NMT-RTG

Introduction

The page provides details on how to translate documents (via the Tika.translate API) using Reader Translator Generator, a neural machine translation toolkit.

The benefits of using this approach for machine translation through Tika are as follows;

- It's free! As opposed to several other translation services currently available via Tika, NMT via RTG is free.
  - You are not restricted under usage ceiling, and you don't have to allocate monthly payments. There is no paid service behind the scene, you can use this method completely unrestricted.
- You will have full control over the whole pipeline.You may either build NMT models or download pretrained models, set up server and manage backend.
  - Your data and documents are not sent to any services outside of your pipeline. So you can guarantee privacy of your data.

Though, you have to keep these in mind:

- Though you may run the model on CPU for testing, the translation will be very slow on CPUs. GPUs are highly recommended.
- NMT models are not interpretable and explainable. We cannot explain or guarantee that the translations are 100% correct. This is not specific to RTG/NMT; it is generally true for all neural machine translation services.

This is relatively a new addition; the following translation models are currently available:

- Translation from 500 source languages to English

To train models for your desired translation direction, please refer to the documentation at https://isi-nlp.github.io/rtg/#_usage

Integration: Overview

The class `org.apache.tika.language.translate.RTGTranslator` glues Tika system with RTG REST API.
By default, it interacts with http://localhost:6060.
This URL can be customized by adding `translator.rtg.properties` file to classpath with `rtg.base.url` property.

500 Languages to English Translation

Step 1: Start RTG Translator Service

500-English model can be obtained from a docker image as follows

Docker image can be run on CPU (i.e. without GPU, for testing):

```
   docker run --rm -i -p 6060:6060 tgowda/rtg-model:500toEng-v1
```

Using GPU (e.g. Device 0) is recommended for translating a lot of documents:

```
 docker run --rm -i -p 6060:6060 --gpus '"device=0"' tgowda/rtg-model:500toEng-v1
```

Verify that the translator serive is actually running by accessing http://localhost:6060/

Step 2: Start Tika Server Jar

Option 1: Obtain prebuilt jar
Note: This option is for the future versions. The current prebuilt jars do not have this feature integrated. Go to Option 2.

```
wget https://www.apache.org/dyn/closer.cgi/tika/tika-server-2.0.0.jar
```

Option 2: Build Tika Server from source

```
$ git clone https://github.com/apache/tika.git
$ cd tika

# if the pull request is not merged yet; please pull from this repo
$ git checkout -b TIKA-3329
$ git pull https://github.com/thammegowda/tika.git TIKA-3329

# Compile and package Tika

$ mvn clean package -DskipTests
```

```
# Start Tika server
$ java -jar tika-server/target/tika-server-2.0.0-SNAPSHOT.jar
```

Step 3:Translate Documents via Tika + RTG

```
printf "Hola señor\n\nBonjour monsieur\n\n" > tmp.txt
$ curl http://localhost:9998/translate/all/org.apache.tika.language.translate.RTGTranslator/x/eng -X PUT -T
tmp.txt

Hi, sir.
Namaskar
Good morning, sir.
Hi.
```

Optional: Change the base URL of RTG translator service

You may deploy RTG service elsewhere (on a machine with GPU) and point its URL to tika.

Step 1: Create a file named *`translator.rtg.properties`* with *`rtg.base.url`* property

```
    echo "rtg.base.url=http://<myhost>:<port>/rtg/v1" > translator.rtg.properties
```

Step 2: Add the directory having `translator.rtg.properties` to classpath; In this case . i.e, $PWD

```
   java -cp '.:tika-server/target/tika-server-2.0.0-SNAPSHOT.jar' org.apache.tika.server.TikaServerCli
```

Step 3: Interact with Tika Server as usual

```
 $ curl http://localhost:9998/translate/all/org.apache.tika.language.translate.RTGTranslator/x/eng -X PUT -T
tmp.txt
```

Acknowledgements

If you wish to acknowledge or reference either RTG toolkit or the 500-English model, please reference this article: https://arxiv.org/abs/2104.00290

```
@misc{gowda2021manytoenglish,
   title={Many-to-English Machine Translation Tools, Data, and Pretrained Models},
   author={Thamme Gowda and Zhao Zhang and Chris A Mattmann and Jonathan May},
   year={2021},
   eprint={2104.00290},
   archivePrefix={arXiv},
   primaryClass={cs.CL}
}
```