

## Advanced Methods in NLP - Assignment 3: Tagging

### Question 2

```
D:\Daniel\Msc\AdvancedMethodsInNaturalLanguageProcessing\Homework\assignment3\assignment3\code>python most_frequent.py
replaced: 0.0352368561769
replaced: 0.0494304160331
dev: most frequent acc: 0.921105765636

D:\Daniel\Msc\AdvancedMethodsInNaturalLanguageProcessing\Homework\assignment3\assignment3\code>
```

Accuracy: ~0.9211

### Question 3

```
D:\Daniel\Msc\AdvancedMethodsInNaturalLanguageProcessing\Homework\assignment3\assignment3\code>python hmm.py
replaced: 0.0352368561769
replaced: 0.0494304160331
Performing grid search on lambda values...

11=0.00 11=0.05 11=0.10 11=0.15 11=0.20 11=0.25 11=0.30 11=0.35 11=0.40 11=0.45 11=0.50 11=0.55 11=0.60 11=0.65 11=0.70 11=0.75 11=0.80 11=0.85 11=0.90 11=0.95
12=0.00 0.9186 0.9261 0.9318 0.9356 0.9384 0.9408 0.9422 0.9440 0.9452 0.9463 0.9467 0.9474 0.9476 0.9482 0.9520 0.9525 0.9530 0.9532 0.9537 0.9533
12=0.05 0.9210 0.9283 0.9330 0.9361 0.9387 0.9403 0.9420 0.9432 0.9443 0.9453 0.9486 0.9493 0.9500 0.9507 0.9515 0.9518 0.9523 0.9525 0.9521 -
12=0.10 0.9244 0.9300 0.9341 0.9369 0.9387 0.9408 0.9424 0.9436 0.9473 0.9478 0.9483 0.9490 0.9496 0.9503 0.9508 0.9514 0.9517 0.9520 - -
12=0.15 0.9269 0.9315 0.9345 0.9374 0.9390 0.9407 0.9423 0.9464 0.9473 0.9478 0.9483 0.9489 0.9495 0.9503 0.9504 0.9510 0.9510 - -
12=0.20 0.9288 0.9326 0.9352 0.9371 0.9393 0.9405 0.9447 0.9459 0.9468 0.9479 0.9485 0.9487 0.9493 0.9498 0.9501 0.9506 - -
12=0.25 0.9292 0.9330 0.9355 0.9375 0.9419 0.9434 0.9448 0.9454 0.9468 0.9477 0.9479 0.9487 0.9491 0.9495 0.9497 - -
12=0.30 0.9302 0.9330 0.9357 0.9379 0.9420 0.9434 0.9447 0.9454 0.9464 0.9472 0.9479 0.9483 0.9488 0.9493 - -
12=0.35 0.9308 0.9338 0.9361 0.9403 0.9423 0.9439 0.9445 0.9454 0.9461 0.9471 0.9481 0.9484 0.9487 - -
12=0.40 0.9316 0.9343 0.9392 0.9412 0.9428 0.9441 0.9449 0.9453 0.9465 0.9472 0.9479 0.9485 - -
12=0.45 0.9321 0.9352 0.9397 0.9421 0.9431 0.9446 0.9450 0.9460 0.9466 0.9475 0.9480 - -
12=0.50 0.9331 0.9386 0.9409 0.9426 0.9439 0.9449 0.9453 0.9461 0.9471 0.9478 - -
12=0.55 0.9343 0.9393 0.9414 0.9428 0.9441 0.9453 0.9457 0.9466 0.9474 - -
12=0.60 0.9379 0.9402 0.9421 0.9437 0.9447 0.9456 0.9462 0.9469 - -
12=0.65 0.9389 0.9409 0.9432 0.9441 0.9453 0.9462 0.9470 - -
12=0.70 0.9395 0.9421 0.9435 0.9447 0.9460 0.9469 - -
12=0.75 0.9410 0.9427 0.9443 0.9457 0.9469 - -
12=0.80 0.9417 0.9430 0.9454 0.9466 - -
12=0.85 0.9426 0.9442 0.9462 - -
12=0.90 0.9431 0.9455 - -
12=0.95 0.9420 - -

Found best lambda: 0.9 0.0 0.1.
dev: acc hmm viterbi: 0.95368547

D:\Daniel\Msc\AdvancedMethodsInNaturalLanguageProcessing\Homework\assignment3\assignment3\code>
```

Accuracy: ~0.9537

### Question 4

*[Note: We got a MemoryError when running this part on Nova server, even with very little features. Everything worked fine on our PC.]*

#### Optimizations

We've implemented the following prediction optimizations:

- *Probability and feature caching* – We use the dictionaries **features\_cache** and **predictions\_cache** to avoid excessive calculation of features for similar sentences, and avoid multiple computations of probabilities for the same input, respectively.
- *Looking at relevant tags only for each word* – We use the dictionary **tagsOfWord** to store, for each word, all the possible parts-of-speech we've seen for that word, thus map each word to relevant tags only.

#### Results

dev: acc memm greedy: 0.951940573822

dev: acc memm viterbi: 0.954184011766

## Error analysis

We've sampled and analyzed erroneous results, to which we offer the following explanations:

- Uses of words in a rare part-of-speech form -  
This is the classic, expected reason for errors in tagging; some words that usually play a certain grammatical part, can be more rarely used for different grammatical purposes. For example, in the sentence "What's more, such short-term cataclysms are survivable and are no cause for panic **selling**", we mis-tagged the word **selling** as VBG (verb) while it is actually a NN (noun). This makes sense, because the word 'selling' is much more frequently used as a verb.
- Training data is not rich enough -  
For example, in the sentence: "**Anybody** can vote as they want," said Rep. William Lehman..." we mis-tagged **Anybody**. We were surprised to discover that this word appears only one time in the training set (in the capitalized form & at the beginning of sentence), which can explain the mistake.
- Errors and ambiguities in training data –  
We've found several sentences that are quite ambiguous, or even words that are objectively tagged incorrectly.  
For example, in the sentence "Gold was nowhere the spectacular **performer** it was two years ago on Black Monday", we tagged the word **performer** as NN while it should have been tagged, according to the dev set, as JJ. This sentence is quite hard to understand, so it's difficult to justify this tagging.