

Module: `tf.contrib.seq2seq`[Contents](#)[Classes](#)[Functions](#)

Defined in `tensorflow/contrib/seq2seq/__init__.py`.

Ops for building neural network seq2seq decoders and losses.

See the [Seq2seq Library \(contrib\)](#) guide.

## Classes

---

`class AttentionMechanism`

`class AttentionWrapper` : Wraps another `RNNCell` with attention.

`class AttentionWrapperState` : `namedtuple` storing the state of a `AttentionWrapper`.

`class BahdanauAttention` : Implements Bahdanau-style (additive) attention.

`class BahdanauMonotonicAttention` : Monotonic attention mechanism with Bahadanau-style energy function.

`class BasicDecoder` : Basic sampling decoder.

`class BasicDecoderOutput`

`class BeamSearchDecoder` : BeamSearch sampling decoder.

`class BeamSearchDecoderOutput`

`class BeamSearchDecoderState`

`class CustomHelper` : Base abstract class that allows the user to customize sampling.

`class Decoder` : An RNN Decoder abstract interface object.

`class FinalBeamSearchDecoderOutput` : Final outputs returned by the beam search after all decoding is finished.

`class GreedyEmbeddingHelper` : A helper for use during inference.

`class Helper` : Interface for implementing sampling in seq2seq decoders.

`class InferenceHelper` : A helper to use during inference with a custom sampling function.

`class LuongAttention` : Implements Luong-style (multiplicative) attention scoring.

`class LuongMonotonicAttention` : Monotonic attention mechanism with Luong-style energy function.

`class SampleEmbeddingHelper` : A helper for use during inference.

`class ScheduledEmbeddingTrainingHelper` : A training helper that adds scheduled sampling.

`class ScheduledOutputTrainingHelper` : A training helper that adds scheduled sampling directly to outputs.

**class TrainingHelper** : A helper for use during training. Only reads inputs.

## Functions

---

**dynamic\_decode(...)** : Perform dynamic decoding with **decoder** .

**gather\_tree(...)** : Calculates the full beams from the per-step ids and parent beam ids.

**hardmax(...)** : Returns batched one-hot vectors.

**monotonic\_attention(...)** : Compute monotonic attention distribution from choosing probabilities.

**safe\_cumprod(...)** : Computes cumprod of x in logspace using cumsum to avoid underflow.

**sequence\_loss(...)** : Weighted cross-entropy loss for a sequence of logits.

**tile\_batch(...)** : Tile the batch dimension of a (possibly nested structure of) tensor(s) t.

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated November 2, 2017.*

### Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

### Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

**English**

[Terms](#) | [Privacy](#)