Middle East Technical University   Department of Computer Engineering

CENG 463

Introduction to Natural Language Processing

Fall '2012-2013

Programming Assignment 2

Due date: 25 November 2012, Sunday, 23:55

1 Objectives

In this assignment, you are expected to implement a part of speech tagger for Turkish. You will be asked to tag a given sentence in Turkish.
You are also expected to write a report on your implementation details.

Keywords: *Part of Speech Tagging, ENGTWOL Tagger, Forward-Backward Algorithm, Baum-Welch Algorithm, Viterbi Algorithm, Brill Tagger*

2 Part of Speech

A part of speech of a word is the lexical category of the word (also word class, morphological class, lexical class, or lexical tag). A part of speech of a word provides information about a word and its neighbours, word's pronunciation, its morphological analysis or can be used in speech recognition, parsing, translation, information retrieval and question answering tasks. Many NLP applications may benefit from syntactically disambiguated text.

3 Part of Speech Tagging

Part of speech tagging is the process of assigning tags (class markers) to words. A word can be a member of multiple classes and main task of a tagger is to solve this ambiguity. Successful approaches can sole over 95% of these ambiguities. There are many machine learning algorithms applied to tagging problem and can achieve over 95% accuracy. However even a 1% accuracy matters. For example, in the case of a 15-word sentence where a tagger with 98% overall accuracy has 74% chance of tagging all the words, a tagger with 97% overall accuracy has only 63% chance. Your performance depends considerably on the amount of training data, your tag set, your dictionary and unknown words.

3.1 Rule-Based Part of Speech Tagging

Earliest algorithms for tagging were rule-based and had two phases. First, all possible tags for a word are assigned to the word and then disambiguation rules are applied to remove tags until a single part of speech is left.

3.2 Stochastic Part of Speech Tagging

Hidden Markov Models are used to tag sequence of words. There are several algorithms for learning and decoding for HMMs. Baum-Welch and Viterbi Algorithms are commonly used for these problems. HMMs model word sequences (sentences) as emissions of tags which depend only on the previous ones. These are called Markov chains and have two properties: Limited horizon and time invariance.

3.3 Transformation-Based Part of Speech Tagging

Described by Eric Brill, transformation-based taggers use rules to specify which tags are possible for words and supervised learning to examine possible transformations, improvements and re-tagging. Transformation-based tagging can use lexical and syntactic regularities, words and context compared to HMMs that only depends on previous tags.

4 Specifications

1. **N**ame your module pos_tagger
2. **Y**ou will implement a function that will have single sentence and return (word,tag) pairs of the sentence.

   ```
   tag(sentence)
   ```

3. **Y**ou are expected to train your system with provided development file which has 5110 sentences. Development file will be available on the test system
4. **Y**our module will be tested with an additional test.py file which imports your module and calls implemented functions.

   ```
    from pos_tagger import tag
   ```

5 Training

You are given a development file containing sentences of tagged words. Each line is a word tag pair in the form of 'word'|'tag'. Sentences are separated with empty lines. Your development file contains 5110 sentences. You are expected to divide your development set into two parts namely training set and dev-test set. You will train your Part of Speech Tagger with training set and analyze your performance on your dev-test set to make improvements in your tagger. You are recommended to use 85-90% of sentences for training and remaining for dev-test. You can use several techniques such as k-fold cross validation to perform analysis of your tagger.

6 Samples

In development file:

```
Ve|Conj
hüzün|Noun_Nom
yüzündeki|Adj_Noun
kırışıklıkları|Noun_Acc
biraz|Adj
daha|Adv
oyarak|Adv_Verb
derinleştiriyor|Verb_Adj
.|Punc
```

Test system:

```
>>> #Tokens will be separated with spaces.
... tag('Ve hüzün yüzündeki kırışıklıkları biraz daha oyarak derinleştiriyor .')
[('Ve','Conj'),('hüzün','Noun_Nom'),('yüzündeki','Adj_Noun'),('kırışıklıkları','↩
    Noun_Acc'),('biraz','Adj'),('daha','Adv'),('oyarak','Adv_Verb'),('derinleş↩
    tiriyor','Verb_Adj'),('.','Punc')]
```

7 Regulations

1. **Programming Language:** You will use Natural Language Toolkit with Python language.
2. **Late Submission:** Penalty for each day is calculated by (number of days) *10.
3. **Cheating:** We have zero tolerance policy for cheating. In case of cheating, all parts involved (source(s) and receiver(s)) get zero. People involved in cheating will be punished according to the university regulations.
4. **R**emember that students of this course are bounded to code of honor and its violation is subject to severe punishment.
5. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.
6. **Evaluation:** Your Part of Speech Taggers will be evaluated on a test set of 500 sentences which contains different sentences from your development set. Your program will be evaluated automatically using 'black-box' technique so make sure to obey the specifications. Your tagger will be tested with an additional test.py file which imports your module and calls your 'tag(sentence)' function.

```
>>> from pos_tagger import tag
>>> tag('Bunu başından beri biliyordum zaten .')
[('Bunu','Pron'),('başından','Noun_Abl'),('beri','Postp'),('biliyordum','↩
    Verb'),('zaten','Adv'),('.','Punc')]
```

$$your\ grade = \frac{number\ of\ correctly\ tagged\ tokens}{number\ of\ all\ tokens} \tag{1}$$

8 Submission

- Submission will be done via COW.
  Create a tar.gz file named `hw2.tar.gz` that contains all your source code files and a copy of your report in pdf format.

9 References

- Speech and Language Processing, Daniel Jurafsky and James H. Martin
- Foundations of Statistical Natural Language Processing, Christopher D. Manning and Hinrich Schütze