LIN570: HW3 – morphological segmentation (1/2) (100pts)

Yao Yan (1669598)

Due date: 11pm on Oct 22, 2019 (Tuesday)

1. Reproduce morphological segmentation inside-out results (**?**)[1]

   - • **converting to hierarchy structure:**
     1. use regex to find patterns like "(pomp:stem)";
     2. change the pattern to new patterns like (WORD (SIGMA p) (SIGMA o) (SIGMA m) (SIGMA p)) by converting S, stem to WORD and disconnecting letter in 'pomp'
     3. similar ways to convert the files provided to flat structure, but in flat structure, there aren't any hierarchical differences among segments of a word
     4. `run: cat train0 | ./convert_hierachy_penn.sh > train0.hierarchy.penn`
        `run: cat train0 | ./convert_flat_penn.sh > train0.flat.penn`
     5. apply the same command to test0 and dev0
   - **train using Berkeley Parser:**
     1. `cat train0.flat.penn dev0.flat.penn > flat.penn cat train0.hierarchy.penn dev0.hierarchy.penn > hierarchy.penn`
     2. `train the model by using berkeley parser: java -cp berkeleyParser.jar edu.berkeley.nlp.PCFGLA.GrammarTrainer -path hierarchy.penn -out hierarchy.model`
     3. `convert test0.hierarchy.penn to test0.leaf cat test0.hierarchy.penn | ./convert_leaf.sh > test0.leaf`
     4. `parse test0.leaf using hierachy.model and flat.model cat test0.leaf | java -jar berkeleyParser-1.7.jar -gr hierarchy.model > test0.hierarchy.parsed`
     5. `evaluate by using EVALB: ./evalb test0.hierarchy.penn test0.hierarchy.parsed`

2. Reproduce a frequency based metric using the monolingual German corpus in described in :

   - • **splitting words:**

     $$\underset{S}{\operatorname{argmax}}\,(\prod_{p_i \in S} C(p_i))^{\frac{1}{n}} \tag{1}$$

     1. `calculate the frequency of each segment by counting its appearances in europarl-v7.de-en.true.de`
     2. `keep the original word, e.g.aktionsplan`

[1] 10-fold data split available at `https://ryancotterell.github.io/data/splits.tar.gz`

3. split the original word aktionsplan in half, e.g.aktions plan
4. split the original word aktionsplan into three parts, e.g.aktio ns plan. Notice if the middle part is a connector(-n-, -en-, -s-, -es-), the frequency of the connector word is not counted, the calculation will be the same as the one used in 3.
5. split the original word aktionsplan into for parts,only consider the situation where the second or the third segment is a connector word
6. only print out non-zero frequency splits in a frequency descended order
7. cat file.txt | ./frequency$_m$etric$_g$erman.sh > file.score