

Quick start

🕒 Last updated: 15 February 2022

Recommended software

[GPU version](#)

[CPU version](#)

[Ubuntu packages](#)

Installation

Running Marian

[Training](#)

[Translation](#)

Resources

Recommended software

GPU version

Minimal requirements tested on Ubuntu 18.04 LTS:

- CMake 3.10.2
- GCC/G++ 7.3
- CUDA 9.2

Notes:

- GCC 8+/CUDA 10+ or higher are recommended
- CUDA 10.0+ requires CMake 3.12.2+ due to bugs in earlier versions
- Compilation of the web-server tool requires Boost 1.65.1+

CPU version

A Marian CPU build requires [Intel MKL](#) (recommended) or [OpenBLAS](#). CPU build can be enabled by adding `-DCOMPILER_CPU=on` to the CMake command.

Ubuntu packages

Assuming a fresh Ubuntu LTS installation with CUDA, the following packages need to be installed to compile Marian with minimal dependencies:

- Ubuntu 18.04 (or newer) + CUDA 9.2 (the default is gcc 7.3.0):

```
sudo apt-get install git cmake build-essential
```

In general the standard packages of recent Ubuntu LTS editions should work, but some configurations of C++ compiler and CUDA may be incompatible with each other. Additional [packages](#) can be

installed to compile Marian with the web server, built-in SentencePiece and TCMalloc support.

Installation

Clone a fresh copy from github:

```
git clone https://github.com/arian-nmt/arian
```

The project is a standard CMake out-of-source build, which on Linux can be compiled by executing the following commands:

```
mkdir marian/build
cd marian/build
cmake ..
make -j4
```

If run for the first time, this will also download several submodule repositories.

For details on installation under Windows see [the documentation](#).

Running Marian

Training

Marian is the training framework of Marian. Assuming `corpus.en` and `corpus.ro` are corresponding and preprocessed files of a English-Romanian parallel corpus, the following command will create a Nematus-compatible neural machine translation model.

```
./marian/build/arian \
--train-sets corpus.en corpus.ro \
--vocabs vocab.en vocab.ro \
--model model.npz
```

See the [documentation](#) for more details or the [examples](#) of how to train different models with Marian.

Translation

If a trained model is available, run:

```
echo "This is a test." | ./marian/build/arian-decoder -m model.npz -v vocab.en vocab.ro
```

For translation on CPU, add `--cpu-threads N` (assuming Marian has been compiled with CPU support):

```
echo "This is a test." | ./marian/build/marian-decoder -m model.npz -v vocab.en vocab.ro --cpu-threads 1
```

See the [documentation](#) for more details or the [examples](#) of how to use Edinburgh's WMT models for translation.

Resources

- [User Documentation](#) and [FAQ](#)
- [Google Discussion Group](#)
- Issues on GitHub repositories: [marian](#) and [marian-dev](#)

Marian - an efficient Neural Machine Translation framework written in pure C++.
Mainly developed at Microsoft Translator and at the University of Edinburgh.

[Marian](#) is licensed under the [MIT license](#).

Based on the theme PrettyDocs designed by [Xiaoying Riley](#) with modifications.