CS 410 Technology Review: Topic Discovery with MeTA in a Docker environment

Hao Mou (haomou2)

Fall, 2022

The goal of this review is to demonstrate topic discovery with MeTA[1], a modern C++ data sciences toolkit, in a Docker environment. We will first go through the steps of setting up MeTA in a Docker container. Then, we will follow the guide[2] on the MeTA website to run LDA over an example corpus.

1 Set up MeTA in Docker

MeTA is a C++ toolkit, which means you will need to compile it first before using it. However, the availability, naming, and usage of C++ build tools differ from platform to platform and sometimes from one version to another of the same platform. This complicates the build process and, if we stick with native builds, restricts the applicability of this review.

Docker build circumvents this restriction by "packaging" the environment of a specific version of a platform into an image that can run on all compatible platforms. This gives us a reproducible build and runtime environment where we can build and run MeTA.

1.1 Build the Docker Image

For this review, we will be packaging MeTA into a Docker image based on Ubuntu 14.04 official image[3] (ubuntu:trusty), which could run on MacOS, Linux, and other compatible platforms. The image is described by the following Dockerfile. Note that we have updated the url to the ICU package[4], as the orginal link is no longer valid.

```
FROM ubuntu:trusty
RUN apt-get update && apt-get install -y software-properties-common &&\
    add-apt-repository ppa:george-edison55/cmake-3.x && apt-get update &&\
    apt-get install -y g++ cmake libicu-dev git libjemalloc-dev zlib1g-dev &&\
    rm -rf /var/lib/apt/lists/*
RUN git clone --depth 1 --branch v3.0.2
 ↪  https://github.com/meta-toolkit/meta.git &&\
    cd meta && git submodule update --init --recursive
WORKDIR meta
RUN sed -i 's,http://download.icu-project.org/files/icu4c/58.2/,'\
'https://github.com/unicode-org/icu/releases/download/release-58-2/,'
 ↪  CMakeLists.txt &&\
    mkdir build && cd build && cp ../config.toml . &&\
    cmake ../ -DCMAKE_BUILD_TYPE=Release && make
WORKDIR /meta/build
```

To build the image, we will run the following command inside the terminal of a machine that has Docker installed and running. The Dockerfile is in the working directory. We will tag the image as `meta` in our example.

```
$ docker build --tag meta .
```

1.2 Run the Image as a Container

Once we have the image, we can run it as a container using the following command. We will name the container `tmp` for now.

```
$ docker run --name tmp -it meta
```

This should start a container running the image we have just built and then attach a terminal to the container. We will be greeted by the prompt of the default shell. We can confirm the system is working by running MeTA's unit tests.

```
> ./unit-test --reporter=spec
```

2 Topic Discovery with MeTA

With MeTA set up inside a Docker container, we can now follow the Topic Models Tutorial[2] and find some topics! Specifically, we will run the topic modeling application bundled with MeTA to apply LDA to the corpus to produce a `.phi` file that stores $P(w|z)$ for each (w, z). Then, we will use the bundled `./lda-topics` tool to report the top words in each found topic.

First, let's examine the LDA section in the default `config.toml` in the MeTA project. You can see we are using Gibbs Sampling[5] here. You can also specify `cvb` for Collapsed Variational Bayes[6] or `pargibbs` for Parallel Gibbs Sampling[7]. The maximum number of iterations is set to 1000 so that `lda` will stop once it converges or it has run 1000 iterations, whichever comes first. We set the number of topics to 4.

```
[lda]
inference = "gibbs"
max-iters = 1000
alpha = 1.0
beta = 1.0
topics = 4
model-prefix = "lda-model"
```

The corpus in this case is the `ceeaus` dataset bundled with MeTA. We can run LDA on it by running the following command.

```
> ./lda config.toml
...
Iteration 264 log likelihood (log P(W|Z)): -643071
 Found convergence after 264 iterations!
1667651692: [info]     Finished maximum iterations, or found convergence!
↪  (/meta/src/topics/lda_gibbs.cpp:78)
```

LDA in this case converges in 264 iterations. We can examine the top 5 words in the 4 topics we have found by running the following:

```
> ./lda-topics config.toml lda-model.phi 5
Topic 0:
-----------------------
right (3088): 0.017068
educ (1125): 0.0168412
```

```
believ (331): 0.0168016
busi (459): 0.0152975
financi (1371): 0.0143164

Topic 1:
-----------------------
pass (2646): 0.0072825
demerit (916): 0.00628473
suicid (3589): 0.00420504
pub (2882): 0.00420504
foreign (1420): 0.00395189

Topic 2:
-----------------------
</s> (0): 0.352298
<s> (1): 0.329171
job (1970): 0.213573
part (2637): 0.17553
time (3761): 0.164785

Topic 3:
-----------------------
smoke (3343): 0.543953
</s> (0): 0.364234
<s> (1): 0.340936
restaur (3047): 0.190615
smoker (3348): 0.10802
```

3 Closing Remarks

In this review, we have showcased how to set up MeTA in a Docker container. By using containerized MeTA, we no longer need to worry about the difference in build and runtime environment. We have also demonstrated finding topics using LDA with MeTA. The config based approach is straightforward and friendly to iterative parameter tuning. The ease of use of the config based approach and the pull and use provided by Docker makes MeTA an approachable and powerful tool for topic discovery.

4 Reference

1. MeTA
2. MeTA Topic Models Tutorial
3. Ubuntu official image
4. ICU
5. MeTA Gibbs Sampling
6. MeTA Collapsed Variational Bayes
7. MeTA Parallel Gibbs Sampling