Natural Language Processing
Lab 7: Lemmatization

1 Introduction

Full morphological analysis and lemmatization normally presupposes a comprehensive lexicon. In this assignment, we will investigate a lexicon-free method for lemmatization of English based on rule-based stemming techniques.

2 Data

We are going to make use of the same data from the English Web Treebank that we used in the tagging assignment. However, since we are using an approach based on hand-crafted rules, we just need a development set and not a separate training set. As input to the lemmatizer, we will use the file `ewt-dev-wt.txt`, which contains both words and part-of-speech tags, and the task is to predict the base form of the word. The gold standard is provided in the file `ewt-dev-wtl.txt`, which contains words, tags and lemmas. All the files needed can be found in `/local/kurs/nlp/tagging/`.

3 Starter code

The file `lemmatizer.py` contains skeleton code for a rule-based lemmatizer:

```python
import sys

def noun_lemma(word):
    if word.endswith("s"):
        return word[:-1]
    else:
        return word

def verb_lemma(word):
    if word.endswith("ed"):
        return word[:-2]
    else:
        return word

def adj_lemma(word):
    if word.endswith("er"):
        return word[:-2]
    elif word.endswith("est"):
        return word[:-3]
    else:
        return word

for line in sys.stdin:
    if line.strip():
        (word, tag) = line.strip().split("\t")
        lemma = word
        if tag == "NOUN":
            lemma = noun_lemma(word)
        elif tag == "VERB":
            lemma = verb_lemma(word)
        elif tag == "ADJ":
            lemma = adj_lemma(word)
        else:
            lemma = word
        print("{0}\t{1}\t{2}".format(word, tag, lemma))
    else:
        print()
```

The main program reads the input file line by line and adds a lemma after the word and tag. It uses three simple methods for lemmatization of nouns, verbs and adjectives and otherwise treats the token as its own lemma. The program is run as follows:

```
python lemmatizer.py < ewt-dev-wt.txt > ewt-dev-out.txt
```

To evaluate the quality of the lemmatization, you can use `score.py` with the flag `lemma`:

```
python score.py lemma ewt-dev-wtl.txt ewt-dev-out.txt
```

4 Improve lemmatization

The simple baseline lemmatizer has an accuracy of 80%. Your task is to improve the accuracy as much as possible by modifying the code. You may expand existing methods, add new methods, or modify the main program, as you see fit. To get inspiration, you can consult the sections on stemming in the textbook. To find out which words your lemmatizer still gets wrong, you may use the `diff` command on the command line:

```
diff ewt-dev-wtl.txt ewt-dev-out.txt | less
```

or use vimdiff (see Lab 2):

```
vimdiff ewt-dev-wtl.txt ewt-dev-out.txt
```

The goal is to reach at least 85% accuracy on the development set.