-

Install Learn

- Introduction New to TensorFlow?
- TensorFlow The core open source ML library
- For JavaScript TensorFlow.js for ML using JavaScript
- For Mobile & IoT TensorFlow Lite for mobile and embedded devices
- For Production TensorFlow Extended for end-to-end ML components
- Swift for TensorFlow (in beta)

API

- API
- r2.0 (stable)
- r2.1 (rc)
- API r1
- r1.15
- More…

Resources

- Models & datasets Pre-trained models and datasets built by Google and the community
- Tools Ecosystem of tools to help you use TensorFlow
- Libraries & extensions Libraries and extensions built on TensorFlow
- Learn ML Educational resources to learn the fundamentals of ML with TensorFlow

Community Why TensorFlow

- About
- Case studies
- Trusted Partner Program

Language

English

中文 – 简体

Language

- Language
- English
- 中文 – 简体

GitHub
Sign in

- Datasets v1.3.2

Overview Catalog Guide API

-
- Install
- Learn
  - More
- API
  - More
- Resources
  - More
  - Overview
  - Catalog
  - Guide
  - API
- Community
- Why TensorFlow
  - More
- GitHub
- tfds
  - Overview
  - as_numpy
  - builder
  - disable_progress_bar
  - is_dataset_on_gcs
  - list_builders
  - load
  - percent
  - show_examples
  - Split
- tfds.core
  - Overview
  - BeamBasedBuilder
  - BeamMetadataDict
  - BuilderConfig
  - DatasetBuilder
  - DatasetInfo
  - disallow_positional_args

  - Experiment
  - GeneratorBasedBuilder
  - get_tfds_path
  - lazy_imports
  - Metadata
  - MetadataDict
  - NamedSplit
  - ReadInstruction
  - SplitBase
  - SplitDict
  - SplitGenerator
  - SplitInfo
  - Version
- tfds.decode
  - Overview
  - Decoder
  - make_decoder
  - SkipDecoding
- tfds.download
  - Overview
  - add_checksums_dir
  - ComputeStatsMode
  - DownloadConfig
  - DownloadManager
  - ExtractMethod
  - GenerateMode
  - iter_archive
  - Resource
- tfds.features
  - Overview
  - Audio
  - BBox
  - BBoxFeature
  - ClassLabel
  - FeatureConnector
  - FeaturesDict
  - Image
  - Sequence
  - Tensor
  - TensorInfo
  - Text
  - Video
  - text
    - Overview

tfds.features.text.SubwordTextEncoder

View source on GitHub

Class `SubwordTextEncoder`

Invertible `TextEncoder` using word pieces with a byte-level fallback.

Inherits From: `TextEncoder`

Used in the tutorials:

- Transformer model for language understanding

Encoding is fully invertible because all out-of-vocab wordpieces are byte-encoded.

The vocabulary is "trained" on a corpus and all wordpieces are stored in a vocabulary file. To generate a vocabulary from a corpus, use `tfds.features.text.SubwordTextEncoder.build_from_corpus`.

Typical usage:

```
# Build
encoder =
tfds.features.text.SubwordTextEncoder.build_from_corpus(
    corpus_generator, target_vocab_size=2**15)
```

```
encoder.save_to_file(vocab_filename)

# Load
encoder =
tfds.features.text.SubwordTextEncoder.load_from_file(vocab_
filename)
ids = encoder.encode("hello world")
text = encoder.decode([1, 2, 3, 4])
```

`__init__`

View source

```
__init__(vocab_list=None)
```

Constructs a SubwordTextEncoder from a vocabulary list.

Note: To generate a vocabulary from a corpus, use `tfds.features.text.SubwordTextEncoder.build_from_corpus`.
Args:

- **`vocab_list`**: `list<str>`, list of subwords for the vocabulary. Note that an underscore at the end of a subword indicates the end of the word (i.e. a space will be inserted afterwards when decoding). Underscores in the interior of subwords are disallowed and should use the underscore escape sequence.

Properties

`subwords`

`vocab_size`

Size of the vocabulary. Decode produces ints [1, vocab_size).

Methods

`build_from_corpus`

View source

```
@classmethod
build_from_corpus(
    cls,
    corpus_generator,
    target_vocab_size,
    max_subword_length=20,
    max_corpus_chars=None,
    reserved_tokens=None
)
```

Builds a `SubwordTextEncoder` based on the `corpus_generator`.

Args:

- **`corpus_generator`**: generator yielding `str`, from which subwords will be constructed.
- **`target_vocab_size`**: `int`, approximate size of the vocabulary to create.
- **`max_subword_length`**: `int`, maximum length of a subword. Note that memory and compute scale quadratically in the length of the longest token.
- **`max_corpus_chars`**: `int`, the maximum number of characters to consume from `corpus_generator` for the purposes of building the subword vocabulary.
- **`reserved_tokens`**: `list<str>`, list of tokens that will always be treated as whole tokens and not split up. Note that these must contain a mix of alphanumeric and non-alphanumeric characters (e.g. "") and not end in an underscore.

Returns:

`SubwordTextEncoder`.

decode

View source

```
decode(ids)
```

Decodes a list of integers into text.

encode

View source

```
encode(s)
```

Encodes text into a list of integers.

load_from_file

View source

```
@classmethod
load_from_file(
    cls,
    filename_prefix
)
```

Extracts list of subwords from file.

`save_to_file`

View source

```
save_to_file(filename_prefix)
```

Save the vocabulary to a file.

Was this page helpful?

- **Stay connected**
  - Blog
  - GitHub
  - Twitter
  - YouTube
- **Support**
  - Issue tracker
  - Release notes
  - Stack Overflow
  - Brand guidelines
- Terms
- Privacy
- Sign up for the TensorFlow monthly newsletter Subscribe

Language

English

中文 – 简体

Language

- Language
- English
- 中文 – 简体