Chen, Ethan
November 5, 2020
CS10, 20F, Pierson

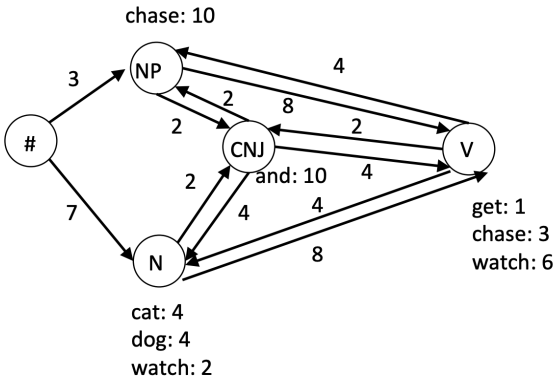
Discussion of PS5 - The Viterbi Decoding Algorithm

First, I should mention that I coded my program to read from files before I did major testing. I thought that it is easier to test with sentences rather than inputting arrays of words into a method.

I decided that the best way to test if my algorithm works was to use a similar example as the one from the programming drill. In order to do this, I created a set of sentences that would create the graph from the drill, although perhaps with separate values. The sentences and tags I used:

Chase watch dog and cat .
Chase watch dog chase cat .
Chase and dog get watch .
Cat chase and watch Chase .
Cat watch Chase chase dog .
Cat chase dog and Chase watch cat and dog .
Cat get dog and chase watch .
Dog watch chase and cat get watch .

NP V N CNJ N .
NP V N V N .
NP CNJ N V N .
N V CNJ V NP .
N V NP V N .
N V N CNJ NP V N CNJ N .
N V N CNJ V N .
N V NP CNJ N V N .



I also created some test sentences (same as the one given programming drill) to test the sentences.

Chase .
Chase get watch and cat .
Dog and cat chase Chase .

With expected tags:

NP .
NP V N CNJ N .
N CNJ N V NP .

And sure enough, I got

```
NP .
NP V N CNJ N .
N CNJ N V NP .
```

So that's great, I got the output I was hoping for. And if this was not convincing enough, I went ahead and tested it on the test set and the brown set. Here is the side by side, with the expected results on the left, and the results from the algorithm on the right:

```
1  N P DET N VD CNJ DET ADJ N N V V VN ADV .
2  ADV , PRO V VN DET N N N V DET ADV V ADJ TO V DET NUM P NUM N P N N N .
3  DET N N N VD ADV ADV V DET N ADV TO V DET N NUM P N NUM N P ADJ N N N P ADJ N N N .
4  NP ADV VD DET N P DET N , WH PRO V VN ADV ADV CNJ PRO N P NUM ADJ N , P DET ADJ N P DET N N P N CNJ N .
5  DET N , WH NP VD PRO V ADV , MOD V N , N N , N N CNJ DET N TO V ADJ N P DET N N .
6  PRO ADV VD CNJ * DET N V VN P DET N , CNJ DET N V VN ADV DET N .
7  NP VD P ADJ N N CNJ PRO VD ADJ N N P P NUM N VN P DET N N , VD DET N P DET ADJ N N , CNJ V * V PRO P DET ADJ N .
8  CNJ PRO V ADJ CNJ DET ADJ N V V VN TO V DET DET N .
9  PRO VD DET N DET N MOD V DET N N P ADV P DET N CNJ MOD ADV V TO V ADV DET DET N P DET VG N N TO V DET N N .
10 PRO MOD V NUM P DET N N P DET NP N P DET N CNJ DET DET P DET NP N .
11 `` WH V CNJ PRO VD N .
12 DET N WH VBZ N N N N N CNJ PRO MOD V VN P DET N P DET N P N P N P DET N P NUM N CNJ NP NP VBZ TO V DET N .
13 PRO VD NP NP CNJ DET N CNJ PRO MOD V DET ADJ N CNJ V ADV P DET N CNJ ADV CNJ ADJ .
14 DET N
15 PRO MOD ADV VN P NP NP NP '' .
16 NP VD CNJ PRO MOD V P ADJ N P VG N P DET N P NP , NUM .
17 CNJ ADJ , PRO VD DET N TO V P N P NP NP NP NP , DET NP N .
18 N P DET N P N CNJ DET ADJ N V VN P N CNJ ADJ N .
19 N VBZ
20 `` PRO DET N '' .
21 DET N VD PRO V DET ADJ TO V WH MOD V VN CNJ DET N V VN P NUM N .
22 N NUM , P DET N V VN P NUM N .
23 DET N P DET N P VG ADV DET P DET N VG P DET N N N MOD V DET N P VG CNJ PRO MOD V DET VN N N .
24 NP MOD V DET N N CNJ DET N TO V DET N P ADJ N P DET N P N .
25 N P DET N N N MOD ADV V NP DET N TO V DET N .
26 PRO VD CNJ DET N N P N V VN N P DET N . . .
27 MOD V N
28 NP NP VD ADV CNJ DET N V P VG DET N P VG P NP NP N .
29 DET N VN ADJ N CNJ N . .
30 -- PRO ADJ N P DET N
31 -- CNJ PRO N P DET ADJ N P NP .
32 ' N VBZ N '
```

```
1  N P DET N VD CNJ DET ADJ N N V V VN ADV .
2  ADV , PRO V VN DET N N N V DET ADV V ADJ TO V DET NUM N NUM N P N N N .
3  DET N N N VD EX ADV V DET N ADV TO V DET N NUM P N NUM N P ADJ N N N P ADJ N N N .
4  NP ADV VD DET N P DET N , WH PRO V VN ADV ADV P PRO N P NUM ADJ N , P DET ADJ N P DET N N P N CNJ N .
5  DET N , WH NP VD PRO VD ADV , MOD V N , N N , N N CNJ DET N TO V ADJ N P DET N N .
6  PRO ADV VD CNJ * DET N V VN P DET N , CNJ DET N V VN ADV DET N .
7  NP VD P ADJ N N CNJ PRO VD ADJ N N P ADV NUM N VN P DET N N , VD DET N P DET ADJ N N , CNJ V * V PRO P DET ADJ N .
8  CNJ PRO V ADJ CNJ DET ADJ N V V VN TO V DET DET N .
9  PRO VD DET N DET N MOD V DET N N P ADV N DET N CNJ MOD ADV V TO V ADV DET DET N P DET VG N N TO V DET N N .
10 PRO MOD V NUM P DET N N P DET NP N P DET N CNJ DET DET P DET NP N .
11 `` DET V CNJ PRO VD N .
12 DET N WH VBZ N N N VBZ N CNJ PRO MOD V VN P DET N P DET N P N P N P DET N P NUM N CNJ NP NP VBZ TO V DET N .
13 PRO VD NP NP CNJ DET N CNJ PRO MOD V DET ADJ N CNJ V ADV P DET N DET ADV CNJ ADJ .
14 VBZ N
15 PRO MOD ADV V P NP NP N '' .
16 NP VD CNJ PRO MOD V P ADJ N P VG N P DET N P NP , NUM .
17 CNJ ADJ , PRO VD DET N TO V P N P NP NP NP NP , DET NP N .
18 N P DET N P N CNJ DET ADJ N V VN P N CNJ ADJ N .
19 N VBZ
20 `` PRO DET N '' .
21 DET N VD PRO V DET ADV TO V WH MOD V VN CNJ DET N V VN P NUM N .
22 N NUM , P DET N N VN P NUM N .
23 DET N P DET N P VG ADV DET P DET N VG P DET N N N MOD V DET N P VG CNJ PRO MOD V DET VN N N .
24 NP MOD V DET N N CNJ DET N TO V DET N P ADJ N P DET N P N .
25 N P DET N N N MOD ADV V NP DET N TO V DET N .
26 PRO VD CNJ DET N N P N N VD N P DET N NP . .
27 MOD V N
28 NP NP VD ADV CNJ DET N V P VG DET N P VG P NP NP N .
29 DET N VD ADJ N CNJ N . .
30 -- PRO ADJ N P DET N
31 -- CNJ PRO N P DET ADJ N P NP .
32 ' N VBZ N '
```

If you look really closely, there are small errors, such as the first tag on line 11 following the ‘. However, otherwise, they are nearly identical. And, as the stats show, this algorithm has a 96.48% tag success rate.

Performance, as expected, was quite strong, with aforementioned 96.48% success rate on the Brown test sentences and
Most sentences were tagged correctly. For example, a very simple sentence such as “The cow jumped over the moon” is always tagged correctly. It was very difficult to come up with sentences that the brown-corpus decoder failed: however eventually I found a few.

```
Peter Piper picked a peck of pickled peppers .
```

```
NP NP VD DET N P VN . .
```

For some reason, the computer saw “pickled” as a past participle instead of as an adjective and the peppers as a “.”. I believe that this may be because of how infrequently verbs used as adjectives occurs in official documents, as the Brown Corpus seems to be. That got me thinking, perhaps more common language and slang is also impossible to pick up, as well as names that may also be used as other words as well.

Somehow though, the decoder managed to pick up the sentence “what good my homie how you doin today” with no problem, despite both the incorrect grammar and the fact that this definitely did not show up in the corpus.

One more thing I noticed while testing the decoder trained by the Brown Corpus was it’s surprisingly low accuracy rating on the test sentences. Of course, now looking at it, I noticed that most were just because the test doesn’t differentiate between tenses. However, the last sentence was a genuine error. “My dog trains to bark”. The decoder decided that “trains” was a noun. After scouring the brown corpus for occurrences of “trains”, there were 4 occurrences of the word as a noun to just 1 as a verb. Therefore, choosing sentences that have words rarely found in the corpus but that can take on different parts of speech can fool the decoder.

As for the U-value, I have tested multiple different values to try to optimize performance. I noticed that when an extremely low value is chosen, (-30, -100, -1000, etc.) the result does not end up changing - this is because all words that do not exist in a certain state are immediately removed from participation for that state. However, I also realized that if a high-ish value (-5) is chosen, success greatly decreases. Therefore, the testing is dependent on the U-value up to a certain point, but I also figured that if you choose the correct U-value, you could actually increase performance from this case where we lose the possibilities of words we haven’t seen. After trial and error, I eventually found it as 1.5x the worst/lowest value of any word at any state. For the Brown Corpus, this value is roughly -18, but it increases the performance from 35109/36394 to 35113/36394. The decoder found 4 more correct tags.