Ceng 563
Homework #3 + Take Home

JPost-it – Technical Report

Tugcem Oral, e1705805

09/06/2014

1. Introduction

JPost-it (Java Part-of-Speech Tagging in Turkish) is an automated part-of-speech tagger for Turkish languge. It uses Viterbi algorithm with HMM (Hidden Markov Model) to find appropriate tag for each word for given sentence (separated by white space). To tag a human readable sentence, one must present training corpus set and first train HMM states.

JPost-it has mainly three different types of parser for given input sequence, as requested in Assignment #3.

2. Usage

To use JPost-it, you can run it via:

```
>> java -jar jpost-it.jar (-t /path/to/training.txt [optional]) -pt (postagger1|
postagger2|postagger3) -o (convert|tag) "sentence"
```

command. Notice that you have JRE 1.7.x on your machine installed. The input parameters:

-t training/file/path (optional). You can set training file path used for tagging. Default training file path is "./training.txt". So you need to put *training.txt* file to the same folder with *jpost-it.jar*

-pt postagger1 | postagger2 | postagger3 (required). Represents the model for parsing the input.

-o convert | tag "sentence" (required). Operation. Could convert given input sequence to part-of-speech tags or tags given human readable sentence. "sentence" parameter must be quotated.

3. Evaluation, Scoring & Comments

To evaluate the success of JPost-it, various tests are executed. Since there exists two input files (training.txt and development.txt), 9 tests are executed for each part-of-speech tagger. The execution results and success percents are given below. Here, "merged" file is generated by merging training and development corpus. tr.edu.metu.ceng.postit.main.JPostitScorer deals with given tests.

Scores for PosTagger1:

Training Corpus / Development Corpus	**True Tagged**	**# of sentences tagged**
Merged / Merged	94.772% (2,012 / 2,123)	162 sentences
Merged / Train	94.797% (1,895 / 1,999)	150 sentences
Merged / Dev	94.355% (117 / 124)	12 sentences
Train / Merged	93.453% (1,984 / 2,123)	162 sentences
Train / Train	94.547% (1,890 / 1,999)	150 sentences
Train / Dev	**75.806% (94 / 124)**	**12 sentences**
Dev / Merged	64.107% (1,361 / 2,123)	162 sentences

Dev / Train	61.931% (1,238 / 1,999)	150 sentences
Dev / Dev	99.194% (123 / 124)	12 sentences

Scores for PosTagger2:

Training Corpus / Development Corpus	**True Tagged**	**# of sentences tagged**
Merged / Merged	94.301% (2,002 / 2,123)	162 sentences
Merged / Train	94.247% (1,884 / 1,999)	150 sentences
Merged / Dev	95.161% (118 / 124)	12 sentences
Train / Merged	92.087% (1,955 / 2,123)	162 sentences
Train / Train	94.047% (1,880 / 1,999)	150 sentences
Train / Dev	**60.484% (75 / 124)**	**12 sentences**
Dev / Merged	48.281% (1,025 / 2,123)	162 sentences
Dev / Train	45.123% (902 / 1,999)	150 sentences
Dev / Dev	99.194% (123 / 124)	12 sentences

Scores for PosTagger3:

Training Corpus / Development Corpus	**True Tagged**	**# of sentences tagged**
Merged / Merged	94.724% (2,011 / 2,123)	162 sentences
Merged / Train	94.697% (1,893 / 1,999)	150 sentences
Merged / Dev	95.161% (118 / 124)	12 sentences
Train / Merged	92.181% (1,957 / 2,123)	162 sentences
Train / Train	94.497% (1,889 / 1,999)	150 sentences
Train / Dev	**54.839% (68 / 124)**	**12 sentences**
Dev / Merged	45.125% (958 / 2,123)	162 sentences
Dev / Train	41.721% (834 / 1,999)	150 sentences
Dev / Dev	100% (124 / 124)	12 sentences

For all these tests, first HMM is initialized and trained via training corpus, then all sentences in development corpus are tagged. As their initial tags are known, each generated tag is compared with existing tag and percents are found.

There are several hints for JPost-it; first of all a constant probability is returned if next word to tag is not found in vocabulary. This probability could be improved by defining several rules such tagging a “noun” after an “adjective”. For all PosTaggers, we can see from first three results that if our vocabulary covers all words found in testing corpus, taggers seem to be successful with around 94%. Bolded lines are expected percents for training with *training.txt* and evaluating with *development.txt*.

Also, used tags for all three taggers are given as below. Notice that <S> represents “start” tag and <E> represents “ending” tag used in HMM state transitions.

Tagger Name	**# of Tags Used**	**Tag Set**
PosTagger1	14	Noun, Postp, Num, Pron, Adverb, Det, Interj, Verb, ?, Conj, Ques, Adj, <S>, <E>
PosTagger2	22	Postp, Num, Det, Noun+Gen, Interj, Noun+Acc, Verb, Noun+Nom, Noun+Abl, Ques, Noun+Loc, Noun+Ins, Noun+Equ, Noun+Dat, Pron, Adverb, ?, Noun+Grn, Conj, Adj, <S>, <E>
PosTagger3	62	Verb+Noun+Nom, Noun+Verb+A3sg, Interj, Num+Noun+Nom, Verb+Noun+Abl, Adj+Noun+Abl, Noun+Ins, Noun+Adj+With, Verb+Noun+Acc, Pron, Verb+Noun+Loc, Adj+Noun+Loc, Verb+Adj+Noun+Dat, Conj, Verb+Noun+Gen, Adj, Adj+Adverb, Num, Adj+Noun+Acc, Noun+Verb+Cop, Noun+Acc, Verb+Adj+Noun+Gen, Noun+Verb+A1sg, Verb+Noun+A3sg, Noun+Adverb+Since, Adj+Noun+Ins, Num+Noun+Adj+With, Verb+Noun+Dat, Verb+Adj+Noun+Nom, ?, Adj+Noun+Dat, Verb+Adverb, Num+Noun+Equ, Pron+Verb, Noun+Gen, Det, Verb, Noun+Verb+Adj+PresPart, Noun+Abl, Ques, Noun+Adj+Verb+A3pl, Noun+Adj+Agt, Noun+Dat, Noun+Adj+Without, Noun+Adj+Nom, Adverb, Verb+Adj, Noun+Adj+Verb+Cop, Noun+Verb+A3pl, Postp, Verb+Noun+Ins, Adj+Noun+Nom, Verb+Noun+Cop, Noun+Nom, Adj+Verb, Noun+Loc, Noun+Adj+Rel, Noun+Adj+Related, Noun+Grn, Postp+Verb, <S>, <E>