Assignment–3

Course Name: Natural Language Processing (COMP 8780)

Course Instructor: Professor Vasile Rus

Submitted By

Student Name: Anjana Tiha
UID: U00619942
Date: 03/01/2018

Problem 1

Build a baseline statistical tagger.

Problem (i) [10 points]

Use the assignment#2's hash of hashes to train a baseline lexicalized statistical tagger on the entire BROWN corpus.

(ii) [20 points]

Use the baseline lexicalized statistical tagger to tag all the words in the SnapshotBROWN.pos.all.txt file. Evaluate and report the performance of this baseline tagger on the Snapshot file.

(iii) [20 points]

add few rules to handle unknown words for the tagger in (ii). The rules can be morphological, contextual, or of other nature. Use 25 new sentences to evaluate this tagger (the (ii) tagger + unknown word rules). You can pick 25 sentences from a news article from the web and report the performance on those.

NOTE: You should only consider the 45 proper tags from Penn Treebank tagset (available in the slides). You should disregard tags such as -NONE-, etc.

Answer 1:

Functionality :

Python script Build a baseline statistical tagger using the assignment#2's hash of hashes to train a baseline lexicalized statistical tagger on the entire BROWN corpus. Uses the baseline lexicalized statistical tagger to tag all the words in the SnapshotBROWN.pos.all.txt file. Evaluates and reports the performance of this baseline tagger on the Snapshot file.
Adds few rules to handle unknown words for the tagger in (ii). Uses 26 new sentences from the web to evaluate this tagger performance(the (ii) tagger + unknown word rules). For tagger training, the most frequent tag for a word is used.

Method :

1. Read Complete brown corpus file line by line.
2. Removes special character and everything except word and tag/pos.
3. Writes each tag and word pair space separated in a clean file "BROWN-clean.pos.txt" where original line is maintained.
4. After writing complete file, reads text from "BROWN-clean.pos.txt" line.

5. The text is then processed tag-word pair manner and stored in a hashmap where first level key is word and first level value is another hashmap where key is pos/tag and value is frequency of the tag/POS for word in the whole file.

6. A new hash map is created where key is the original word and value is a tag/pos which is most common for that particular word.

7. The The "SnapshotBROWN.pos.all.txt" file is read line by line.

8. The snapshot file is cleaned similar to main corpus file and saved in a clean file in tag-word manner.

9. For all tag-word pair, their validity is calculated.

10. Performance is evaluated in terms of accuracy, error and unknown words in percentage.

11. New text is collected from web and saved in "article.txt"

12. Text is cleaned and preprocessed to remove extra empty lines.

13. Some rules for handling unknown words is added to the tagger.

14. Using the new adjusted tagger with unknown word handling unit is tested on "article.txt"

15. Performance of new tagger is calculated with couple of measures.

Performance Report :

Percentile of known words tagged = Known Tagged words/Total words in article

Percentile of Unknown words tagged = Unknown Tagged words/Total unknown words in article

Percentile of Unknown words not tagged = Unknown untagged words/Total unknown words in article

Table 1: Performance for file - "SnapshotBROWN.pos.all.txt"

Accuracy Percentile	91.29%
Error Percentile	8.71%
Unspecified word in tagset(percentile)	0.0%

Table 2: Performance for file - "BROWN.pos.all.txt"

Total Number of Words	:	844	
Tagged Words Known (percentile among all words)	:	716	(84.83%)
New Words(percentile among all words)(percentile)	:	128	(15.17%)
Words Tagged(percentile among all new words)	:	113	(88.28%)
Words Could Not Tag (percentile in new words)	:	15	(11.72%)

Inputs :

Script Name: "Anjana_Tiha_NLP_Assignment_3.py"
or for notebook "Anjana_Tiha_NLP_Assignment_3.ipynb"

Input : In command line please type: python *Anjana_Tiha_NLP_Assignment_3.py*

or run the *"Anjana_Tiha_NLP_Assignment_3.py.ipynb"* script
in jupyter notebook of Anaconda.

Note: For running ipynb file, please install "Anaconda". For python please enter absolute location of python.

Output: Output is printed in jupyter notebook or windows terminal.