gnes.score_fn.normalize*), 71
 Normalizer5 (class in *gnes.score_fn.normalize*), 72
 num_chunks (*gnes.indexer.base.BaseChunkIndexer* attribute), 51
 num_chunks (*gnes.indexer.base.BaseIndexer* attribute), 52
 num_chunks_in_doc() (*gnes.indexer.base.BaseChunkIndexer* method), 51
 num_docs (*gnes.indexer.base.BaseChunkIndexer* attribute), 51
 num_docs (*gnes.indexer.base.BaseIndexer* attribute), 52
 NumpyIndexer (class in *gnes.indexer.chunk.numpy*), 48
 NumpyKeyIndexer (class in *gnes.indexer.chunk.helper*), 47

P

PAIR_BIND (*gnes.service.base.SocketType* attribute), 73
 PAIR_CONNECT (*gnes.service.base.SocketType* attribute), 73
 PAIR_CONNECT (*gnes.service.base.SocketType* attribute), 74
 ParallelType (class in *gnes.service.base*), 73
 parse_arg() (in module *gnes.helper*), 75
 parse_http_data() (in module *gnes.composer.base*), 24
 parse_media_details() (in module *gnes.preprocessor.io_utils.ffmpeg*), 57
 PathImporter (class in *gnes.helper*), 76
 PCAEncoder (class in *gnes.encoder.numeric.pca*), 28
 PCALocalEncoder (class in *gnes.encoder.numeric.pca*), 29
 periodic_hann() (in module *gnes.preprocessor.audio.vggish_example_helper.mel_features*), 53
 phash_descriptor() (in module *gnes.preprocessor.helper*), 63
 pool_weight() (*gnes.preprocessor.video.ffmpeg.FFmpegPreprocessor* static method), 59
 PipelineEncoder (class in *gnes.encoder.base*), 37
 PipelinePreprocessor (class in *gnes.preprocessor.base*), 62
 PipelineRouter (class in *gnes.router.base*), 66
 PoolingEncoder (class in *gnes.encoder.numeric.pooling*), 29
 post_init() (*gnes.base.TrainableBase* method), 20
 post_init() (*gnes.encoder.audio.vggish.VggishEncoder* method), 26
 post_init() (*gnes.encoder.image.cvae.CVAEEncoder* method), 27
 post_init() (*gnes.encoder.image.inception.TFInceptionEncoder* method), 27
 post_init() (*gnes.encoder.image.onnx.BaseONNXImageEncoder* method), 27
 post_init() (*gnes.encoder.image.torchvision.TorchvisionEncoder* method), 28
 post_init() (*gnes.encoder.numeric.pca.PCAEncoder* method), 29

post_init () (*gnes.encoder.numeric.pooling.PoolingEncoder* *method*), 29
 post_init () (*gnes.encoder.numeric.standarder.StandarderEncoder* *method*), 30
 post_init () (*gnes.encoder.numeric.tf_pq.TFPQEncoder* *method*), 30
 post_init () (*gnes.encoder.text.bert.BertEncoder* *method*), 31
 post_init () (*gnes.encoder.text.bert.BertEncoderServer* *method*), 31
 post_init () (*gnes.encoder.text.flair.FlairEncoder* *method*), 32
 post_init () (*gnes.encoder.text.w2v.Word2VecEncoder* *method*), 33
 post_init () (*gnes.encoder.video.incep_mixture.InceptionMixtureEncoder* *method*), 34
 post_init () (*gnes.encoder.video.inception.InceptionVideoEncoder* *method*), 34
 post_init () (*gnes.encoder.video.yt8m_feature_extractor.YouTube8MFeatureExtractor* *method*), 35
 post_init () (*gnes.encoder.video.yt8m_model.YouTube8MModel* *method*), 35
 post_init () (*gnes.indexer.chunk.annoy.AnnoyIndexer* *method*), 45
 post_init () (*gnes.indexer.chunk.faiss.FaissIndexer* *method*), 46
 post_init () (*gnes.indexer.doc.leveldb.AsyncLVDBIndexer* *method*), 49
 post_init () (*gnes.indexer.doc.leveldb.LVDBIndexer* *method*), 50
 post_init () (*gnes.indexer.doc.rocksdb.RocksDBIndexer* *method*), 50
 post_init () (*gnes.preprocessor.image.segmentation.SegmentPreprocessor* *method*), 56
 post_init () (*gnes.router.base.BaseTopkReduceRouter* *method*), 66
 post_init () (*gnes.score_fn.base.CombinedScoreFn* *method*), 69
 post_init () (*gnes.score_fn.base.ModifierScoreFn* *method*), 69
 post_init () (*gnes.service.base.BaseService* *method*), 72
 post_init () (*gnes.service.encoder.EncoderService* *method*), 74
 post_init () (*gnes.service.grpc.GRPCService* *method*), 74
 post_init () (*gnes.service.indexer.IndexerService* *method*), 75
 post_init () (*gnes.service.preprocessor.PreprocessorService* *method*), 75
 post_init () (*gnes.service.router.RouterService* *method*), 75
 postprocess () (*gnes.encoder.audio.vggish_cores.vggish_postprocess* *method*), 25
 postprocessor (class in *gnes.encoder.audio.vggish_cores.vggish_postprocess*), 25
 PQEncoder (class in *gnes.encoder.numeric.pq*), 29
 pre_init () (*gnes.base.TrainableBase* class method), 20
 pre_init () (*gnes.encoder.numeric.tf_pq.TFPQEncoder* class method), 30
 pred_kmeans () (*gnes.encoder.numeric.hash.HashEncoder* *method*), 28
 preprocess () (in module *gnes.cli.api*), 21
 Preprocessor (*gnes.flow.helper.Service* attribute), 39
 PreprocessorService (class in *gnes.service.preprocessor*), 75
 PreprocessorService (in module *gnes.preprocessor.io_utils.ffmpeg*), 57
 ProgressBar (class in *gnes.helper*), 75
 ProgressBar (class in *gnes.client.cli*), 22
 YouTube8MFeatureExtractor (in module *gnes.helper*), 76
 PUB_BIND (*gnes.service.base.SocketType* attribute), 73
 PUB_BLOCK (*gnes.service.base.ParallelType* attribute), 73
 PUB_CONNECT (*gnes.service.base.SocketType* attribute), 73
 PUB_NONBLOCK (*gnes.service.base.ParallelType* attribute), 73
 PublishRouter (class in *gnes.router.map*), 67
 PULL_BIND (*gnes.service.base.SocketType* attribute), 74
 PULL_CONNECT (*gnes.service.base.SocketType* attribute), 74
 PUSH_BIND (*gnes.service.base.SocketType* attribute), 74
 PUSH_BLOCK (*gnes.service.base.ParallelType* attribute), 73
 PUSH_CONNECT (*gnes.service.base.SocketType* attribute), 74
 PUSH_NONBLOCK (*gnes.service.base.ParallelType* attribute), 73
 pyramid_descriptor () (in module *gnes.preprocessor.helper*), 63

Q

QuantizerEncoder (class in *gnes.encoder.numeric.quantizer*), 30
 query () (*gnes.client.cli.CLIClient* method), 22
 query () (*gnes.flow.Flow* method), 42
 query () (*gnes.indexer.base.BaseChunkIndexer* method), 51
 query () (*gnes.indexer.base.BaseChunkIndexerHelper* method), 51
 query () (*gnes.indexer.base.BaseDocIndexer* method), 52
 query () (*gnes.indexer.base.BaseIndexer* method), 52

- query () (*gnes.indexer.base.JointIndexer* method), 52
 query () (*gnes.indexer.chunk.annoy.AnnoyIndexer* method), 45
 query () (*gnes.indexer.chunk.faiss.FaissIndexer* method), 46
 query () (*gnes.indexer.chunk.helper.DictKeyIndexer* method), 46
 query () (*gnes.indexer.chunk.helper.ListKeyIndexer* method), 47
 query () (*gnes.indexer.chunk.helper.ListNumpyKeyIndexer* method), 47
 query () (*gnes.indexer.chunk.helper.NumpyKeyIndexer* method), 47
 query () (*gnes.indexer.chunk.numpy.NumpyIndexer* method), 48
 query () (*gnes.indexer.doc.dict.DictIndexer* method), 48
 query () (*gnes.indexer.doc.filesys.DirectoryIndexer* method), 49
 query () (*gnes.indexer.doc.leveldb.AsyncLVDBIndexer* method), 49
 query () (*gnes.indexer.doc.leveldb.LVDBIndexer* method), 50
 query () (*gnes.indexer.doc.rocksdb.RocksDBIndexer* method), 50
 Query () (*gnes.proto.gnes_pb2_grpc.GnesRPCService* method), 64
 query () (*gnes.proto.RequestGenerator* static method), 64
 query_and_score () (*gnes.indexer.base.BaseChunkIndexer* method), 51
 query_and_score () (*gnes.indexer.base.BaseDocIndexer* method), 52
 query_and_score () (*gnes.indexer.base.BaseIndexer* method), 52
- ## R
- ran_gen () (*gnes.encoder.numeric.hash.HashEncoder* method), 28
 random_port () (in module *gnes.cli.parser*), 21
 raw_to_chunk () (*gnes.preprocessor.base.UnaryPreprocessor* method), 62
 RawChunkPreprocessor (class in *gnes.preprocessor.base*), 62
 reciprocal (*gnes.score_fn.base.ScoreOps* attribute), 70
 reciprocal1p (*gnes.score_fn.base.ScoreOps* attribute), 70
 recv_message () (*gnes.client.base.ZmqClient* method), 22
 recv_message () (in module *gnes.proto*), 64
 reduce_embedding () (*gnes.router.base.BaseEmbedReduceRouter* method), 65
 reduce_embedding () (*gnes.router.reduce.AvgEmbedRouter* method), 67
 reduce_embedding () (*gnes.router.reduce.ConcatEmbedRouter* method), 68
 ReduceOp (class in *gnes.service.base*), 73
 register () (*gnes.client.base.ResponseHandler* method), 22
 register () (*gnes.service.base.MessageHandler* method), 73
 register_hook () (*gnes.service.base.MessageHandler* method), 73
 release () (*gnes.helper.FileLock* method), 76
 remove () (*gnes.flow.Flow* method), 42
 RequestGenerator (class in *gnes.proto*), 64
 ResizeChunkPreprocessor (class in *gnes.preprocessor.image.resize*), 56
 resolve_py_path () (in module *gnes.cli.parser*), 21
 resolve_yaml_path () (in module *gnes.cli.parser*), 21
 ResponseHandler (class in *gnes.client.base*), 22
 rgb_histogram () (in module *gnes.preprocessor.helper*), 63
 RocksDBIndexer (class in *gnes.indexer.doc.rocksdb*), 50
 route () (in module *gnes.cli.api*), 21
 Router (*gnes.flow.helper.Service* attribute), 39
 RouterService (class in *gnes.service.router*), 75
 run () (*gnes.composer.flask.YamlComposerFlask* method), 24
 run () (*gnes.composer.http.YamlComposerHttp* method), 24
 run () (*gnes.service.base.BaseService* method), 72
 run_command () (in module *gnes.preprocessor.io_utils.helper*), 57
 run_command_async () (in module *gnes.preprocessor.io_utils.helper*), 57
 RUNTIME (*gnes.flow.helper.BuildLevel* attribute), 39
- ## S
- scale_video () (in module *gnes.preprocessor.io_utils.video*), 58
 scan () (*gnes.indexer.doc.rocksdb.RocksDBIndexer* method), 50
 ScoreOps (class in *gnes.score_fn.base*), 69
 SegmentPreprocessor (class in *gnes.preprocessor.image.segmentation*), 56
 send_ctrl_message () (in module *gnes.service.base*), 74

send_message() (*gnes.client.base.ZmqClient method*), 22
 send_message() (*in module gnes.proto*), 64
 send_request() (*gnes.client.stream.StreamingClient method*), 23
 send_request() (*gnes.client.stream.SyncClient method*), 23
 SentSplitPreprocessor (*class in gnes.preprocessor.text.split*), 58
 Service (*class in gnes.flow.helper*), 39
 ServiceError, 73
 ServiceManager (*class in gnes.service.base*), 73
 set() (*gnes.flow.Flow method*), 42
 set_base_parser() (*in module gnes.cli.parser*), 21
 set_client_cli_parser() (*in module gnes.cli.parser*), 21
 set_client_http_parser() (*in module gnes.cli.parser*), 21
 set_composer_flask_parser() (*in module gnes.cli.parser*), 21
 set_composer_parser() (*in module gnes.cli.parser*), 21
 set_encoder_parser() (*in module gnes.cli.parser*), 21
 set_frontend_parser() (*in module gnes.cli.parser*), 21
 set_grpc_service_parser() (*in module gnes.cli.parser*), 21
 set_healthcheck_parser() (*in module gnes.cli.parser*), 21
 set_indexer_parser() (*in module gnes.cli.parser*), 21
 set_key() (*gnes.router.base.BaseTopkReduceRouter method*), 66
 set_key() (*gnes.router.reduce.Chunk2DocTopkReducer method*), 67
 set_key() (*gnes.router.reduce.ChunkTopkReducer method*), 68
 set_key() (*gnes.router.reduce.DocTopkReducer method*), 68
 set_last_service() (*gnes.flow.Flow method*), 43
 set_logger() (*in module gnes.helper*), 75
 set_preprocessor_parser() (*in module gnes.cli.parser*), 21
 set_router_parser() (*in module gnes.cli.parser*), 21
 set_service_parser() (*in module gnes.cli.parser*), 21
 ShotDetectorPreprocessor (*class in gnes.preprocessor.video.shot_detector*), 60
 Singleton (*class in gnes.helper*), 76
 SizedPreprocessor (*class in gnes.preprocessor.image.resize*), 56
 SocketType (*class in gnes.service.base*), 73
 spectrogram_to_mel_matrix() (*in module gnes.preprocessor.audio.vggish_example_helper.mel_features*), 54
 split_mp4_random() (*in module gnes.preprocessor.helper*), 63
 split_video_frames() (*in module gnes.preprocessor.helper*), 63
 sqrt (*gnes.score_fn.base.ScoreOps attribute*), 70
 square (*gnes.score_fn.base.ScoreOps attribute*), 70
 StandarderEncoder (*class in gnes.encoder.numeric.standarder*), 30
 start() (*gnes.client.base.GrpcClient method*), 22
 start() (*gnes.client.cli.CLIClient method*), 22
 start() (*gnes.client.http.HttpClient method*), 22
 status (*gnes.service.base.BaseService attribute*), 72
 stft_magnitude() (*in module gnes.preprocessor.audio.vggish_example_helper.mel_features*), 54
 stop() (*gnes.service.frontend.FrontendService method*), 74
 store_args_kwargs (*gnes.base.TrainableBase attribute*), 20
 store_args_kwargs (*gnes.encoder.text.bert.BertEncoder attribute*), 31
 store_args_kwargs (*gnes.encoder.text.bert.BertEncoderServer attribute*), 31
 store_args_kwargs (*gnes.preprocessor.video.shot_detector.ShotDetectorPreprocessor attribute*), 60
 store_args_kwargs (*gnes.preprocessor.video.video_decoder.VideoDecoderPreprocessor attribute*), 61
 stream_call() (*gnes.client.base.GrpcClient method*), 22
 StreamCall() (*gnes.proto.gnes_pb2_grpc.GnesRPCService method*), 64
 StreamingClient (*class in gnes.client.stream*), 23
 SUB_BIND (*gnes.service.base.SocketType attribute*), 74
 SUB_CONNECT (*gnes.service.base.SocketType attribute*), 74
 sum (*gnes.score_fn.base.ScoreOps attribute*), 70
 supported_ops (*gnes.score_fn.base.CombinedScoreFn attribute*), 69
 supported_ops (*gnes.score_fn.base.ModifierScoreFn attribute*), 69
 SyncClient (*class in gnes.client.stream*), 23

T

TFIDFChunkScoreFn (*class in gnes.score_fn.chunk*), 70
 TFInceptionEncoder (*class in gnes.encoder.image.inception*), 27

TFPQEncoder (class in *gnes.encoder.numeric.tf_pq*), 30

thre_algo() (in module *gnes.preprocessor.helper*), 63

time_profile() (in module *gnes.helper*), 75

to_jpg() (*gnes.flow.Flow* method), 43

to_k8s_yaml() (*gnes.flow.Flow* method), 43

to_mermaid() (*gnes.flow.Flow* method), 43

to_python_code() (*gnes.flow.Flow* method), 43

to_shell_script() (*gnes.flow.Flow* method), 44

to_swarm_yaml() (*gnes.flow.Flow* method), 44

to_url() (*gnes.flow.Flow* method), 44

to_yaml() (*gnes.base.CompositionalTrainableBase* class method), 20

to_yaml() (*gnes.base.TrainableBase* class method), 20

torch_transform() (in module *gnes.preprocessor.helper*), 63

TorchvisionEncoder (class in *gnes.encoder.image.torchvision*), 28

train() (*gnes.base.CompositionalTrainableBase* method), 20

train() (*gnes.base.TrainableBase* method), 20

train() (*gnes.client.cli.CLIClient* method), 22

train() (*gnes.encoder.audio.mfcc.MfccEncoder* method), 25

train() (*gnes.encoder.audio.vggish.VggishEncoder* method), 26

train() (*gnes.encoder.base.BaseAudioEncoder* method), 36

train() (*gnes.encoder.base.BaseBinaryEncoder* method), 36

train() (*gnes.encoder.base.BaseEncoder* method), 36

train() (*gnes.encoder.base.BaseImageEncoder* method), 36

train() (*gnes.encoder.base.BaseNumericEncoder* method), 36

train() (*gnes.encoder.base.BaseTextEncoder* method), 37

train() (*gnes.encoder.base.BaseVideoEncoder* method), 37

train() (*gnes.encoder.base.PipelineEncoder* method), 37

train() (*gnes.encoder.image.cvae.CVAEEncoder* method), 27

train() (*gnes.encoder.image.inception.TFInceptionEncoder* method), 27

train() (*gnes.encoder.image.onnx.BaseONNXImageEncoder* method), 27

train() (*gnes.encoder.image.torchvision.TorchvisionEncoder* method), 28

train() (*gnes.encoder.numeric.hash.HashEncoder* method), 28

train() (*gnes.encoder.numeric.pca.PCAEncoder* method), 29

train() (*gnes.encoder.numeric.pca.PCALocalEncoder* method), 29

train() (*gnes.encoder.numeric.pooling.PoolingEncoder* method), 29

train() (*gnes.encoder.numeric.pq.PQEncoder* method), 29

train() (*gnes.encoder.numeric.quantizer.QuantizerEncoder* method), 30

train() (*gnes.encoder.numeric.standarder.StandarderEncoder* method), 30

train() (*gnes.encoder.numeric.tf_pq.TFPQEncoder* method), 30

train() (*gnes.encoder.numeric.vlad.VladEncoder* method), 31

train() (*gnes.encoder.text.bert.BertEncoder* method), 31

train() (*gnes.encoder.text.bert.BertEncoderServer* method), 31

train() (*gnes.encoder.text.bert.BertEncoderWithServer* method), 32

train() (*gnes.encoder.text.char.CharEmbeddingEncoder* method), 32

train() (*gnes.encoder.text.flair.FlairEncoder* method), 32

train() (*gnes.encoder.text.w2v.Word2VecEncoder* method), 33

train() (*gnes.encoder.video.incep_mixture.IncepMixtureEncoder* method), 34

train() (*gnes.encoder.video.inception.InceptionVideoEncoder* method), 34

train() (*gnes.encoder.video.yt8m_feature_extractor.YouTube8MFeatureExtractor* method), 35

train() (*gnes.encoder.video.yt8m_model.YouTube8MEncoder* method), 36

train() (*gnes.flow.base.BaseIndexFlow* method), 37

train() (*gnes.flow.base.BaseQueryFlow* method), 38

train() (*gnes.flow.Flow* method), 44

train() (*gnes.indexer.base.BaseChunkIndexer* method), 51

train() (*gnes.indexer.base.BaseChunkIndexerHelper* method), 51

train() (*gnes.indexer.base.BaseDocIndexer* method), 52

train() (*gnes.indexer.base.BaseIndexer* method), 52

train() (*gnes.indexer.base.JointIndexer* method), 52

train() (*gnes.indexer.chunk.annoy.AnnoyIndexer* method), 46

train() (*gnes.indexer.chunk.faiss.FaissIndexer* method), 46

train() (*gnes.indexer.chunk.helper.DictKeyIndexer* method), 46

train() (*gnes.indexer.chunk.helper.ListKeyIndexer* method), 47

[train\(\) \(gnes.indexer.chunk.helper.ListNumpyKeyIndexer method\), 47](#)
[train\(\) \(gnes.indexer.chunk.helper.NumpyKeyIndexer method\), 48](#)
[train\(\) \(gnes.indexer.chunk.numpy.NumpyIndexer method\), 48](#)
[train\(\) \(gnes.indexer.doc.dict.DictIndexer method\), 48](#)
[train\(\) \(gnes.indexer.doc.filesys.DirectoryIndexer method\), 49](#)
[train\(\) \(gnes.indexer.doc.leveldb.AsyncLVDBIndexer method\), 49](#)
[train\(\) \(gnes.indexer.doc.leveldb.LVDBIndexer method\), 50](#)
[train\(\) \(gnes.indexer.doc.rocksdb.RocksDBIndexer method\), 50](#)
[train\(\) \(gnes.preprocessor.audio.vggish_example.VggishPreprocessor method\), 55](#)
[train\(\) \(gnes.preprocessor.base.BaseAudioPreprocessor method\), 61](#)
[train\(\) \(gnes.preprocessor.base.BaseImagePreprocessor method\), 61](#)
[train\(\) \(gnes.preprocessor.base.BasePreprocessor method\), 61](#)
[train\(\) \(gnes.preprocessor.base.BaseTextPreprocessor method\), 62](#)
[train\(\) \(gnes.preprocessor.base.BaseVideoPreprocessor method\), 62](#)
[train\(\) \(gnes.preprocessor.base.PipelinePreprocessor method\), 62](#)
[train\(\) \(gnes.preprocessor.base.RawChunkPreprocessor method\), 62](#)
[train\(\) \(gnes.preprocessor.base.UnaryPreprocessor method\), 62](#)
[train\(\) \(gnes.preprocessor.image.resize.ResizeChunkPreprocessor method\), 56](#)
[train\(\) \(gnes.preprocessor.image.resize.SizedPreprocessor method\), 56](#)
[train\(\) \(gnes.preprocessor.image.segmentation.SegmentPreprocessor method\), 56](#)
[train\(\) \(gnes.preprocessor.image.sliding_window.VanillaSlidingPreprocessor method\), 56](#)
[train\(\) \(gnes.preprocessor.image.sliding_window.WeightedSlidingPreprocessor method\), 57](#)
[train\(\) \(gnes.preprocessor.text.split.SentSplitPreprocessor method\), 58](#)
[train\(\) \(gnes.preprocessor.video.ffmpeg.FFmpegPreprocessor method\), 59](#)
[train\(\) \(gnes.preprocessor.video.ffmpeg.FFmpegVideoSegmentor method\), 59](#)
[train\(\) \(gnes.preprocessor.video.ffmpeg.GifChunkPreprocessor method\), 59](#)
[train\(\) \(gnes.preprocessor.video.frame_select.FrameSelectPreprocessor method\), 60](#)
[train\(\) \(gnes.preprocessor.video.shot_detector.ShotDetectorPreprocessor method\), 60](#)
[train\(\) \(gnes.preprocessor.video.video_decoder.VideoDecoderPreprocessor method\), 61](#)
[train\(\) \(gnes.preprocessor.video.video_encoder.VideoEncoderPreprocessor method\), 61](#)
[Train\(\) \(gnes.proto.gnes_pb2_grpc.GnesRPCService method\), 64](#)
[train\(\) \(gnes.proto.RequestGenerator static method\), 64](#)
[train\(\) \(gnes.router.base.BaseEmbedReduceRouter method\), 65](#)
[train\(\) \(gnes.router.base.BaseMapRouter method\), 65](#)
[train\(\) \(gnes.router.base.BaseReduceRouter method\), 65](#)
[train\(\) \(gnes.router.base.BaseRouter method\), 66](#)
[train\(\) \(gnes.router.base.BaseTopkReduceRouter method\), 66](#)
[train\(\) \(gnes.router.base.PipelineRouter method\), 66](#)
[train\(\) \(gnes.router.map.BlockRouter method\), 66](#)
[train\(\) \(gnes.router.map.DocBatchRouter method\), 67](#)
[train\(\) \(gnes.router.map.PublishRouter method\), 67](#)
[train\(\) \(gnes.router.reduce.AvgEmbedRouter method\), 67](#)
[train\(\) \(gnes.router.reduce.Chunk2DocTopkReducer method\), 67](#)
[train\(\) \(gnes.router.reduce.ChunkTopkReducer method\), 68](#)
[train\(\) \(gnes.router.reduce.ConcatEmbedRouter method\), 68](#)
[train\(\) \(gnes.router.reduce.DocFillReducer method\), 68](#)
[train\(\) \(gnes.router.reduce.DocTopkReducer method\), 68](#)
[train\(\) \(gnes.router.reduce.DocTopkReducer method\), 69](#)
[train\(\) \(gnes.score_fn.base.BaseScoreFn method\), 69](#)
[train\(\) \(gnes.score_fn.base.CombinedScoreFn method\), 69](#)
[train\(\) \(gnes.score_fn.base.ModifierScoreFn method\), 69](#)
[train\(\) \(gnes.score_fn.chunk.BM25ChunkScoreFn method\), 70](#)
[train\(\) \(gnes.score_fn.chunk.CoordChunkScoreFn method\), 70](#)
[train\(\) \(gnes.score_fn.chunk.TFIDFChunkScoreFn method\), 70](#)
[train\(\) \(gnes.score_fn.chunk.WeightedChunkOffsetScoreFn method\), 70](#)
[train\(\) \(gnes.score_fn.chunk.WeightedChunkScoreFn method\), 71](#)
[train\(\) \(gnes.score_fn.doc.CoordDocScoreFn method\), 71](#)
[train\(\) \(gnes.score_fn.doc.WeightedDocScoreFn method\), 71](#)

train() (*gnes.score_fn.normalize.Normalizer1* method), 71
 train() (*gnes.score_fn.normalize.Normalizer2* method), 71
 train() (*gnes.score_fn.normalize.Normalizer3* method), 71
 train() (*gnes.score_fn.normalize.Normalizer4* method), 72
 train() (*gnes.score_fn.normalize.Normalizer5* method), 72
 train_kmeans() (*gnes.encoder.numeric.hash.HashEncoder* method), 28
 train_required() (in module *gnes.helper*), 76
 TrainableBase (class in *gnes.base*), 19

U

UnaryPreprocessor (class in *gnes.preprocessor.base*), 62
 update() (*gnes.client.cli.ProgressBar* method), 22
 update_counter() (*gnes.indexer.base.BaseChunkIndexer* static method), 51
 update_counter() (*gnes.indexer.base.BaseDocIndexer* static method), 52
 update_helper_indexer() (*gnes.indexer.base.BaseChunkIndexer* static method), 51

V

VanillaSlidingPreprocessor (class in *gnes.preprocessor.image.sliding_window*), 56
 VggishEncoder (class in *gnes.encoder.audio.vggish*), 26
 VggishPreprocessor (class in *gnes.preprocessor.audio.vggish_example*), 55
 VideoDecoderPreprocessor (class in *gnes.preprocessor.video.video_decoder*), 60
 VideoEncoderPreprocessor (class in *gnes.preprocessor.video.video_encoder*), 61
 VladEncoder (class in *gnes.encoder.numeric.vlad*), 31

W

wait() (in module *gnes.preprocessor.io_utils.helper*), 58
 warn_unnamed (*gnes.score_fn.base.BaseScoreFn* attribute), 69
 waveform_to_examples() (*gnes.preprocessor.audio.vggish_example.VggishPreprocessor* method), 55
 WeightedChunkOffsetScoreFn (class in *gnes.score_fn.chunk*), 70

WeightedChunkScoreFn (class in *gnes.score_fn.chunk*), 71
 WeightedDocScoreFn (class in *gnes.score_fn.doc*), 71
 WeightedSlidingPreprocessor (class in *gnes.preprocessor.image.sliding_window*), 56
 Word2VecEncoder (class in *gnes.encoder.text.w2v*), 32

Y

yaml_full_path (*gnes.base.TrainableBase* attribute), 20
 YamlComposer (class in *gnes.composer.base*), 23
 YamlComposer.Layer (class in *gnes.composer.base*), 23
 YamlComposerFlask (class in *gnes.composer.flask*), 24
 YamlComposerHttp (class in *gnes.composer.http*), 24
 YouTube8MEncoder (class in *gnes.encoder.video.yt8m_model*), 35
 YouTube8MFeatureExtractor (class in *gnes.encoder.video.yt8m_feature_extractor*), 35

Z

ZmqClient (class in *gnes.client.base*), 22