

CIS 530 Spring 2008

Assignment 3

Lucas Champollion (based on a problem set by Mitch Marcus)

Due: Thursday, March 18, 2008, 4:30pm

Overview of this assignment

In statistical natural language processing, one can use powerful machine learning algorithms to achieve reasonably good results. In order to improve performance, one should take a look at what errors the algorithm is making. Error analysis serves as a starting point to deciding what to do next, whether that means using more features, modifying algorithm parameters, or even switching to a different learning algorithm because one has decided that the current model simply isn't adequate.

In this assignment, you will perform an error analysis on three versions of Brill's transformation-based part-of-speech tagger on text from both the Brown and the Wall Street Journal (WSJ) corpora. Both corpora are available in annotated format in the Penn Treebank. Some programming will be necessary, but the main part of the assignment will be your analysis, and you should accordingly concentrate on the written part of the assignment, not on the coding.

Training a tagger on almost the complete Penn Treebank requires lots of time or memory and is not possible on eniac due to the restrictions on time and RAM in place. For this reason, the training and applying has been done for you beforehand. Should you ever want to run the tagger yourself (e.g. for a term project), please feel free to contact us.

In case you're curious, the software package we used, `fnl`¹, is an extremely fast reimplementation of Brill's original tagger. Training of the three taggers was done in 30 to 70 minutes each on alpha. The same task used to take us two weeks on comparable machines using Brill's code.

Motivating the problem

The Penn Treebank contains 50,000 sentences (1,000,000 words) of WSJ text from 1989. Its current version also contains about the same amount of words from the Brown corpus and is available on eniac under `/project/cis/nldb/cdrom`. The Brown corpus is a *balanced* corpus – it contains texts from a wide variety of written sources. The WSJ corpus, not surprisingly, consists exclusively of financial news items. You might expect that the Brown corpus would be a better choice for training taggers and parsers. But the first and second versions of the Penn Treebank only contained the WSJ corpus, and for better or for worse it has established itself as the standard training and test data in the field. But is it really beneficial to train and test on the “same” data? Even though the actual training and test data have of course always been kept disjoint, they are still taken from

¹available at <http://nlp.cs.jhu.edu/~rflorian/fntl/>

the same genre of text. We might observe results that are skewed upwards, and fool ourselves into thinking that our tools perform better than they would ever do in “real life” applications. Plus, training on homogeneous data that’s taken from only one genre might make us find fake generalizations that are not true of the English language in general – in other words, we might overfit.

So what happens to performance when we test on a different corpus (Brown) as opposed to another section of the same corpus (WSJ) that we’ve trained on? And what happens when we make our training data more heterogeneous by switching to the Brown corpus, or training on parts of both corpora? In this assignment, you will answer these questions for one specific case: POS tagging. You will be asked to compare the performance of various instances of the Brill tagger. You will evaluate both the overall accuracy of the taggers and their performance on the level of individual tags. Error analysis is a tedious task, but the only way to figure out what’s wrong with your system is to sit down and look at the data it produces...

What to turn in

For this homework, you do not need to hand in any of your code. Submit your answers in a single .pdf or .txt file (not Microsoft Word .doc). Please use the electronic submission system.

Description of the files

Download and unpack the file http://www.cis.upenn.edu/~cis530/hw_2008/hw3data.zip. It contains six files, corresponding to the application of three versions of the Brill tagger to two different texts each. The three versions of the tagger are:

brillwsj	Brill’s TBL Tagger, trained on the Wall Street Journal
brillbrown	Brill’s TBL Tagger, trained on the Brown corpus
brillboth	Brill’s TBL Tagger, trained on both Brown and WSJ

The names are given to provide you with an easy way of referring to each tagger in the subsequent problems.

The files are named according to the following convention: *taggername_testfile.txt*. For example, *brillbrown_wsj.txt* is the result of training Brill’s tagger on the Brown corpus and applying it to the WSJ.

For the WSJ corpus, the following tradition has evolved in NLP: sections 2-21 are used for training; section 22 or 24 are held out for development; and section 23 is used as the final test. (Sections 0 and 1 are perceived to be less reliable – the annotators were still warming up.)² The above taggers are trained on sections 2 to 21 and applied to section 23. For the Brown corpus, there is no convention. So each section of the corpus was split in approximately two halves to create two balanced subcorpora, one for training and one for test.

In each file you downloaded, each line contains a token, followed by the tag assigned to it by the tagger, followed by the correct (gold standard) tag. Sentences are followed by a blank line. For example, the line **bidding** VBG NN indicates that the word **bidding** has been incorrectly tagged as a present participle (VBG), but really is a common noun (NN) – or at least that’s what the human annotators considered it. A list of the Penn Treebank tagset is included at the end of this assignment.

²See <http://www.stanford.edu/class/cs224n/handouts/cs224n-section3-corpora.txt>

1 Evaluate the taggers (20 points)

In Python or in your favorite language, write a program that evaluates the performance of each tagger on each corpus. Your program should calculate and print the accuracy in percent for the given tagger output, compared to the given gold standard. The accuracy is defined as the number of correctly tagged tokens divided by the total number of tokens in the corpus. Remember that some lines in the files are blank and do not contain any tokens. You do not need to turn in the source code of your program, however, you must write it on your own.

Report the accuracy of each of the taggers on each corpus. Arrange your results in a 3-by-2 table. Discuss briefly (1 paragraph) the following questions: Can you make any generalizations from these accuracy rates? What are some factors that might account for these scores?

2 Error analysis

2.1 Data preparation (10 points)

In this exercise, you are asked for each tagger/corpus combination to examine the errors the tagger makes. Write a program that takes a file containing the tagger output and gold standard and computes a *confusion matrix* – a table that records for each combination of two non-identical tags a and b the number of times that a token was mistakenly tagged as a when its goldstandard tag was in fact b .

Some of the words in the Penn Treebank have been tagged with two tags rather than one. This also shows up in the tagger output. The two tags are then separated by a vertical bar:

```
CD|NN
```

For the purpose of this assignment, please treat these double tags as just another kind of tag, i.e. do not map them to either of the two tags, but consider them a single tag.

You do not need to hand in your code, nor the six confusion matrices. Instead, for each tagger/corpus file, hand in a table showing the top ten entries of its confusion matrix with their respective number of occurrences in the file. For example, for the file *brillbrown_wsj.txt*, your table might look like this:

```
Trained on: Brown
Tested on: WSJ
Tagger output | Gold standard | Number of occurrences
JJ            | NN            | 24
NN            | NNP           | 22
...
```

These are not the actual entries you will get for this file.

2.2 Analysis (70 points)

Discuss your findings (1-3 pages). You should at least answer the following questions. We are aware that a strong linguistic background is not a prerequisite for the class, and we will take

this into consideration when we review your responses. It is all right if some of your answers are speculative, as long as you explain your reasoning clearly and coherently. You might find it useful to consult the tagging guidelines that were used by the annotators, available under `ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz`.

- How does varying the training and test data affect the rankings, in other words, how much overlap is there in the errors? Given what you know about the different nature of the corpora, how do you explain your observations?
- What are the most frequent errors overall? What do their POS tags stand for? Give examples of sentences or phrases (i.e. sentence parts) that contain these errors and discuss why an automatic tagger might have trouble with them. If there is significant variation across taggers and/or corpora, and if you have ideas about them, you might also want to discuss for some errors why they might be more severe in some cases and less in others.
- How could tagger performance be increased? Would it help to write additional transformational rules and add them to the rule list generated by training the Brill corpus? If so, give an example of such a rule. If not, explain why not.

3 Debriefing

1. Approximately how many hours did you spend on this assignment?
2. Would you rate it as easy, moderate, or difficult?
3. How deeply do you feel you understand the material it covers (0%-100%)?
4. Any other comments?

This question is intended to help us calibrate the homework assignments. Your answers will not affect your grade.

4 Appendix: The Penn Treebank Tagset

1. CC Coordinating conjunction
2. CD Cardinal number
3. DT Determiner
4. EX Existential there
5. FW Foreign word
6. IN Preposition or subordinating conjunction
7. JJ Adjective
8. JJR Adjective, comparative
9. JJS Adjective, superlative
10. LS List item marker
11. MD Modal
12. NN Noun, singular or mass
13. NNS Noun, plural
14. NNP Proper noun, singular
15. NNPS Proper noun, plural
16. PDT Predeterminer
17. POS Possessive ending
18. PRP Personal pronoun
19. PRP\$ Possessive pronoun
20. RB Adverb
21. RBR Adverb, comparative
22. RBS Adverb, superlative
23. RP Particle
24. SYM Symbol
25. TO to
26. UH Interjection
27. VB Verb, base form
28. VBD Verb, past tense
29. VBG Verb, gerund or present participle
30. VBN Verb, past participle
31. VBP Verb, non-3rd person singular present
32. VBZ Verb, 3rd person singular present
33. WDT Wh-determiner
34. WP Wh-pronoun
35. WP\$ Possessive wh-pronoun
36. WRB Wh-adverb
37. # Pound sign
38. \$ Dollar sign
39. . Sentence-final punctuation
40. , Comma
41. : Colon, semi-colon
42. (Left bracket character
43.) Right bracket character
44. " Straight double quote
45. ' Left open single quote
46. '' Left open double quote
47. ' Right close single quote
48. '' Right close double quote