CISC7021 - Applied Natural Language Processing - Assignment 2

Zhang Huakang
M-C3-5095-0

1 Environment Setup

In this project, four software packages are used: **Boost**, which is used for providing C++ libraries that are useful for tasks such as linear algebra, multithreading; **cmph**, which is a C library for minimal perfect hashing; **mosesdecoder**, which is a statistical machine translation system, and **mgiza**, which is a multi-threaded implementation of the word alignment tool GIZA++.

The installation is done on a docker container with Ubuntu 14.04. This *Dockerfile* is published on Github.

Since the unfamiliarity of the software packages, I spent a lot of time on the installation.

Because the tokenization of Chinese is different from English, I use *ansjTokenizer* to tokenize Chinese, which is implemented by Java. *openjdk:8-jdk* image is used to run the Java program.

2 Tokenize

Listing 1: Chinese Tokenization

```
openjdk:8-jdk \
java -jar /mnt/ansjTokenizer.jar \
/mnt/train.tags.zh-en.zh \
/mnt/train.token.zh
```

Listing 2: English Tokenization

```
$MOSES_TOKEN/tokenizer.perl\
-l en \
-threads 4 \
< /mnt/train.tags.zh-en.en \
> /mnt/train.token.en
```

Listing 3: Reduce Parallel Corpus

```
$MOSES_TRAINING$/clean-corpus-n.perl\
/mnt/train.token \
en \
zh \
/mnt/train.token.clean.50 \
1 \
50 \
-lowercase 1
```

By the code in *Listing 3*, we limit the length of the parallel corpus text to no more than fifty words, and convert the English to all lowercase.

3 3-gram Language Model

LMPLZ is used to train the language model, which is estimates language models with Modified Kneser-Ney smoothing and no pruning. The command is shown in *Listing 4*.

```
$MOSE_BIN/lmplz \
-o 3 \
-S 50% \
-T /mnt/tmp \
--text /mnt/train.token.clean.50.zh \
--arpa /mnt/train.token.clean.50.lm.zh \
--discount_fallback

$MOSE_BIN/lmplz \
-o 3 \
-S 50% \
-T /mnt/tmp \
--text /mnt/train.token.clean.50.en \
--arpa /mnt/train.token.clean.50.lm.en \
--discount_fallback
```