DeDuplicationDict

Release 1.0.4

Vivswan Shah

Jul 13, 2023

CONTENTS

A dictionary that de-duplicates values.

A dictionary-like class that deduplicates values by storing them in a separate dictionary and replacing them with their corresponding hash values. This class is particularly useful for large dictionaries with repetitive entries, as it can save memory by storing values only once and substituting recurring values with their hash representations.

This class supports nested structures by automatically converting nested dictionaries into `DeDuplicationDict` instances. It also provides various conversion methods to convert between regular dictionaries and `DeDuplicationDict` instances.

CHAPTER

ONE

INSTALLATION

```
pip install deduplicationdict
```

CHAPTER

TWO

USAGE

```python
from deduplicationdict import DeDuplicationDict

# Create a new DeDuplicationDict instance
dedup_dict = DeDuplicationDict.from_dict({'a': [5, 6, 7], 'b': 2, 'c': [5, 6, 7]})
# or
dedup_dict = DeDuplicationDict(**{'a': [5, 6, 7], 'b': 2, 'c': [5, 6, 7]})

# Add a new duplicate key-value pair
dedup_dict['d'] = [1, 2, 3]
dedup_dict['e'] = [1, 2, 3]

# Print the dictionary
print(f"dedup_dict.to_dict(): {dedup_dict.to_dict()}")
# output: {'a': [5, 6, 7], 'b': 2, 'c': [5, 6, 7], 'd': [1, 2, 3], 'e': [1, 2, 3]}

# Print the deduplicated dictionary internal
print(f"dedup_dict.key_dict: {dedup_dict.key_dict}")
# output: {'a': '7511debb', 'b': '7c7ad8f0', 'c': '7511debb', 'd': 'f9343d7d', 'e': 'f9343d7d'}
print(f"dedup_dict.value_dict: {dedup_dict.value_dict}")
# output: {'7511debb': [5, 6, 7], '7c7ad8f0': 2, 'f9343d7d': [1, 2, 3]}

# Print the deduplicated dictionary
print(f"to_json_save_dict: {dedup_dict.to_json_save_dict()}")
# output: {'key_dict': {'a': '7511debb', 'b': '7c7ad8f0', 'c': '7511debb', 'd': 'f9343d7d', 'e':
↪'f9343d7d'}, 'value_dict': {'7511debb': [5, 6, 7], '7c7ad8f0': 2, 'f9343d7d': [1, 2, 3]}}

assert dedup_dict["a"] == [5, 6, 7]
assert dedup_dict["b"] == 2
assert dedup_dict["c"] == [5, 6, 7]
assert dedup_dict["d"] == [1, 2, 3]
assert dedup_dict["e"] == [1, 2, 3]
assert DeDuplicationDict.from_json_save_dict(dedup_dict.to_json_save_dict()).to_dict()␣
↪== dedup_dict.to_dict()
```

Usage with SqliteDict: SqliteDeDuplicationDict.py

CHAPTER

THREE

RESULTS FROM TESTING

Method	JSON Memory (MB)	In-Memory (MB)
`dict`	14.089 MB	27.542 MB
`DeDuplicationDict`	1.7906 MB	3.806 MB
Memory Saving	**7.868x**	**7.235x**

CHAPTER

FOUR

DOCUMENTATION

The documentation for this project is hosted on Read the Docs.

CHAPTER

FIVE

LICENSE

This project is licensed under the terms of the Mozilla Public License 2.0.

CHAPTER

SIX

TABLE OF CONTENTS

6.1 API Reference

This page contains auto-generated API reference documentation[1].

6.1.1 deduplicationdict

Package Contents

Classes

DeDuplicationDict	A dictionary that de-duplicates values.

Attributes

__package__
__author__
__version__

deduplicationdict.**__package__** = **'deduplicationdict'**

deduplicationdict.**__author__** = **'Vivswan Shah (vivswanshah@pitt.edu)'**

deduplicationdict.**__version__**

class deduplicationdict.**DeDuplicationDict**(**args*, *_value_dict: dict = None*, ***kwargs*)

Bases: collections.abc.MutableMapping

A dictionary that de-duplicates values.

A dictionary-like class that deduplicates values by storing them in a separate dictionary and replacing them with their corresponding hash values. This class is particularly useful for large dictionaries with repetitive entries, as it can save memory by storing values only once and substituting recurring values with their hash representations.

[1] Created with sphinx-autoapi

This class supports nested structures by automatically converting nested dictionaries into DeDuplicationDict instances. It also provides various conversion methods to convert between regular dictionaries and DeDuplicationDict instances.

Variables

- **hash_length** (*int*) – The length of the hash value used for deduplication.
- **auto_clean_up** (*bool*) – Whether to automatically clean up unused hash values when deleting items.
- **skip_update_on_setitem** (*bool*) – Whether to skip updating the value dictionary when setting an item.
- **key_dict** (*dict*) – A dictionary that maps hash values to their corresponding values.
- **value_dict** (*dict*) – A dictionary that maps values to their corresponding hash values.

_set_value_dict(*value_dict: dict*, *skip_update: bool = False*) → *DeDuplicationDict*

Update the value dictionary and propagate the changes to nested DeDuplicationDict instances.

Parameters

- **value_dict** (*dict*) – The new value dictionary to use for deduplication.
- **skip_update** (*bool*) – Whether to skip updating the value dictionary of nested

Returns
self

Return type
DeDuplicationDict

__setitem__(*key: KT*, *value: VT*) → None

Set the value for the given key, deduplicating the value if necessary.

Parameters

- **key** (*KT*) – The key to set the value for.
- **value** (*VT*) – The value to set for the given key.

__getitem__(*key: KT*) → VT_co

Get the value for the given key.

Parameters
key (*KT*) – The key to get the value for.

Returns
The value for the given key.

Return type
VT_co

Raises

- **KeyError** – If the key is not found in the dictionary.
- **TypeError** – If the value type is not supported.

all_hashes_in_use() → set

Get all hash values currently in use.

Returns
A set of all hash values in use.

Return type
set

`clean_up()` → *DeDuplicationDict*

Remove unused hash values from the value dictionary.

Returns
self

Return type
DeDuplicationDict

`detach()` → *DeDuplicationDict*

Detach the DeDuplicationDict instance from its value dictionary, creating a standalone instance.

Returns
A new DeDuplicationDict instance with its own value dictionary.

Return type
DeDuplicationDict

`__deepcopy__`(*memo: dict*) → *DeDuplicationDict*

Create a deep copy of the DeDuplicationDict instance.

Parameters
memo (*dict*) – A dictionary of memoized values.

Returns
A new DeDuplicationDict instance with its own value dictionary.

Return type
DeDuplicationDict

`_del_detach()` → *DeDuplicationDict*

Detach the DeDuplicationDict instance from its value dictionary and clean up unused hash values.

Returns
self

Return type
DeDuplicationDict

`__delitem__`(*key: KT*) → None

Delete the item with the given key.

Parameters
key (*KT*) – The key of the item to delete.

Raises
KeyError – If the key is not found in the dictionary.

`__len__()` → int

Get the number of items in the dictionary.

Returns
The number of items in the dictionary.

Return type
int

`__iter__`() → Iterator[T_co]

Get an iterator over the keys in the dictionary.

Returns
An iterator over the keys in the dictionary.

Return type
Iterator[T_co]

`__repr__`() → str

Get a string representation of the DeDuplicationDict instance.

Returns
A string representation of the DeDuplicationDict instance.

Return type
str

`to_dict`() → dict

Convert the DeDuplicationDict instance to a regular dictionary.

Returns
A regular dictionary with the same key-value pairs as the DeDuplicationDict instance.

Return type
dict

`classmethod from_dict`(*d: dict*) → *DeDuplicationDict*

Create a DeDuplicationDict instance from a regular dictionary.

Parameters
d (*dict*) – The dictionary to create the DeDuplicationDict instance from.

Returns
A new DeDuplicationDict instance with the same key-value pairs as the given dictionary.

Return type
DeDuplicationDict

`_get_key_dict`() → dict

Get the key dictionary of the DeDuplicationDict instance in a normal dictionary format.

Returns
The key dictionary of the DeDuplicationDict instance.

Return type
dict

`to_json_save_dict`() → dict

Convert the DeDuplicationDict instance to a dictionary that can be saved to a JSON file.

Returns
A dictionary that can be saved to a JSON file.

Return type
dict

`_set_key_dict`(*key_dict: dict*) → *DeDuplicationDict*

Set the key dictionary of the DeDuplicationDict instance from a normal dictionary format.

Parameters
key_dict (*dict*) – The key dictionary to set.

Returns
self

Return type
DeDuplicationDict

classmethod from_json_save_dict(*d: dict*, *_v: dict = None*) → *DeDuplicationDict*

Create a DeDuplicationDict instance from a dictionary that was saved to a JSON file.

Parameters

- **d** (*dict*) – The dictionary that was saved to a JSON file.
- **_v** (*dict, optional*) – The value dictionary to use. Defaults to None.

Returns
A new DeDuplicationDict instance with the same key-value pairs as the given dictionary.

Return type
DeDuplicationDict

6.2 Changelog

6.2.1 1.0.4

- Added support for multi hash length

6.2.2 1.0.3

- Moved all class variables to object variables

6.2.3 1.0.2

- Added `__deepcopy__` and more tests
- Optimizations of `__setitem__` by value of type DeDuplication has `value_dict` consistent
- Added `skip_update_on_setitem` to `DeDuplication` to skip update `value_dict` on `__setitem__`

6.2.4 1.0.1

- Optimizations and remove redundant code

6.2.5 1.0.0

- Public release

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

d

INDEX

Symbols

A

C

D

F

M

T