LING 570: HW4 – morphological segmentation (2/2) (100 pts)

Yao Yan (1669598)

2.1. generate translation.lexicon.proba

Find pattern german_word ({number}) from en-de.A3.final.gz and record the counts of every corresponding English translation normalized by the total counts. Set the threshold as 0.01, only incoporate English translation that have a frequency over 0.01. **Also, I didn't incorporate numbers and punctuation in the traslation_lexicon table.**

2.2 split each german word, using:

command for running 2b is :

zcat en-de.A3.final.gz | ./translation_lexicon.sh > translation.lexicon.proba

sort translation.lexicon.proba | head > translation.lexicon.proba.head

cat file.txt | ./translation_lexicon_german.sh > file.segmented_translation_lexicon

1. keep the entire word, if the word has English translation, print this
2. split the word to two parts, if both of the two splits have English translation, print this.
3. split the word to three parts, exlude connector words[s, en, es, n], if each split has English translation, print this split
4. split the word to three parts, only consider 4-split when connector words[s, en, es, n] exists, exclude the connector word, if the left three splits have English translation, print this.

Note: if a German word has multiple English translation, use the most frequent English translation from 2.1 I generated a for_2b file which has the most frequent English translation with each German word: inside is like. I had a separate for_3b file for 3b.

```
er the
sitzungsperiode the session
ich i
erkläre declare
die the
am on
freitag friday
dezember december
unterbrochene resumed
des of the
```

```
europäischen european
parlaments parliament
wiederaufgenommen resumed
wünsche wish
ihnen you
nochmals once again
alles everything
gute good
jahreswechsel end ... year
und and
```

My .sh file for 2b (translation_lexicon_german.sh) is

```
#!/bin/sh
python3 translation_lexicon_german.py for_2b$@
```

My .sh file for 3b (second_translation _german.sh)is

```
#!/bin/sh
python3 translation_lexicon_german.py for_3b$@


~
```

3.1 convert europarl-v7.de-en.lower-de into europarl-v7.de-en.lower-segmented-de. This used the frequency metrics from hw3.

3.2 . I had a separate for_3b file for segmenting file.txt for 3b.

command for running 3b is :

en-de.A3.final2.gz is generated by running GIZA++ and mosesdecoder

```
zcat en-de.A3.final2.gz | ./translation_lexicon.sh > second.translation.lexicon.proba

sort  second.translation.lexicon.proba | head > second.translation.lexicon.proba.head

cat file.txt | ./second_translation_german.sh > file.segmented_second_t1
```