tf.keras.preprocessing.text.Tokenizer

Contents

Class `Tokenizer`

Defined in [`tensorflow/python/keras/_impl/keras/preprocessing/text.py`](tensorflow/python/keras/_impl/keras/preprocessing/text.py).

Text tokenization utility class.

This class allows to vectorize a text corpus, by turning each text into either a sequence of integers (each integer being the index of a token in a dictionary) or into a vector where the coefficient for each token could be binary, based on word count, based on tf-idf...

Arguments:

- `num_words`: the maximum number of words to keep, based on word frequency. Only the most common **`num_words`** words will be kept.
- `filters`: a string where each element is a character that will be filtered from the texts. The default is all punctuation, plus tabs and line breaks, minus the `'` character.
- `lower`: boolean. Whether to convert the texts to lowercase.
- `split`: character or string to use for token splitting.
- `char_level`: if True, every character will be treated as a token.

By default, all punctuation is removed, turning the texts into space-separated sequences of words (words maybe include the `'` character). These sequences are then split into lists of tokens. They will then be indexed or vectorized.

`0` is a reserved index that won't be assigned to any word.

Methods

`__init__`

```
__init__(
    num_words=None,
    filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n',
    lower=True,
    split=' ',
    char_level=False
)
```

`fit_on_sequences`

```
fit_on_sequences(sequences)
```

Updates internal vocabulary based on a list of sequences.

Required before using **`sequences_to_matrix`** (if **`fit_on_texts`** was never called).

Arguments:

- `sequences`: A list of sequence. A "sequence" is a list of integer word indices.

`fit_on_texts`

```
fit_on_texts(texts)
```

Updates internal vocabulary based on a list of texts.

Required before using **`texts_to_sequences`** or **`texts_to_matrix`**.

Arguments:

- `texts`: can be a list of strings, or a generator of strings (for memory-efficiency)

`sequences_to_matrix`

```
sequences_to_matrix(
    sequences,
    mode='binary'
)
```

Converts a list of sequences into a Numpy matrix.

Arguments:

- `sequences`: list of sequences (a sequence is a list of integer word indices).
- `mode`: one of "binary", "count", "tfidf", "freq"

Returns:

A Numpy matrix.

Raises:

- `ValueError`: In case of invalid **`mode`** argument, or if the Tokenizer requires to be fit to sample data.

`texts_to_matrix`

```
texts_to_matrix(
    texts,
    mode='binary'
)
```

Convert a list of texts to a Numpy matrix.

Arguments:

- `texts`: list of strings.
- `mode`: one of "binary", "count", "tfidf", "freq".

Returns:

A Numpy matrix.

`texts_to_sequences`

```
texts_to_sequences(texts)
```

Transforms each text in texts in a sequence of integers.

Only top "num_words" most frequent words will be taken into account. Only words known by the tokenizer will be taken into account.

Arguments:

- `texts`: A list of texts (strings).

Returns:

A list of sequences.

`texts_to_sequences_generator`

```
texts_to_sequences_generator(texts)
```

Transforms each text in texts in a sequence of integers.

Only top "num_words" most frequent words will be taken into account. Only words known by the tokenizer will be taken into account.

Arguments:

- `texts`: A list of texts (strings).

Yields:

Yields individual sequences.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**