Machine Translation
Homework 03 - Preprocessing

Aqeel Labash

November 16, 2016

1 Introduction

For this home work I used `Opensubtitle` to translate from `Arabic` to `English`. The original file contained $16000,000 \sim$ sentence. I used `Trainin: 1000000`, `dev:5000` and `test:2000`.

2 General Code

```
MOSES=/home/aqeel/MT/moses
echo Preparing Data
tst=2000
trn=1000000
dev=5000
all=0
let "all=tst+trn+dev"

echo Take $trn for training , $dev for dev , $tst for testing
sed -n 1,"$all"p OpenSubtitles2016.ar-en.ar > all.ar
sed -n 1,"$all"p OpenSubtitles2016.ar-en.en > all.en
echo Split the data
lstart=0
lend=0
let "stlimit+=1"
let "lend=trn"
sed -n "$stlimit","$lend"p all.ar > train.ar
sed -n "$stlimit","$lend"p all.en > train.en
let "stlimit+=trn"
let "lend=stlimit+dev"
sed -n "$stlimit","$lend"p all.ar > dev.ar
sed -n "$stlimit","$lend"p all.en > dev.en

let "stlimit+=dev"
let "lend = stlimit+tst"
sed -n "$stlimit","$lend"p  all.ar > test.ar
sed -n "$stlimit","$lend"p all.en > test.en
rm all.ar all.en
echo Tokinizing Data
for f in {train,dev,test}.{ar,en};
do
  $MOSES/scripts/tokenizer/tokenizer.perl -threads 8 < $f > tok-$f
  rm $f
done
echo Truecasing English
$MOSES/scripts/recaser/train-truecaser.perl --model en-truecase.mdl --corpus tok-train.en
for f in {test,dev,train}.en;
do
  $MOSES/scripts/recaser/truecase.perl --model en-truecase.mdl < tok-$f > tc-tok-$f
  rm tok-$f
done
for f in {train,dev,test};
do
  mv tok-$f.ar tc-tok-$f.ar
  rm tok-$f.ar
done

$MOSES/+scripts/training/clean-corpus-n.perl tokenized-and-lowecased ar en cleaned 1 100

$MOSES/bin/lmplz -o 5 -S 50% < tc-tok-train.en > lm-en.arpa
```

```
$MOSES/bin/build_binary lm-en.arpa lm-en.blm

$MOSES/scripts/training/train-model.perl --corpus tc-tok-train --f ar --e en --external-bin-dir $MOSES/bin --lm 0:5:$(pwd)/lm-en.blm --root-dir mt-experiment-1 --reordering msd-bidirectional-fe --mgiza --mgiza-cpus 8
# To Enhance Peroframce Later (on Testing level).
cd mt-experiment-1
mkdir binarised-model
$MOSES/bin/processPhraseTableMin -in model/phrase-table.gz -nscores 4 -out binarised-model/phrase-table
$MOSES/bin/processLexicalTableMin -in model/reordering-table.wbe-msd-bidirectional-fe.gz -out binarised-model/reordering-table
cd ..


for i in `seq 1 3`;
do
  echo ======================$i======================
# $MOSES/scripts/training/mert-moses.pl $(pwd)/tc-tok-dev.ar $(pwd)/tc-tok-dev.en $MOSES/bin/moses train/model/moses.ini --mertdir $MOSES/bin/ --decoder-flags="-threads all" &> mert$i.out
  $MOSES/scripts/training/mert-moses.pl $(pwd)/tc-tok-dev.ar $(pwd)/tc-tok-dev.en $MOSES/bin/moses $(pwd)/mt-experiment-1/model/moses.ini --working-dir $(pwd)/mt-experiment-1/mert-$i --threads 4 --decoder-flags "--threads 4" > mert-$i.out
done
```

Since I used `cmph` to decrease the amount of memory required later, I had to split the bash file to update the `Path` value in `moses.ini` in `mert` files.[1]
The second part is to calculate the `BLEU` score.

```
for i in `seq 1 3`;
do
  $MOSES/bin/moses -f mt-experiment-1/mert-$i/moses.ini -i tc-tok-test.ar > mt-experiment-1/mert-$i/hypothesis0.ar
  $MOSES/scripts/generic/multi-bleu.perl tc-tok-test.en < mt-experiment-1/mert-$i/hypothesis0.ar > out_$i.txt
done
```

3 The Random Sentences

For random sentences I just ran a the following code :

```
import random
for i in range(10):
  print random.randint(0,2001)
```

Then I enhanced the chosen sentences (to decrease the number of small sentences). in Figure 1 we can see the sentences[2].

[1] Thanks To Maksym, didn't know about it before.
[2] I used images to put arabic text due to compatability issue between arabic and latex and my needs.

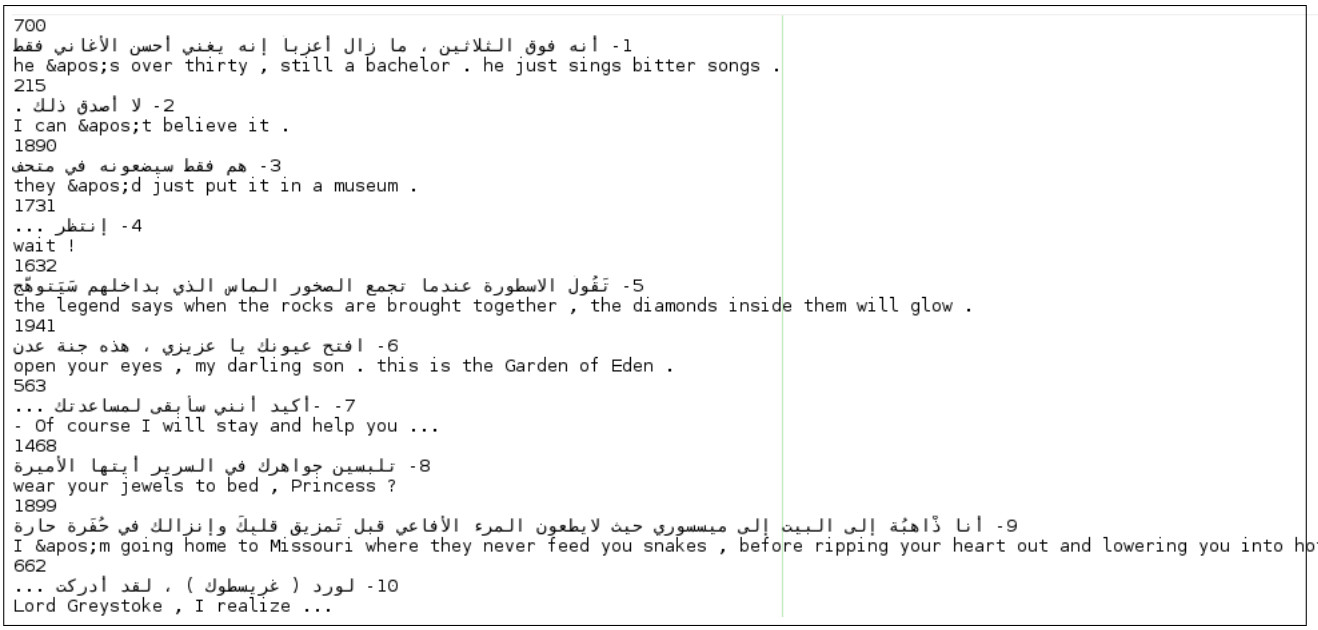

Fig. 1: the sentence number in random file, followed by Arabic sentence (right to left) starting with sentence number (1 to 10) followed by the english sentence.(left to right)

4 Notes on the original translation

- I would say the translation it self is not accurate in sense of using different expresion (using best instead of better) and in sense of the using mix of accents with standard accent.
- The translation it self is not the best form that I would use to translate those seneteces.
- diarectecs is also included in some cases (removing them lead to lose of meaning in some cases).

5 Baseline

For the baseline (used the previous bash code exactly) I got the following score[3] score:
`BLEU = 24.23, 57.5/31.8/19.5/12.1 (BP=0.945, ratio=0.946, hyp_len=13502, ref_len=14266)`
Figure 2 show the baseline translation.

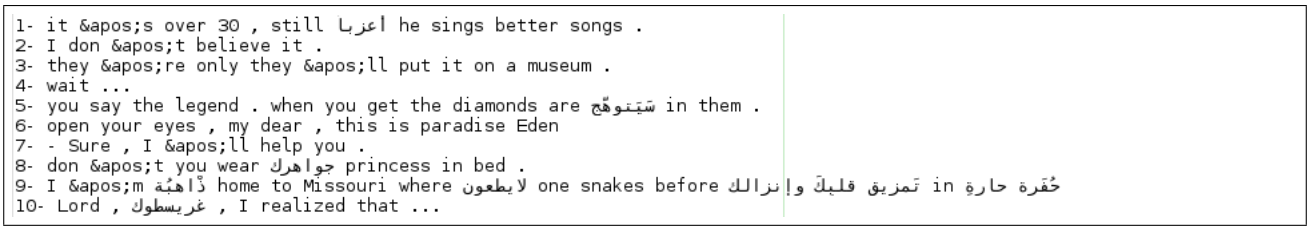

Fig. 2: baseline translation with id

6 Compund Split

7 Byte Pair Encoding

I used this way (took around one day per one tuning process and couldn't afford more time to invest

[3]The best score between all the 3 tuning operations