
MatchZoo Documentation

Release 2.1

MatchZoo

Jun 02, 2021

CONTENTS:

1	matchzoo	3
1.1	matchzoo package	3
2	MatchZoo Model Reference	33
2.1	Naive	33
2.2	DSSM	33
2.3	CDSSM	34
2.4	DenseBaseline	35
2.5	ArcI	36
2.6	ArcII	40
2.7	MatchPyramid	41
2.8	KNRM	42
2.9	DUET	43
2.10	DRMMTKS	45
2.11	DRMM	46
2.12	ANMM	47
2.13	MVLSTM	48
2.14	MatchLSTM	49
2.15	ConvKNRM	50
3	Indices and tables	53
	Python Module Index	55
	Index	57



MatchZoo is a toolkit for text matching. It was developed with a focus on facilitating the designing, comparing and sharing of deep text matching models. There are a number of deep matching methods, such as DRMM, MatchPyramid, MV-LSTM, aNMM, DUET, ARC-I, ARC-II, DSSM, and CDSSM, designed with a unified interface. Potential tasks related to MatchZoo include document retrieval, question answering, conversational response ranking, paraphrase identification, etc. We are always happy to receive any code contributions, suggestions, comments from all our MatchZoo users.

MATCHZOO

1.1 matchzoo package

1.1.1 Subpackages

matchzoo.auto package

Subpackages

matchzoo.auto.preparer package

Submodules

matchzoo.auto.preparer.prepare module

matchzoo.auto.preparer.preparer module

Module contents

matchzoo.auto.tuner package

Subpackages

matchzoo.auto.tuner.callbacks package

Submodules

matchzoo.auto.tuner.callbacks.callback module

matchzoo.auto.tuner.callbacks.lambda_callback module

matchzoo.auto.tuner.callbacks.load_embedding_matrix module

matchzoo.auto.tuner.callbacks.save_model module

Module contents

Submodules

`matchzoo.auto.tuner.tune` module

`matchzoo.auto.tuner.tuner` module

Module contents

Module contents

`matchzoo.data_generator` package

Subpackages

`matchzoo.data_generator.callbacks` package

Submodules

`matchzoo.data_generator.callbacks.callback` module

`matchzoo.data_generator.callbacks.dynamic_pooling` module

`matchzoo.data_generator.callbacks.histogram` module

`matchzoo.data_generator.callbacks.lambda_callback` module

Module contents

Submodules

`matchzoo.data_generator.data_generator` module

`matchzoo.data_generator.data_generator_builder` module

Module contents

`matchzoo.data_pack` package

Submodules

`matchzoo.data_pack.data_pack` module

Matchzoo DataPack, pair-wise tuple (feature) and context as input.

```
class matchzoo.data_pack.data_pack.DataPack(relation, left, right)
```

Bases: object

Matchzoo *DataPack* data structure, store dataframe and context.

DataPack is a MatchZoo native data structure that most MatchZoo data handling processes build upon. A *DataPack* consists of three parts: *left*, *right* and *relation*, each one of is a *pandas.DataFrame*.

Parameters

- **relation** (DataFrame) – Store the relation between left document and right document use ids.
- **left** (DataFrame) – Store the content or features for id_left.
- **right** (DataFrame) – Store the content or features for id_right.

Example

```
>>> left = [
...     ['qid1', 'query 1'],
...     ['qid2', 'query 2']
... ]
>>> right = [
...     ['did1', 'document 1'],
...     ['did2', 'document 2']
... ]
>>> relation = [['qid1', 'did1', 1], ['qid2', 'did2', 1]]
>>> relation_df = pd.DataFrame(relation)
>>> left = pd.DataFrame(left)
>>> right = pd.DataFrame(right)
>>> dp = DataPack(
...     relation=relation_df,
...     left=left,
...     right=right,
... )
>>> len(dp)
2
```

```
DATA_FILENAME = 'data.dill'
```

```
class FrameView(data_pack)
```

Bases: object

FrameView.

```
append_text_length(verbose=1)
```

Append *length_left* and *length_right* columns.

Parameters

- **inplace** – *True* to modify inplace, *False* to return a modified copy. (default: *False*)
- **verbose** – Verbosity.

Example

```
>>> import matchzoo as mz
>>> data_pack = mz.datasets.toy.load_data()
>>> 'length_left' in data_pack.frame[0].columns
False
>>> new_data_pack = data_pack.append_text_length(verbose=0)
>>> 'length_left' in new_data_pack.frame[0].columns
True
>>> 'length_left' in data_pack.frame[0].columns
False
>>> data_pack.append_text_length(inplace=True, verbose=0)
>>> 'length_left' in data_pack.frame[0].columns
True
```

apply_on_text(*func*, *mode*='both', *rename*=None, *verbose*=1)
Apply *func* to text columns based on *mode*.

Parameters

- **func** (Callable) – The function to apply.
- **mode** (str) – One of “both”, “left” and “right”.
- **rename** (Optional[str]) – If set, use new names for results instead of replacing the original columns. To set *rename* in “both” mode, use a tuple of *str*, e.g. (“text_left_new_name”, “text_right_new_name”).
- **inplace** – *True* to modify inplace, *False* to return a modified copy. (default: *False*)
- **verbose** (int) – Verbosity.

Examples::

```
>>> import matchzoo as mz
>>> data_pack = mz.datasets.toy.load_data()
>>> frame = data_pack.frame
```

To apply *len* on the left text and add the result as ‘length_left’:

```
>>> data_pack.apply_on_text(len, mode='left',
...                          rename='length_left',
...                          inplace=True,
...                          verbose=0)
>>> list(frame[0].columns) # noqa: E501
['id_left', 'text_left', 'length_left', 'id_right', 'text_right', 'label']
```

To do the same to the right text:

```
>>> data_pack.apply_on_text(len, mode='right',
...                          rename='length_right',
...                          inplace=True,
...                          verbose=0)
>>> list(frame[0].columns) # noqa: E501
['id_left', 'text_left', 'length_left', 'id_right', 'text_right', 'length_
↪right', 'label']
```

To do the same to the both texts at the same time:

```
>>> data_pack.apply_on_text(len, mode='both',
...                         rename=('extra_left', 'extra_right'),
...                         inplace=True,
...                         verbose=0)
>>> list(frame[0].columns) # noqa: E501
['id_left', 'text_left', 'length_left', 'extra_left', 'id_right', 'text_
↪right', 'length_right', 'extra_right', 'label']
```

To suppress outputs:

```
>>> data_pack.apply_on_text(len, mode='both', verbose=0,
...                         inplace=True)
```

copy()

Return type *DataPack*

Returns A deep copy.

drop_invalid()

Remove rows from the data pack where the length is zero.

Parameters **inplace** – *True* to modify inplace, *False* to return a modified copy. (default: *False*)

Example

```
>>> import matchzoo as mz
>>> data_pack = mz.datasets.toy.load_data()
>>> data_pack.append_text_length(inplace=True, verbose=0)
>>> data_pack.drop_invalid(inplace=True)
```

drop_label()

Remove *label* column from the data pack.

Parameters **inplace** – *True* to modify inplace, *False* to return a modified copy. (default: *False*)

Example

```
>>> import matchzoo as mz
>>> data_pack = mz.datasets.toy.load_data()
>>> data_pack.has_label
True
>>> data_pack.drop_label(inplace=True)
>>> data_pack.has_label
False
```

property frame: matchzoo.data_pack.data_pack.DataPack.FrameView

View the data pack as a pandas.DataFrame.

Returned data frame is created by merging the left data frame, the right dataframe and the relation data frame. Use *[]* to access an item or a slice of items.

Return type *FrameView*

Returns A `matchzoo.DataPack.FrameView` instance.

Example

```
>>> import matchzoo as mz
>>> data_pack = mz.datasets.toy.load_data()
>>> type(data_pack.frame)
<class 'matchzoo.data_pack.data_pack.DataPack.FrameView'>
>>> frame_slice = data_pack.frame[0:5]
>>> type(frame_slice)
<class 'pandas.core.frame.DataFrame'>
>>> list(frame_slice.columns)
['id_left', 'text_left', 'id_right', 'text_right', 'label']
>>> full_frame = data_pack.frame()
>>> len(full_frame) == len(data_pack)
True
```

property `has_label`: bool

True if *label* column exists, *False* other wise.

Type return

Return type bool

property `left`: `pandas.core.frame.DataFrame`

Get *left()* of *DataPack*.

Return type `DataFrame`

one_hot_encode_label(*num_classes=2*)

One-hot encode *label* column of *relation*.

Parameters

- **num_classes** – Number of classes.
- **inplace** – *True* to modify inplace, *False* to return a modified copy. (default: *False*)

Returns

property `relation`

relation getter.

property `right`: `pandas.core.frame.DataFrame`

Get *right()* of *DataPack*.

Return type `DataFrame`

save(*dirpath*)

Save the *DataPack* object.

A saved *DataPack* is represented as a directory with a *DataPack* object (transformed user input as features and context), it will be saved by *pickle*.

Parameters **dirpath** (Union[str, Path]) – directory path of the saved *DataPack*.

shuffle()

Shuffle the data pack by shuffling the relation column.

Parameters **inplace** – *True* to modify inplace, *False* to return a modified copy. (default: *False*)

Example

```
>>> import matchzoo as mz
>>> import numpy.random
>>> numpy.random.seed(0)
>>> data_pack = mz.datasets.toy.load_data()
>>> orig_ids = data_pack.relation['id_left']
>>> shuffled = data_pack.shuffle()
>>> (shuffled.relation['id_left'] != orig_ids).any()
True
```

unpack()

Unpack the data for training.

The return value can be directly feed to *model.fit* or *model.fit_generator*.

Return type Tuple[Dict[str, array], Optional[array]]

Returns A tuple of (X, y). y is *None* if *self* has no label.

Example

```
>>> import matchzoo as mz
>>> data_pack = mz.datasets.toy.load_data()
>>> X, y = data_pack.unpack()
>>> type(X)
<class 'dict'>
>>> sorted(X.keys())
['id_left', 'id_right', 'text_left', 'text_right']
>>> type(y)
<class 'numpy.ndarray'>
>>> X, y = data_pack.drop_label().unpack()
>>> type(y)
<class 'NoneType'>
```

`matchzoo.data_pack.data_pack.load_data_pack(dirpath)`

Load a *DataPack*. The reverse function of *save()*.

Parameters *dirpath* (Union[str, Path]) – directory path of the saved model.

Return type *DataPack*

Returns a *DataPack* instance.

matchzoo.data_pack.pack module

Convert list of input into class:*DataPack* expected format.

`matchzoo.data_pack.pack.pack(df)`

Pack a *DataPack* using *df*.

The *df* must have *text_left* and *text_right* columns. Optionally, the *df* can have *id_left*, *id_right* to index *text_left* and *text_right* respectively. *id_left*, *id_right* will be automatically generated if not specified.

Parameters *df* (DataFrame) – Input pandas.DataFrame to use.

Examples::

```
>>> import matchzoo as mz
>>> import pandas as pd
>>> df = pd.DataFrame(data={'text_left': list('AABC'),
...                        'text_right': list('abbc'),
...                        'label': [0, 1, 1, 0]})
>>> mz.pack(df).frame()
  id_left text_left id_right text_right  label
0     L-0         A      R-0         a      0
1     L-0         A      R-1         b      1
2     L-1         B      R-1         b      1
3     L-2         C      R-2         c      0
```

Return type *DataPack*

Module contents

matchzoo.datasets package

Subpackages

matchzoo.datasets.embeddings package

Submodules

matchzoo.datasets.embeddings.load_glove_embedding module

Module contents

matchzoo.datasets.quora_qp package

Submodules

matchzoo.datasets.quora_qp.load_data module

Module contents

matchzoo.datasets.snli package

Submodules

matchzoo.datasets.snli.load_data module

Module contents

matchzoo.datasets.toy package

Module contents

matchzoo.datasets.wiki_qa package

Submodules

matchzoo.datasets.wiki_qa.load_data module

Module contents

Module contents

matchzoo.embedding package

Submodules

matchzoo.embedding.embedding module

Module contents

matchzoo.engine package

Submodules

matchzoo.engine.base_metric module

Metric base class and some related utilities.

```
class matchzoo.engine.base_metric.BaseMetric
```

Bases: abc.ABC

Metric base class.

```
ALIAS = 'base_metric'
```

```
matchzoo.engine.base_metric.sort_and_couple(labels, scores)
```

Zip the *labels* with *scores* into a single list.

Return type array

matchzoo.engine.base_model module

matchzoo.engine.base_preprocessor module

matchzoo.engine.base_task module

Base task.

```
class matchzoo.engine.base_task.BaseTask(loss=None, metrics=None)
```

Bases: abc.ABC

Base Task, shouldn't be used directly.

abstract classmethod `list_available_losses()`

Return type `list`

Returns a list of available losses.

abstract classmethod `list_available_metrics()`

Return type `list`

Returns a list of available metrics.

property `loss`

Loss used in the task.

Type `return`

property `metrics`

Metrics used in the task.

Type `return`

abstract property `output_dtype`

output data type for specific task.

Type `return`

abstract property `output_shape: tuple`

output shape of a single sample of the task.

Type `return`

Return type `tuple`

matchzoo.engine.callbacks module

matchzoo.engine.hyper_spaces module

Hyper parameter search spaces wrapping *hyperopt*.

class `matchzoo.engine.hyper_spaces.HyperoptProxy(hyperopt_func, **kwargs)`

Bases: `object`

Hyperopt proxy class.

See *hyperopt*'s documentation for more details: <https://github.com/hyperopt/hyperopt/wiki/FMin>

Reason of these wrappers:

A hyper space in *hyperopt* requires a *label* to instantiate. This *label* is used later as a reference to original hyper space that is sampled. In *matchzoo*, hyper spaces are used in `matchzoo.engine.Param`. Only if a hyper space's label matches its parent `matchzoo.engine.Param`'s name, *matchzoo* can correctly back-referenced the parameter got sampled. This can be done by asking the user always use the same name for a parameter and its hyper space, but typos can occur. As a result, these wrappers are created to hide hyper spaces' *label*, and always correctly bind them with its parameter's name.

Examples::

```
>>> import matchzoo as mz
>>> from hyperopt.pyll.stochastic import sample
```

Basic Usage:

```
>>> model = mz.models.DenseBaseline()
>>> sample(model.params.hyper_space)
{'mlp_num_layers': 1.0, 'mlp_num_units': 274.0}
```

Arithmetic Operations:

```
>>> new_space = 2 ** mz.hyper_spaces.quniform(2, 6)
>>> model.params.get('mlp_num_layers').hyper_space = new_space
>>> sample(model.params.hyper_space)
{'mlp_num_layers': 8.0, 'mlp_num_units': 292.0}
```

convert(name)

Attach *name* as *hyperopt.hp*'s label.

Parameters *name* (str) –

Return type Apply

Returns a *hyperopt* ready search space

class matchzoo.engine.hyper_spaces.**choice**(*options*)

Bases: [matchzoo.engine.hyper_spaces.HyperoptProxy](#)

hyperopt.hp.choice() proxy.

class matchzoo.engine.hyper_spaces.**quniform**(*low, high, q=1*)

Bases: [matchzoo.engine.hyper_spaces.HyperoptProxy](#)

hyperopt.hp.quniform() proxy.

matchzoo.engine.hyper_spaces.**sample**(*space*)

Take a sample in the hyper space.

This method is stateless, so the distribution of the samples is different from that of *tune* call. This function just gives a general idea of what a sample from the *space* looks like.

Example

```
>>> import matchzoo as mz
>>> space = mz.models.Naive.get_default_params().hyper_space
>>> mz.hyper_spaces.sample(space)
{'optimizer': ...}
```

class matchzoo.engine.hyper_spaces.**uniform**(*low, high*)

Bases: [matchzoo.engine.hyper_spaces.HyperoptProxy](#)

hyperopt.hp.uniform() proxy.

matchzoo.engine.param module

Parameter class.

```
class matchzoo.engine.param.Param(name, value=None, hyper_space=None, validator=None, desc=None)
```

Bases: object

Parameter class.

Basic usages with a name and value:

```
>>> param = Param('my_param', 10)
>>> param.name
'my_param'
>>> param.value
10
```

Use with a validator to make sure the parameter always keeps a valid value.

```
>>> param = Param(
...     name='my_param',
...     value=5,
...     validator=lambda x: 0 < x < 20
... )
>>> param.validator
<function <lambda> at 0x...>
>>> param.value
5
>>> param.value = 10
>>> param.value
10
>>> param.value = -1
Traceback (most recent call last):
...
ValueError: Validator not satisfied.
The validator's definition is as follows:
validator=lambda x: 0 < x < 20
```

Use with a hyper space. Setting up a hyper space for a parameter makes the parameter tunable in a `matchzoo.engine.Tuner`.

```
>>> from matchzoo.engine.hyper_spaces import uniform
>>> param = Param(
...     name='positive_num',
...     value=1,
...     hyper_space=uniform(low=1, high=5)
... )
>>> param.hyper_space
<matchzoo.engine.hyper_spaces.uniform object at ...>
>>> from hyperopt.pyll.stochastic import sample
>>> hyperopt_space = param.hyper_space.convert(param.name)
>>> samples = [sample(hyperopt_space) for _ in range(64)]
>>> set(samples) == {1, 2, 3, 4, 5}
True
```

The boolean value of a `Param` instance is only `True` when the value is not `None`. This is because some default

falsy values like zero or an empty list are valid parameter values. In other words, the boolean value means to be “if the parameter value is filled”.

```
>>> param = Param('dropout')
>>> if param:
...     print('OK')
>>> param = Param('dropout', 0)
>>> if param:
...     print('OK')
OK
```

A `_pre_assignment_hook` is initialized as a data type convertor if the value is set as a number to keep data type consistency of the parameter. This conversion supports python built-in numbers, *numpy* numbers, and any number that inherits `numbers.Number`.

```
>>> param = Param('float_param', 0.5)
>>> param.value = 10
>>> param.value
10.0
>>> type(param.value)
<class 'float'>
```

property desc: str

Parameter description.

Type return

Return type str

property hyper_space: Union[hyperopt.pyll.base.Apply, matchzoo.engine.hyper_spaces.HyperoptProxy]

Hyper space of the parameter.

Type return

Return type Union[Apply, *HyperoptProxy*]

property name: str

Name of the parameter.

Type return

Return type str

reset()

Set the parameter’s value to *None*, which means “not set”.

This method bypasses validator.

Example

```
>>> import matchzoo as mz
>>> param = mz.Param(
...     name='str', validator=lambda x: isinstance(x, str))
>>> param.value = 'hello'
>>> param.value = None
Traceback (most recent call last):
...
ValueError: Validator not satisfied.
The validator's definition is as follows:
name='str', validator=lambda x: isinstance(x, str))
>>> param.reset()
>>> param.value is None
True
```

set_default(*val*, *verbose=1*)

Set default value, has no effect if already has a value.

Parameters

- **val** – Default value to set.
- **verbose** – Verbosity.

property validator: Callable[Any, bool]

Validator of the parameter.

Type return

Return type Callable[[Any], bool]

property value: Any

Value of the parameter.

Type return

Return type Any

matchzoo.engine.param_table module

Parameters table class.

class matchzoo.engine.param_table.ParamTable

Bases: object

Parameter table class.

Example

```

>>> params = ParamTable()
>>> params.add(Param('ham', 'Parma Ham'))
>>> params.add(Param('egg', 'Over Easy'))
>>> params['ham']
'Parma Ham'
>>> params['egg']
'Over Easy'
>>> print(params)
ham                Parma Ham
egg                Over Easy
>>> params.add(Param('egg', 'Sunny side Up'))
Traceback (most recent call last):
...
ValueError: Parameter named egg already exists.
To re-assign parameter egg value, use `params["egg"] = value` instead.

```

add(*param*)

Parameters *param* (*Param*) – parameter to add.

completed()

Return type bool

Returns *True* if all params are filled, *False* otherwise.

Example

```

>>> import matchzoo
>>> model = matchzoo.models.Naive()
>>> model.params.completed()
False
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.params.completed()
True

```

get(*key*)

Return type *Param*

Returns The parameter in the table named *key*.

property hyper_space: dict

Hyper space of the table, a valid *hyperopt* graph.

Type return

Return type dict

keys()

Return type *KeysView*

Returns Parameter table keys.

set(*key*, *param*)

Set *key* to parameter *param*.

to_frame()

Convert the parameter table into a pandas data frame.

Return type DataFrame

Returns A *pandas.DataFrame*.

Example

```
>>> import matchzoo as mz
>>> table = mz.ParamTable()
>>> table.add(mz.Param(name='x', value=10, desc='my x'))
>>> table.add(mz.Param(name='y', value=20, desc='my y'))
>>> table.to_frame()
  Name Description  Value Hyper-Space
0    x      my x     10         None
1    y      my y     20         None
```

update(*other*)

Update *self*.

Update *self* with the key/value pairs from *other*, overwriting existing keys. Notice that this does not add new keys to *self*.

This method is usually used by models to obtain useful information from a preprocessor's context.

Parameters **other** (dict) – The dictionary used update.

Example

```
>>> import matchzoo as mz
>>> model = mz.models.DenseBaseline()
>>> model.params['input_shapes'] is None
True
>>> prpr = model.get_default_preprocessor()
>>> _ = prpr.fit(mz.datasets.toy.load_data(), verbose=0)
>>> model.params.update(prpr.context)
>>> model.params['input_shapes']
[(30,), (30,)]
```

matchzoo.engine.parse_metric module

matchzoo.engine.parse_metric.**parse_metric**(metric, task=None)

Parse input metric in any form into a BaseMetric instance.

Parameters

- **metric** (Union[str, Type[BaseMetric], BaseMetric]) – Input metric in any form.
- **task** (Optional[BaseTask]) – Task type for determining specific metric.

Return type Union[BaseMetric, str]

Returns A BaseMetric instance

Examples::

```
>>> from matchzoo import metrics
>>> from matchzoo.engine.parse_metric import parse_metric
```

Use *str* as keras native metrics:

```
>>> parse_metric('mse')
'mse'
```

Use *str* as MatchZoo metrics:

```
>>> mz_metric = parse_metric('map')
>>> type(mz_metric)
<class 'matchzoo.metrics.mean_average_precision.MeanAveragePrecision'>
```

Use matchzoo.engine.BaseMetric subclasses as MatchZoo metrics:

```
>>> type(parse_metric(metrics.AveragePrecision))
<class 'matchzoo.metrics.average_precision.AveragePrecision'>
```

Use matchzoo.engine.BaseMetric instances as MatchZoo metrics:

```
>>> type(parse_metric(metrics.AveragePrecision()))
<class 'matchzoo.metrics.average_precision.AveragePrecision'>
```

Module contents

matchzoo.layers package

Submodules

matchzoo.layers.dynamic_pooling_layer module

matchzoo.layers.matching_layer module

Module contents

matchzoo.losses package

Submodules

matchzoo.losses.rank_cross_entropy_loss module

matchzoo.losses.rank_hinge_loss module

Module contents

matchzoo.metrics package

Submodules

matchzoo.metrics.average_precision module

Average precision metric for ranking.

```
class matchzoo.metrics.average_precision.AveragePrecision(threshold=0.0)
```

Bases: *matchzoo.engine.base_metric.BaseMetric*

Average precision metric.

```
ALIAS = ['average_precision', 'ap']
```

matchzoo.metrics.discounted_cumulative_gain module

Discounted cumulative gain metric for ranking.

```
class matchzoo.metrics.discounted_cumulative_gain.DiscountedCumulativeGain(k=1,
                                                                              threshold=0.0)
```

Bases: *matchzoo.engine.base_metric.BaseMetric*

Disconunted cumulative gain metric.

```
ALIAS = ['discounted_cumulative_gain', 'dcg']
```

matchzoo.metrics.mean_average_precision module

Mean average precision metric for ranking.

```
class matchzoo.metrics.mean_average_precision.MeanAveragePrecision(threshold=0.0)
```

Bases: *matchzoo.engine.base_metric.BaseMetric*

Mean average precision metric.

```
ALIAS = ['mean_average_precision', 'map']
```

matchzoo.metrics.mean_reciprocal_rank module

Mean reciprocal ranking metric.

```
class matchzoo.metrics.mean_reciprocal_rank.MeanReciprocalRank(threshold=0.0)
```

Bases: *matchzoo.engine.base_metric.BaseMetric*

Mean reciprocal rank metric.

```
ALIAS = ['mean_reciprocal_rank', 'mrr']
```

matchzoo.metrics.normalized_discounted_cumulative_gain module

Normalized discounted cumulative gain metric for ranking.

```
class matchzoo.metrics.normalized_discounted_cumulative_gain.NormalizedDiscountedCumulativeGain(k=1, threshold=0.0)
```

Bases: *matchzoo.engine.base_metric.BaseMetric*

Normalized discounted cumulative gain metric.

```
ALIAS = ['normalized_discounted_cumulative_gain', 'ndcg']
```

matchzoo.metrics.precision module

Precision for ranking.

```
class matchzoo.metrics.precision.Precision(k=1, threshold=0.0)
```

Bases: *matchzoo.engine.base_metric.BaseMetric*

Precision metric.

```
ALIAS = 'precision'
```

Module contents

```
matchzoo.metrics.list_available()
```

Return type list

matchzoo.models package

Submodules

matchzoo.models.anmm module

matchzoo.models.arci module

matchzoo.models.arcii module

`matchzoo.models.cdssm` module

`matchzoo.models.conv_knrm` module

`matchzoo.models.dense_baseline` module

`matchzoo.models.drmm` module

`matchzoo.models.drmmtns` module

`matchzoo.models.dssm` module

`matchzoo.models.duet` module

`matchzoo.models.knrm` module

`matchzoo.models.match_pyramid` module

`matchzoo.models.mvlstm` module

`matchzoo.models.naive` module

`matchzoo.models.parameter_readme_generator` module

Module contents

`matchzoo.preprocessors` package

Subpackages

`matchzoo.preprocessors.units` package

Submodules

`matchzoo.preprocessors.units.digit_removal` module

class `matchzoo.preprocessors.units.digit_removal.DigitRemoval`

Bases: `matchzoo.preprocessors.units.unit.Unit`

Process unit to remove digits.

transform(*input_*)

Remove digits from list of tokens.

Parameters **input** – list of tokens to be filtered.

Return tokens tokens of tokens without digits.

Return type list

matchzoo.preprocessors.units.fixed_length module

class matchzoo.preprocessors.units.fixed_length.FixedLength(*text_length*, *pad_value*=0, *pad_mode*='pre', *truncate_mode*='pre')

Bases: *matchzoo.preprocessors.units.unit.Unit*

FixedLengthUnit Class.

Process unit to get the fixed length text.

Examples

```
>>> from matchzoo.preprocessors.units import FixedLength
>>> fixedlen = FixedLength(3)
>>> fixedlen.transform(list(range(1, 6))) == [3, 4, 5]
True
>>> fixedlen.transform(list(range(1, 3))) == [0, 1, 2]
True
```

transform(*input_*)

Transform list of tokenized tokens into the fixed length text.

Parameters *input* – list of tokenized tokens.

Return tokens list of tokenized tokens in fixed length.

Return type list

matchzoo.preprocessors.units.frequency_filter module

class matchzoo.preprocessors.units.frequency_filter.FrequencyFilter(*low*=0, *high*=inf, *mode*='df')

Bases: *matchzoo.preprocessors.units.stateful_unit.StatefulUnit*

Frequency filter unit.

Parameters

- **low** (float) – Lower bound, inclusive.
- **high** (float) – Upper bound, exclusive.
- **mode** (str) – One of *tf* (term frequency), *df* (document frequency), and *idf* (inverse document frequency).

Examples::

```
>>> import matchzoo as mz
```

To filter based on term frequency (tf):

```
>>> tf_filter = mz.preprocessors.units.FrequencyFilter(
...     low=2, mode='tf')
>>> tf_filter.fit(['A', 'B', 'B'], ['C', 'C', 'C'])
>>> tf_filter.transform(['A', 'B', 'C'])
['B', 'C']
```

To filter based on document frequency (df):

```
>>> tf_filter = mz.preprocessors.units.FrequencyFilter(
...     low=2, mode='df')
>>> tf_filter.fit(['A', 'B'], ['B', 'C'])
>>> tf_filter.transform(['A', 'B', 'C'])
['B']
```

To filter based on inverse document frequency (idf):

```
>>> idf_filter = mz.preprocessors.units.FrequencyFilter(
...     low=1.2, mode='idf')
>>> idf_filter.fit(['A', 'B'], ['B', 'C', 'D'])
>>> idf_filter.transform(['A', 'B', 'C'])
['A', 'C']
```

fit(*list_of_tokens*)

Fit *list_of_tokens* by calculating *mode* states.

transform(*input_*)

Transform a list of tokens by filtering out unwanted words.

Return type list

matchzoo.preprocessors.units.lemmatization module

class matchzoo.preprocessors.units.lemmatization.Lemmatization

Bases: [matchzoo.preprocessors.units.unit.Unit](#)

Process unit for token lemmatization.

transform(*input_*)

Lemmatization a sequence of tokens.

Parameters **input** – list of tokens to be lemmatized.

Return tokens list of lemmatized tokens.

Return type list

matchzoo.preprocessors.units.lowercase module

class matchzoo.preprocessors.units.lowercase.Lowercase

Bases: [matchzoo.preprocessors.units.unit.Unit](#)

Process unit for text lower case.

transform(*input_*)

Convert list of tokens to lower case.

Parameters **input** – list of tokens.

Return tokens lower-cased list of tokens.

Return type list

matchzoo.preprocessors.units.matching_histogram module

class matchzoo.preprocessors.units.matching_histogram.**MatchingHistogram**(*bin_size=30, embedding_matrix=None, normalize=True, mode='LCH'*)

Bases: [matchzoo.preprocessors.units.unit.Unit](#)

MatchingHistogramUnit Class.

Parameters

- **bin_size** (int) – The number of bins of the matching histogram.
- **embedding_matrix** – The word embedding matrix applied to calculate the matching histogram.
- **normalize** – Boolean, normalize the embedding or not.
- **mode** (str) – The type of the histogram, it should be one of ‘CH’, ‘NG’, or ‘LCH’.

Examples

```
>>> embedding_matrix = np.array([[1.0, -1.0], [1.0, 2.0], [1.0, 3.0]])
>>> text_left = [0, 1]
>>> text_right = [1, 2]
>>> histogram = MatchingHistogram(3, embedding_matrix, True, 'CH')
>>> histogram.transform([text_left, text_right])
[[3.0, 1.0, 1.0], [1.0, 2.0, 2.0]]
```

transform(*input_*)

Transform the input text.

Return type list

matchzoo.preprocessors.units.ngram_letter module

class matchzoo.preprocessors.units.ngram_letter.**NgramLetter**(*ngram=3, reduce_dim=True*)

Bases: [matchzoo.preprocessors.units.unit.Unit](#)

Process unit for n-letter generation.

Triletter is used in DSSMModel. This processor is expected to execute before *Vocab* has been created.

Examples

```
>>> triletter = NgramLetter()
>>> rv = triletter.transform(['hello', 'word'])
>>> len(rv)
9
>>> rv
['#he', 'hel', 'ell', 'llo', 'lo#', '#wo', 'wor', 'ord', 'rd#']
>>> triletter = NgramLetter(reduce_dim=False)
>>> rv = triletter.transform(['hello', 'word'])
```

(continues on next page)

(continued from previous page)

```
>>> len(rv)
2
>>> rv
[['#he', 'hel', 'ell', 'llo', 'lo#'], ['#wo', 'wor', 'ord', 'rd#']]
```

transform(*input_*)

Transform token into tri-letter.

For example, *word* should be represented as *#wo*, *wor*, *ord* and *rd#*.**Parameters** *input* – list of tokens to be transformed.**Return** *n_letters* generated *n_letters*.**Return type** list

matchzoo.preprocessors.units.punc_removal module

class matchzoo.preprocessors.units.punc_removal.PuncRemovalBases: [matchzoo.preprocessors.units.unit.Unit](#)

Process unit for remove punctuations.

transform(*input_*)

Remove punctuations from list of tokens.

Parameters *input* – list of tokens.**Return** *rv* tokens without punctuation.**Return type** list

matchzoo.preprocessors.units.stateful_unit module

class matchzoo.preprocessors.units.stateful_unit.StatefulUnitBases: [matchzoo.preprocessors.units.unit.Unit](#)

Unit with inner state.

Usually need to be fit before transforming. All information gathered in the fit phrase will be stored into its *context*.**property** contextGet current context. Same as *unit.state*.**abstract** fit(*input_*)

Abstract base method, need to be implemented in subclass.

property stateGet current context. Same as *unit.context*.Deprecated since v2.2.0, and will be removed in the future. Used *unit.context* instead.

matchzoo.preprocessors.units.stemming module

class matchzoo.preprocessors.units.stemming.Stemming(*stemmer='porter'*)

Bases: [matchzoo.preprocessors.units.unit.Unit](#)

Process unit for token stemming.

Parameters **stemmer** – stemmer to use, *porter* or *lancaster*.

transform(*input_*)

Reducing inflected words to their word stem, base or root form.

Parameters **input** – list of string to be stemmed.

Return type list

matchzoo.preprocessors.units.stop_removal module

class matchzoo.preprocessors.units.stop_removal.StopRemoval(*lang='english'*)

Bases: [matchzoo.preprocessors.units.unit.Unit](#)

Process unit to remove stop words.

Example

```
>>> unit = StopRemoval()
>>> unit.transform(['a', 'the', 'test'])
['test']
>>> type(unit.stopwords)
<class 'list'>
```

property stopwords: list

Get stopwords based on language.

Params **lang** language code.

Return type list

Returns list of stop words.

transform(*input_*)

Remove stopwords from list of tokenized tokens.

Parameters

- **input** – list of tokenized tokens.
- **lang** – language code for stopwords.

Return tokens list of tokenized tokens without stopwords.

Return type list

matchzoo.preprocessors.units.tokenize module

matchzoo.preprocessors.units.unit module

```
class matchzoo.preprocessors.units.unit.Unit
    Bases: object

    Process unit do not persive state (i.e. do not need fit).

    abstract transform(input_)
        Abstract base method, need to be implemented in subclass.
```

matchzoo.preprocessors.units.vocabulary module

matchzoo.preprocessors.units.word_hashing module

Module contents

Submodules

matchzoo.preprocessors.basic_preprocessor module

matchzoo.preprocessors.build_unit_from_data_pack module

matchzoo.preprocessors.build_vocab_unit module

matchzoo.preprocessors.cdssm_preprocessor module

matchzoo.preprocessors.chain_transform module

matchzoo.preprocessors.dssm_preprocessor module

matchzoo.preprocessors.naive_preprocessor module

Module contents

matchzoo.tasks package

Submodules

matchzoo.tasks.classification module

Classification task.

```
class matchzoo.tasks.classification.Classification(num_classes=2, **kwargs)
    Bases: matchzoo.engine.base_task.BaseTask

    Classification task.
```

Examples

```
>>> classification_task = Classification(num_classes=2)
>>> classification_task.metrics = ['precision']
>>> classification_task.num_classes
2
>>> classification_task.output_shape
(2,)
>>> classification_task.output_dtype
<class 'int'>
>>> print(classification_task)
Classification Task with 2 classes
```

classmethod `list_available_losses()`

Return type list

Returns a list of available losses.

classmethod `list_available_metrics()`

Return type list

Returns a list of available metrics.

property `num_classes: int`
number of classes to classify.

Type return

Return type int

property `output_dtype`
target data type, expect *int* as output.

Type return

property `output_shape: tuple`
output shape of a single sample of the task.

Type return

Return type tuple

matchzoo.tasks.ranking module

Ranking task.

class `matchzoo.tasks.ranking.Ranking(loss=None, metrics=None)`

Bases: `matchzoo.engine.base_task.BaseTask`

Ranking Task.

Examples

```
>>> ranking_task = Ranking()
>>> ranking_task.metrics = ['map', 'ndcg']
>>> ranking_task.output_shape
(1,)
>>> ranking_task.output_dtype
<class 'float'>
>>> print(ranking_task)
Ranking Task
```

classmethod `list_available_losses()`

Return type list

Returns a list of available losses.

classmethod `list_available_metrics()`

Return type list

Returns a list of available metrics.

property `output_dtype`

target data type, expect *float* as output.

Type return

property `output_shape: tuple`

output shape of a single sample of the task.

Type return

Return type tuple

Module contents

matchzoo.utils package

Submodules

matchzoo.utils.list_recursive_subclasses module

`matchzoo.utils.list_recursive_subclasses.list_recursive_concrete_subclasses(base)`

List all concrete subclasses of *base* recursively.

matchzoo.utils.make_keras_optimizer_picklable module

matchzoo.utils.one_hot module

One hot vectors.

`matchzoo.utils.one_hot.one_hot(indices, num_classes)`

Return type ndarray

Returns A one-hot encoded vector.

matchzoo.utils.tensor_type module

Define Keras tensor type.

Module contents

1.1.2 Submodules

1.1.3 matchzoo.version module

1.1.4 Module contents

MATCHZOO MODEL REFERENCE

2.1 Naive

2.1.1 Model Documentation

Naive model with a simplest structure for testing purposes.

Bare minimum functioning model. The best choice to get things rolling. The worst choice to fit and evaluate performance.

2.1.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.naive.Naive'>	
1	input_shape	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	choice in ['adam', 'adagrad', 'rmsprop']

2.2 DSSM

2.2.1 Model Documentation

Deep structured semantic model.

Examples:

```
>>> model = DSSM()
>>> model.params['mlp_num_layers'] = 3
>>> model.params['mlp_num_units'] = 300
>>> model.params['mlp_num_fan_out'] = 128
>>> model.params['mlp_activation_func'] = 'relu'
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
```

2.2.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.dssm.DSSM'>	
1	in-put_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_multi_layer_perceptron	A flag of whether a multiple layer perceptron is used. Shouldn't be changed.	True	
5	mlp_num_units	Number of units in first <i>mlp_num_layers</i> layers.	128	quantitative uniform distribution in [8, 256), with a step size of 8
6	mlp_num_layers	Number of layers of the multiple layer perceptron.	3	quantitative uniform distribution in [1, 6), with a step size of 1
7	mlp_num_fans	Number of units of the layer that connects the multiple layer perceptron and the output.	64	quantitative uniform distribution in [4, 128), with a step size of 4
8	mlp_activation_func	Activation function used in the multiple layer perceptron.	relu	

2.3 CDSSM

2.3.1 Model Documentation

CDSSM Model implementation.

Learning Semantic Representations Using Convolutional Neural Networks for Web Search. (2014a) A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. (2014b)

Examples:

```
>>> model = CDSSM()
>>> model.params['optimizer'] = 'adam'
>>> model.params['filters'] = 32
>>> model.params['kernel_size'] = 3
>>> model.params['conv_activation_func'] = 'relu'
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
```

2.3.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.cdssm.CDSSM'>	CDSSM
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_multi_layer_perceptron	A flag to whether a multiple layer perceptron is used. Shouldn't be changed.	True	
5	mlp_num_units	Number of units in first <i>mlp_num_layers</i> layers.	128	quantitative uniform distribution in [8, 256), with a step size of 8
6	mlp_num_layers	Number of layers of the multiple layer perceptron.	3	quantitative uniform distribution in [1, 6), with a step size of 1
7	mlp_num_fan_out	Number of units of the layer that connects the multiple layer perceptron and the output.	64	quantitative uniform distribution in [4, 128), with a step size of 4
8	mlp_activation	Activation function used in the multiple layer perceptron.	relu	
9	filters	Number of filters in the 1D convolution layer.	32	
10	kernel_size	Number of kernel size in the 1D convolution layer.	3	
11	strides	Strides in the 1D convolution layer.	1	
12	padding	The padding mode in the convolution layer. It should be one of <i>same</i> , <i>valid</i> , and <i>causal</i> .	same	
13	conv_activation	Activation function in the convolution layer.	relu	
14	w_initializer		glorot_normal	
15	b_initializer		zeros	
16	dropout_rate	The dropout rate.	0.3	

2.4 DenseBaseline

2.4.1 Model Documentation

A simple densely connected baseline model.

Examples:

```
>>> model = DenseBaseline()
>>> model.params['mlp_num_layers'] = 2
>>> model.params['mlp_num_units'] = 300
>>> model.params['mlp_num_fan_out'] = 128
```

(continues on next page)

(continued from previous page)

```
>>> model.params['mlp_activation_func'] = 'relu'
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
>>> model.compile()
```

2.4.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.dense_baseline.DenseBaseline'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_multi_layer_perceptron	Whether a multiple layer perceptron is used. Shouldn't be changed.	True	
5	mlp_num_units	Number of units in first <i>mlp_num_layers</i> layers.	256	quantitative uniform distribution in [16, 512), with a step size of 1
6	mlp_num_layers	Number of layers of the multiple layer perceptron.	3	quantitative uniform distribution in [1, 5), with a step size of 1
7	mlp_num_fan_out	Number of units of the layer that connects the multiple layer perceptron and the output.	64	quantitative uniform distribution in [4, 128), with a step size of 4
8	mlp_activation_func	Activation function used in the multiple layer perceptron.	relu	

2.5 ArcI

2.5.1 Model Documentation

ArcI Model.

Examples:

```
>>> model = ArcI()
>>> model.params['num_blocks'] = 1
>>> model.params['left_filters'] = [32]
>>> model.params['right_filters'] = [32]
>>> model.params['left_kernel_sizes'] = [3]
>>> model.params['right_kernel_sizes'] = [3]
>>> model.params['left_pool_sizes'] = [2]
>>> model.params['right_pool_sizes'] = [4]
>>> model.params['conv_activation_func'] = 'relu'
```

(continues on next page)

(continued from previous page)

```
>>> model.params['mlp_num_layers'] = 1
>>> model.params['mlp_num_units'] = 64
>>> model.params['mlp_num_fan_out'] = 32
>>> model.params['mlp_activation_func'] = 'relu'
>>> model.params['dropout_rate'] = 0.5
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
```


2.5.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.arci.ArcI'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embeddings	A flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	with_multi_layer_perceptron	A flag to whether a multiple layer perceptron is used. Shouldn't be changed.	True	
9	mlp_num_units	Number of units in first <i>mlp_num_layers</i> layers.	128	quantitative uniform distribution in [8, 256), with a step size of 8
10	mlp_num_layers	Number of layers of the multiple layer perceptron.	3	quantitative uniform distribution in [1, 6), with a step size of 1
11	mlp_num_fan_out	Number of units of the layer that connects the multiple layer perceptron and the output.	64	quantitative uniform distribution in [4, 128), with a step size of 4
12	mlp_activation	Activation function used in the multiple layer perceptron.	relu	
13	num_blocks	Number of convolution blocks.	1	
14	left_filters	The filter size of each convolution blocks for the left input.	[32]	
15	left_kernel_size	The kernel size of each convolution blocks for the left input.	[3]	
16	right_filters	The filter size of each convolution blocks for the right input.	[32]	
17	right_kernel_size	The kernel size of each convolution blocks for the right input.	[3]	
18	conv_activation	The activation function in the convolution layer.	relu	
19	left_pool_size	The pooling size of each convolution blocks for the left input.	[2]	
20	right_pool_size	The pooling size of each convolution blocks for the right input.	[2]	
21	padding	The padding mode in the convolution layer. It should be one of <i>same</i> , <i>valid</i> , and <i>causal</i> .	same	choice in ['same', 'valid', 'causal']
22	dropout_rate	The dropout rate.	0.0	quantitative uniform distribution in [0.0, 0.8), with a step size of 0.01

2.6 ArcII

2.6.1 Model Documentation

ArcII Model.

Examples:

```
>>> model = ArcII()
>>> model.params['embedding_output_dim'] = 300
>>> model.params['num_blocks'] = 2
>>> model.params['kernel_1d_count'] = 32
>>> model.params['kernel_1d_size'] = 3
>>> model.params['kernel_2d_count'] = [16, 32]
>>> model.params['kernel_2d_size'] = [[3, 3], [3, 3]]
>>> model.params['pool_2d_size'] = [[2, 2], [2, 2]]
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
```

2.6.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.arcii.ArcII'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	choice in ['adam', 'rmsprop', 'adagrad']
4	with_embedding_flag	Flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	num_blocks	Number of 2D convolution blocks.	1	
9	kernel_1d_count	Kernel count of 1D convolution layer.	32	
10	kernel_1d_size	Kernel size of 1D convolution layer.	3	
11	kernel_2d_count	Kernel count of 2D convolution layer in each block	[32]	
12	kernel_2d_size	Kernel size of 2D convolution layer in each block.	[[3, 3]]	
13	activation	Activation function.	relu	
14	pool_2d_size	Size of pooling layer in each block.	[[2, 2]]	
15	padding	The padding mode in the convolution layer. It should be one of <i>same</i> , <i>valid</i> .	same	choice in ['same', 'valid']
16	dropout_rate	The dropout rate.	0.0	quantitative uniform distribution in [0.0, 0.8), with a step size of 0.01

2.7 MatchPyramid

2.7.1 Model Documentation

MatchPyramid Model.

Examples:

```
>>> model = MatchPyramid()
>>> model.params['embedding_output_dim'] = 300
>>> model.params['num_blocks'] = 2
>>> model.params['kernel_count'] = [16, 32]
>>> model.params['kernel_size'] = [[3, 3], [3, 3]]
```

(continues on next page)

(continued from previous page)

```
>>> model.params['dpool_size'] = [3, 10]
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
```

2.7.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.match_pyramid.MatchPyramid'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embeddings	Flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	num_blocks	Number of convolution blocks.	1	
9	kernel_count	The kernel count of the 2D convolution of each block.	[32]	
10	kernel_size	The kernel size of the 2D convolution of each block.	[[3, 3]]	
11	activation	The activation function.	relu	
12	dpool_size	The max-pooling size of each block.	[3, 10]	
13	padding	The padding mode in the convolution layer.	same	
14	dropout_rate	The dropout rate.	0.0	quantitative uniform distribution in [0.0, 0.8), with a step size of 0.01

2.8 KNRM

2.8.1 Model Documentation

KNRM model.

Examples:

```
>>> model = KNRM()
>>> model.params['embedding_input_dim'] = 10000
```

(continues on next page)

(continued from previous page)

```

>>> model.params['embedding_output_dim'] = 10
>>> model.params['embedding_trainable'] = True
>>> model.params['kernel_num'] = 11
>>> model.params['sigma'] = 0.1
>>> model.params['exact_sigma'] = 0.001
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()

```

2.8.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.knrm.KNRM'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embedding_flag	Flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	kernel_num	The number of RBF kernels.	11	quantitative uniform distribution in [5, 20), with a step size of 1
9	sigma	The <i>sigma</i> defines the kernel width.	0.1	quantitative uniform distribution in [0.01, 0.2), with a step size of 0.01
10	exact_sigma	The <i>exact_sigma</i> denotes the <i>sigma</i> for exact match.	0.001	

2.9 DUET

2.9.1 Model Documentation

DUET Model.

Examples:

```

>>> model = DUET()
>>> model.params['embedding_input_dim'] = 1000
>>> model.params['embedding_output_dim'] = 300
>>> model.params['lm_filters'] = 32

```

(continues on next page)

(continued from previous page)

```

>>> model.params['lm_hidden_sizes'] = [64, 32]
>>> model.params['dropout_rate'] = 0.5
>>> model.params['dm_filters'] = 32
>>> model.params['dm_kernel_size'] = 3
>>> model.params['dm_d_mpool'] = 4
>>> model.params['dm_hidden_sizes'] = [64, 32]
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()

```

2.9.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.duet.DUET'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embeddings	Flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	lm_filters	Filter size of 1D convolution layer in the local model.	32	
9	lm_hidden_size	A list of hidden size of the MLP layer in the local model.	[32]	
10	dm_filters	Filter size of 1D convolution layer in the distributed model.	32	
11	dm_kernel_size	Kernel size of 1D convolution layer in the distributed model.	3	
12	dm_q_hidden_size	Hidden size of the MLP layer for the left text in the distributed model.	32	
13	dm_d_mpool	Max pooling size for the right text in the distributed model.	3	
14	dm_hidden_size	A list of hidden size of the MLP layer in the distributed model.	[32]	
15	padding	The padding mode in the convolution layer. It should be one of <i>same</i> , <i>valid</i> , and <i>causal</i> .	same	
16	activation_func	Activation function in the convolution layer.	relu	
17	dropout_rate	The dropout rate.	0.5	quantitative uniform distribution in [0.0, 0.8), with a step size of 0.02

2.10 DRMMTKS

2.10.1 Model Documentation

DRMMTKS Model.

Examples:

```
>>> model = DRMMTKS()
>>> model.params['embedding_input_dim'] = 100000
>>> model.params['embedding_output_dim'] = 100
>>> model.params['top_k'] = 20
>>> model.params['mlp_num_layers'] = 1
>>> model.params['mlp_num_units'] = 5
>>> model.params['mlp_num_fan_out'] = 1
>>> model.params['mlp_activation_func'] = 'tanh'
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
```

2.10.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.drmmtns.DRMMTKS'>	
1	in-put_shapes	Dependent on the model and data. Should be set manually.	[(5,), (300,)]	
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embedding	A flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	with_multi_layer_perceptron	A flag to indicate whether a multiple layer perceptron is used. Shouldn't be changed.	True	
9	mlp_num_units	Number of units in first <i>mlp_num_layers</i> layers.	128	quantitative uniform distribution in [8, 256), with a step size of 8
10	mlp_num_layers	Number of layers of the multiple layer perceptron.	3	quantitative uniform distribution in [1, 6), with a step size of 1
11	mlp_num_fan_out	Number of units of the layer that connects the multiple layer perceptron and the output.	64	quantitative uniform distribution in [4, 128), with a step size of 4
12	mlp_activation	Activation function used in the multiple layer perceptron.	relu	
13	mask_value	The value to be masked from inputs.	-1	
14	top_k	Size of top-k pooling layer.	10	quantitative uniform distribution in [2, 100), with a step size of 1

2.11 DRMM

2.11.1 Model Documentation

DRMM Model.

Examples:

```
>>> model = DRMM()
>>> model.params['mlp_num_layers'] = 1
>>> model.params['mlp_num_units'] = 5
>>> model.params['mlp_num_fan_out'] = 1
```

(continues on next page)

(continued from previous page)

```
>>> model.params['mlp_activation_func'] = 'tanh'
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
>>> model.compile()
```

2.11.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.drmm.DRMM'>	
1	input_shapes	Dependent on the model and data. Should be set manually.	[(5,), (5, 30)]	
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embeddings	A flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	with_multi_layer_perceptron	A flag of whether a multiple layer perceptron is used. Shouldn't be changed.	True	
9	mlp_num_units	Number of units in first <i>mlp_num_layers</i> layers.	128	quantitative uniform distribution in [8, 256), with a step size of 8
10	mlp_num_layers	Number of layers of the multiple layer perceptron.	3	quantitative uniform distribution in [1, 6), with a step size of 1
11	mlp_num_fan_out	Number of units of the layer that connects the multiple layer perceptron and the output.	64	quantitative uniform distribution in [4, 128), with a step size of 4
12	mlp_activation_func	Activation function used in the multiple layer perceptron.	relu	
13	mask_value	The value to be masked from inputs.	-1	

2.12 ANMM

2.12.1 Model Documentation

ANMM Model.

Examples:

```
>>> model = ANMM()
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
```

2.12.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.anmm.ANMM'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embeddings	Flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	dropout_rate	The dropout rate.	0.1	quantitative uniform distribution in [0, 1), with a step size of 0.05
9	num_layers	Number of hidden layers in the MLP layer.	2	
10	hidden_sizes	Number of hidden size for each hidden layer	[30, 30]	

2.13 MVLSTM

2.13.1 Model Documentation

MVLSTM Model.

Examples:

```
>>> model = MVLSTM()
>>> model.params['lstm_units'] = 32
>>> model.params['top_k'] = 50
>>> model.params['mlp_num_layers'] = 2
>>> model.params['mlp_num_units'] = 20
>>> model.params['mlp_num_fan_out'] = 10
>>> model.params['mlp_activation_func'] = 'relu'
>>> model.params['dropout_rate'] = 0.5
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()
```

2.13.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.mvlstm.MVLSTM'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embeddings	A flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	with_multi_layer_perceptron	A flag to indicate whether a multiple layer perceptron is used. Shouldn't be changed.	True	
9	mlp_num_units	Number of units in first <i>mlp_num_layers</i> layers.	128	quantitative uniform distribution in [8, 256), with a step size of 8
10	mlp_num_layers	Number of layers of the multiple layer perceptron.	3	quantitative uniform distribution in [1, 6), with a step size of 1
11	mlp_num_fan_out	Number of units of the layer that connects the multiple layer perceptron and the output.	64	quantitative uniform distribution in [4, 128), with a step size of 4
12	mlp_activation	Activation function used in the multiple layer perceptron.	relu	
13	lstm_units	Integer, the hidden size in the bi-directional LSTM layer.	32	
14	dropout_rate	Float, the dropout rate.	0.0	
15	top_k	Integer, the size of top-k pooling layer.	10	quantitative uniform distribution in [2, 100), with a step size of 1

2.14 MatchLSTM

2.14.1 Model Documentation

Match LSTM model.

Examples:

```
>>> model = MatchLSTM()
>>> model.guess_and_fill_missing_params(verbose=0)
```

(continues on next page)

(continued from previous page)

```

>>> model.params['embedding_input_dim'] = 10000
>>> model.params['embedding_output_dim'] = 100
>>> model.params['embedding_trainable'] = True
>>> model.params['fc_num_units'] = 200
>>> model.params['lstm_num_units'] = 256
>>> model.params['dropout_rate'] = 0.5
>>> model.build()

```

2.14.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.contrib.models.match_lstm.MatchLSTM'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embeddings	Flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	lstm_num_units	The hidden size in the LSTM layer.	256	quantitative uniform distribution in [128, 384), with a step size of 32
9	fc_num_units	The hidden size in the full connection layer.	200	quantitative uniform distribution in [100, 300), with a step size of 20
10	dropout_rate	The dropout rate.	0.0	quantitative uniform distribution in [0.0, 0.9), with a step size of 0.01

2.15 ConvKNRM

2.15.1 Model Documentation

ConvKNRM model.

Examples:

```

>>> model = ConvKNRM()
>>> model.params['embedding_input_dim'] = 10000

```

(continues on next page)

(continued from previous page)

```

>>> model.params['embedding_output_dim'] = 300
>>> model.params['embedding_trainable'] = True
>>> model.params['filters'] = 128
>>> model.params['conv_activation_func'] = 'tanh'
>>> model.params['max_ngram'] = 3
>>> model.params['use_crossmatch'] = True
>>> model.params['kernel_num'] = 11
>>> model.params['sigma'] = 0.1
>>> model.params['exact_sigma'] = 0.001
>>> model.guess_and_fill_missing_params(verbose=0)
>>> model.build()

```

2.15.2 Model Hyper Parameters

	Name	Description	Default Value	Default Hyper-Space
0	model_class	Model class. Used internally for save/load. Changing this may cause unexpected behaviors.	<class 'match-zoo.models.conv_knrm.ConvKNRM'>	
1	input_shapes	Dependent on the model and data. Should be set manually.		
2	task	Decides model output shape, loss, and metrics.		
3	optimizer		adam	
4	with_embeddings	Any flag used help <i>auto</i> module. Shouldn't be changed.	True	
5	embedding_input_dim	Usually equals vocab size + 1. Should be set manually.		
6	embedding_output_dim	Should be set manually.		
7	embedding_trainable	<i>True</i> to enable embedding layer training, <i>False</i> to freeze embedding parameters.	True	
8	kernel_num	The number of RBF kernels.	11	quantitative uniform distribution in [5, 20), with a step size of 1
9	sigma	The <i>sigma</i> defines the kernel width.	0.1	quantitative uniform distribution in [0.01, 0.2), with a step size of 0.01
10	exact_sigma	The <i>exact_sigma</i> denotes the <i>sigma</i> for exact match.	0.001	
11	filters	The filter size in the convolution layer.	128	
12	conv_activation_func	The activation function in the convolution layer.	relu	
13	max_ngram	The maximum length of n-grams for the convolution layer.	3	
14	use_crossmatch	Whether to match left n-grams and right n-grams of different lengths	True	

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

m

`matchzoo.utils.tensor_type`, 31

`matchzoo.data_pack`, 10
`matchzoo.data_pack.data_pack`, 4
`matchzoo.data_pack.pack`, 9
`matchzoo.engine`, 19
`matchzoo.engine.base_metric`, 11
`matchzoo.engine.base_task`, 11
`matchzoo.engine.hyper_spaces`, 12
`matchzoo.engine.param`, 14
`matchzoo.engine.param_table`, 16
`matchzoo.engine.parse_metric`, 19
`matchzoo.metrics`, 21
`matchzoo.metrics.average_precision`, 20
`matchzoo.metrics.discounted_cumulative_gain`,
20
`matchzoo.metrics.mean_average_precision`, 20
`matchzoo.metrics.mean_reciprocal_rank`, 21
`matchzoo.metrics.normalized_discounted_cumulative_gain`,
21
`matchzoo.metrics.precision`, 21
`matchzoo.preprocessors.units.digit_removal`,
22
`matchzoo.preprocessors.units.fixed_length`, 23
`matchzoo.preprocessors.units.frequency_filter`,
23
`matchzoo.preprocessors.units.lemmatization`,
24
`matchzoo.preprocessors.units.lowercase`, 24
`matchzoo.preprocessors.units.matching_histogram`,
25
`matchzoo.preprocessors.units.ngram_letter`, 25
`matchzoo.preprocessors.units.punc_removal`, 26
`matchzoo.preprocessors.units.stateful_unit`,
26
`matchzoo.preprocessors.units.stemming`, 27
`matchzoo.preprocessors.units.stop_removal`, 27
`matchzoo.preprocessors.units.unit`, 28
`matchzoo.tasks`, 30
`matchzoo.tasks.classification`, 28
`matchzoo.tasks.ranking`, 29
`matchzoo.utils.list_recursive_subclasses`, 30
`matchzoo.utils.one_hot`, 31

INDEX

A

`add()` (*matchzoo.engine.param_table.ParamTable* method), 17

`ALIAS` (*matchzoo.engine.base_metric.BaseMetric* attribute), 11

`ALIAS` (*matchzoo.metrics.average_precision.AveragePrecision* attribute), 20

`ALIAS` (*matchzoo.metrics.discounted_cumulative_gain.DiscountedCumulativeGain* attribute), 20

`ALIAS` (*matchzoo.metrics.mean_average_precision.MeanAveragePrecision* attribute), 20

`ALIAS` (*matchzoo.metrics.mean_reciprocal_rank.MeanReciprocalRank* attribute), 21

`ALIAS` (*matchzoo.metrics.normalized_discounted_cumulative_gain.NormalizedDiscountedCumulativeGain* attribute), 21

`ALIAS` (*matchzoo.metrics.precision.Precision* attribute), 21

`append_text_length()` (*matchzoo.data_pack.data_pack.DataPack* method), 5

`apply_on_text()` (*matchzoo.data_pack.data_pack.DataPack* method), 6

`AveragePrecision` (class in *matchzoo.metrics.average_precision*), 20

B

`BaseMetric` (class in *matchzoo.engine.base_metric*), 11

`BaseTask` (class in *matchzoo.engine.base_task*), 11

C

`choice` (class in *matchzoo.engine.hyper_spaces*), 13

`Classification` (class in *matchzoo.tasks.classification*), 28

`completed()` (*matchzoo.engine.param_table.ParamTable* method), 17

`context` (*matchzoo.preprocessors.units.stateful_unit.StatefulUnit* property), 26

`convert()` (*matchzoo.engine.hyper_spaces.HyperoptProxy* method), 13

`copy()` (*matchzoo.data_pack.data_pack.DataPack* method), 7

D

`DATA_FILENAME` (*matchzoo.data_pack.data_pack.DataPack* attribute), 5

`DataPack` (class in *matchzoo.data_pack.data_pack*), 4

`DataPack.FrameView` (class in *matchzoo.data_pack.data_pack*), 5

`DiscountedCumulativeGain` (*matchzoo.engine.param.Param* property), 15

`DigitRemoval` (class in *matchzoo.preprocessors.units.digit_removal*), 22

`DiscountedCumulativeGain` (class in *matchzoo.metrics.discounted_cumulative_gain*), 20

`drop_invalid()` (*matchzoo.data_pack.data_pack.DataPack* method), 7

`drop_label()` (*matchzoo.data_pack.data_pack.DataPack* method), 7

F

`fit()` (*matchzoo.preprocessors.units.frequency_filter.FrequencyFilter* method), 24

`fit()` (*matchzoo.preprocessors.units.stateful_unit.StatefulUnit* method), 26

`FixedLength` (class in *matchzoo.preprocessors.units.fixed_length*), 23

`frame` (*matchzoo.data_pack.data_pack.DataPack* property), 7

`FrequencyFilter` (class in *matchzoo.preprocessors.units.frequency_filter*), 23

G

`get()` (*matchzoo.engine.param_table.ParamTable* method), 17

H

`has_label` (*matchzoo.data_pack.data_pack.DataPack* property), 8

`hyper_space` (*matchzoo.engine.param.Param* property), 15

`hyper_space` (*matchzoo.engine.param_table.ParamTable* property), 17
`HyperoptProxy` (class in *matchzoo.engine.hyper_spaces*), 12

K

`keys()` (*matchzoo.engine.param_table.ParamTable* method), 17

L

`left` (*matchzoo.data_pack.data_pack.DataPack* property), 8
`Lemmatization` (class in *matchzoo.preprocessors.units.lemmatization*), 24
`list_available()` (in module *matchzoo.metrics*), 21
`list_available_losses()` (*matchzoo.engine.base_task.BaseTask* class method), 11
`list_available_losses()` (*matchzoo.tasks.classification.Classification* class method), 29
`list_available_losses()` (*matchzoo.tasks.ranking.Ranking* class method), 30
`list_available_metrics()` (*matchzoo.engine.base_task.BaseTask* class method), 12
`list_available_metrics()` (*matchzoo.tasks.classification.Classification* class method), 29
`list_available_metrics()` (*matchzoo.tasks.ranking.Ranking* class method), 30
`list_recursive_concrete_subclasses()` (in module *matchzoo.utils.list_recursive_subclasses*), 30
`load_data_pack()` (in module *matchzoo.data_pack.data_pack*), 9
`loss` (*matchzoo.engine.base_task.BaseTask* property), 12
`Lowercase` (class in *matchzoo.preprocessors.units.lowercase*), 24

M

`MatchingHistogram` (class in *matchzoo.preprocessors.units.matching_histogram*), 25
`matchzoo.data_pack` module, 10
`matchzoo.data_pack.data_pack` module, 4
`matchzoo.data_pack.pack` module, 9
`matchzoo.engine` module, 19

`matchzoo.engine.base_metric` module, 11
`matchzoo.engine.base_task` module, 11
`matchzoo.engine.hyper_spaces` module, 12
`matchzoo.engine.param` module, 14
`matchzoo.engine.param_table` module, 16
`matchzoo.engine.parse_metric` module, 19
`matchzoo.metrics` module, 21
`matchzoo.metrics.average_precision` module, 20
`matchzoo.metrics.discounted_cumulative_gain` module, 20
`matchzoo.metrics.mean_average_precision` module, 20
`matchzoo.metrics.mean_reciprocal_rank` module, 21
`matchzoo.metrics.normalized_discounted_cumulative_gain` module, 21
`matchzoo.metrics.precision` module, 21
`matchzoo.preprocessors.units.digit_removal` module, 22
`matchzoo.preprocessors.units.fixed_length` module, 23
`matchzoo.preprocessors.units.frequency_filter` module, 23
`matchzoo.preprocessors.units.lemmatization` module, 24
`matchzoo.preprocessors.units.lowercase` module, 24
`matchzoo.preprocessors.units.matching_histogram` module, 25
`matchzoo.preprocessors.units.ngram_letter` module, 25
`matchzoo.preprocessors.units.punc_removal` module, 26
`matchzoo.preprocessors.units.stateful_unit` module, 26
`matchzoo.preprocessors.units.stemming` module, 27
`matchzoo.preprocessors.units.stop_removal` module, 27
`matchzoo.preprocessors.units.unit` module, 28
`matchzoo.tasks` module, 30
`matchzoo.tasks.classification` module, 28

[matchzoo.tasks.ranking](#)
[module](#), 29
[matchzoo.utils.list_recursive_subclasses](#)
[module](#), 30
[matchzoo.utils.one_hot](#)
[module](#), 31
[matchzoo.utils.tensor_type](#)
[module](#), 31
[MeanAveragePrecision](#) (class in [matchzoo.metrics.mean_average_precision](#)), 20
[MeanReciprocalRank](#) (class in [matchzoo.metrics.mean_reciprocal_rank](#)), 21
[metrics](#) ([matchzoo.engine.base_task.BaseTask](#) property), 12
[module](#)
[matchzoo.data_pack](#), 10
[matchzoo.data_pack.data_pack](#), 4
[matchzoo.data_pack.pack](#), 9
[matchzoo.engine](#), 19
[matchzoo.engine.base_metric](#), 11
[matchzoo.engine.base_task](#), 11
[matchzoo.engine.hyper_spaces](#), 12
[matchzoo.engine.param](#), 14
[matchzoo.engine.param_table](#), 16
[matchzoo.engine.parse_metric](#), 19
[matchzoo.metrics](#), 21
[matchzoo.metrics.average_precision](#), 20
[matchzoo.metrics.discounted_cumulative_gain](#), 20
[matchzoo.metrics.mean_average_precision](#), 20
[matchzoo.metrics.mean_reciprocal_rank](#), 21
[matchzoo.metrics.normalized_discounted_cumulative_gain](#), 21
[matchzoo.metrics.precision](#), 21
[matchzoo.preprocessors.units.digit_removal](#), 22
[matchzoo.preprocessors.units.fixed_length](#), 23
[matchzoo.preprocessors.units.frequency_filter](#), 23
[matchzoo.preprocessors.units.lemmatization](#), 24
[matchzoo.preprocessors.units.lowercase](#), 24
[matchzoo.preprocessors.units.matching_histogram](#), 25
[matchzoo.preprocessors.units.ngram_letter](#), 25
[matchzoo.preprocessors.units.punc_removal](#), 26
[matchzoo.preprocessors.units.stateful_unit](#), 26
[matchzoo.preprocessors.units.stemming](#), 27
[matchzoo.preprocessors.units.stop_removal](#), 27
[matchzoo.preprocessors.units.unit](#), 28
[matchzoo.tasks](#), 30
[matchzoo.tasks.classification](#), 28
[matchzoo.tasks.ranking](#), 29
[matchzoo.utils.list_recursive_subclasses](#), 30
[matchzoo.utils.one_hot](#), 31
[matchzoo.utils.tensor_type](#), 31

N

[name](#) ([matchzoo.engine.param.Param](#) property), 15
[NgramLetter](#) (class in [matchzoo.preprocessors.units.ngram_letter](#)), 25
[NormalizedDiscountedCumulativeGain](#) (class in [matchzoo.metrics.normalized_discounted_cumulative_gain](#)), 21
[num_classes](#) ([matchzoo.tasks.classification.Classification](#) property), 29

O

[one_hot\(\)](#) (in module [matchzoo.utils.one_hot](#)), 31
[one_hot_encode_label\(\)](#) ([matchzoo.data_pack.data_pack.DataPack](#) method), 8
[output_dtype](#) ([matchzoo.engine.base_task.BaseTask](#) property), 12
[output_dtype](#) ([matchzoo.tasks.classification.Classification](#) property), 29
[output_dtype](#) ([matchzoo.tasks.ranking.Ranking](#) property), 30
[output_shape](#) ([matchzoo.engine.base_task.BaseTask](#) property), 12
[output_shape](#) ([matchzoo.tasks.classification.Classification](#) property), 29
[output_shape](#) ([matchzoo.tasks.ranking.Ranking](#) property), 30

P

[pack\(\)](#) (in module [matchzoo.data_pack.pack](#)), 9
[Param](#) (class in [matchzoo.engine.param](#)), 14
[ParamTable](#) (class in [matchzoo.engine.param_table](#)), 16
[parse_metric\(\)](#) (in module [matchzoo.engine.parse_metric](#)), 19
[Precision](#) (class in [matchzoo.metrics.precision](#)), 21
[PuncRemoval](#) (class in [matchzoo.preprocessors.units.punc_removal](#)), 26

Q

[quniform](#) (class in [matchzoo.engine.hyper_spaces](#)), 13

R

Ranking (class in `matchzoo.tasks.ranking`), 29

relation (`matchzoo.data_pack.data_pack.DataPack` property), 8

reset() (`matchzoo.engine.param.Param` method), 15

right (`matchzoo.data_pack.data_pack.DataPack` property), 8

S

sample() (in module `matchzoo.engine.hyper_spaces`), 13

save() (`matchzoo.data_pack.data_pack.DataPack` method), 8

set() (`matchzoo.engine.param_table.ParamTable` method), 18

set_default() (`matchzoo.engine.param.Param` method), 16

shuffle() (`matchzoo.data_pack.data_pack.DataPack` method), 8

sort_and_couple() (in module `matchzoo.engine.base_metric`), 11

state (`matchzoo.preprocessors.units.stateful_unit.StatefulUnit` property), 26

StatefulUnit (class in `matchzoo.preprocessors.units.stateful_unit`), 26

Stemming (class in `matchzoo.preprocessors.units.stemming`), 27

StopRemoval (class in `matchzoo.preprocessors.units.stop_removal`), 27

stopwords (`matchzoo.preprocessors.units.stop_removal.StopRemoval` property), 27

T

to_frame() (`matchzoo.engine.param_table.ParamTable` method), 18

transform() (`matchzoo.preprocessors.units.digit_removal.DigitRemoval` method), 22

transform() (`matchzoo.preprocessors.units.fixed_length.FixedLength` method), 23

transform() (`matchzoo.preprocessors.units.frequency_filter.FrequencyFilter` method), 24

transform() (`matchzoo.preprocessors.units.lemmatization.Lemmatization` method), 24

transform() (`matchzoo.preprocessors.units.lowercase.Lowercase` method), 24

transform() (`matchzoo.preprocessors.units.matching_histogram.MatchingHistogram` method), 25

transform() (`matchzoo.preprocessors.units.ngram_letter.NgramLetter` method), 26

transform() (`matchzoo.preprocessors.units.punc_removal.PuncRemoval` method), 26

transform() (`matchzoo.preprocessors.units.stemming.Stemming` method), 27

transform() (`matchzoo.preprocessors.units.stop_removal.StopRemoval` method), 27

transform() (`matchzoo.preprocessors.units.unit.Unit` method), 28

U

uniform (class in `matchzoo.engine.hyper_spaces`), 13

Unit (class in `matchzoo.preprocessors.units.unit`), 28

unpack() (`matchzoo.data_pack.data_pack.DataPack` method), 9

update() (`matchzoo.engine.param_table.ParamTable` method), 18

V

validator (`matchzoo.engine.param.Param` property), 16

value (`matchzoo.engine.param.Param` property), 16