Shreenidhi_Hegde_HW3

October 21, 2021

1 HW3 Solution Explanation

Task 1 Vocabulary Creation:

I iterate through each row in the training data and create the "vocab" dictionary which contains the word as a key and it's total number of occurances in the training data as value. Similarly, dictionary of all the tags and it's total number of occurances is stored in dictionary named "pos_dict". I have kept the threshold as 2 and replaces words which has less than the threshold with the special token '< unk >' I created the vocabulary using the training data and did output the vocabulary into vocab.txt.

1) What is the selected threshold for unknown words replacement?

```
........ The selected threshold for unknown words replacement is 2 ........
```

2) What is the total size of your vocabulary?

```
........ Total size of vocabulary is 23183 ........
```

3) What is the total size of your vocabulary and what is the total occurrences of the special token '< unk >' after replacement?

```
........ The total occurrences of the special token '< unk >' after replacement is 20011 .....
```

Task 2 Model Learning:

Emission and transition parameters in HMM are in the following formulation:

t(s |s) = count(s→s)/ count(s)

e(x|s) = count(s→x)/count(s)

where t(· | ·) is the transition parameter and e(· | ·) is the emission parameter. I have calculated transmission and emission probabilities according to the above formula. After learning a model , I have output the learned model into a model file in json format, named hmm.json. The model file contains two dictionaries for the emission and transition parameters, respectively. The first dictionary, named transition, contains items with pairs of (s; s0) as key and t(s0js) as value. The second dictionary, named emission, contains items with pairs of (s; x) as key and e(xjs) as value.

1) How many transition and emission parameters in your HMM?

```
........ The number of emission parameters is  30303 ........

........ The number of transmission parameters is  1392 .......
```

Task 3 Greedy Decoding withHMM:

Greedy decoding: just go left-to-right and pick the highest probability choice each time. For i = 1 to N: choose the tag that maximizes the transition probability × emission probability.

1) What is the accuracy on dev data?

```
........ Accuracy of dev data is 93.52718385439199 ........
```

Task 4 Viterbi Decoding with HMM

The Viterbi algorithm is a dynamic programming algorithm for obtaining the maximum a posteriori probability estimate of the most likely sequence of hidden states—called the Viterbi path—that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models (HMM).

Intuition: the best path of length i ending in state t must include the best path of length i−1 to the previous state. So, – Find the best path of length i−1 to each state. – Consider extending each of those by 1 step, to state t. – Take the best of those options as the best path to state t.

1) What is the accuracy on dev data?

```
........ Accuracy of dev data is 94.8114245812176 ........
```

[1]:
```
import pandas as pd
import csv
from collections import defaultdict
```

1.1 Task 1 Vocabulary Creation

[2]:
```
vocab = {}
pos_dict = {}
sentences = 1
alpha = 1
with open('data/train', newline=
          '') as tsvfile:

    text_data = csv.reader(tsvfile, delimiter='\t')
    for row in text_data:
        try:
            if row[2].strip() not in pos_dict:
                pos_dict[row[2].strip()] = 1
            else:
                pos_dict[row[2].strip()] +=1

            if row[1].strip() not in vocab:
                vocab[row[1].strip()] = 1
            else:
                vocab[row[1].strip()] +=1

        except:
```

```
            sentences += 1
            pass
```

[3]:
```
threshold = 2
unknowns = sum(val if val < threshold else 0 for key,val in vocab.items())
vocab = {key:val for key, val in vocab.items() if val >= threshold}
vocab = dict(sorted(vocab.items(), key=lambda x: x[1], reverse= True))
vocab = {'<unk>':unknowns, **vocab}
```

[4]:
```
print("........ The selected threshold for unknown words replacement is 2 ......
 ↪..\n")
print("........ Total size of vocabulary is "+str(len(vocab))+" ........\n")
print("........ The total occurrences of the special token '< unk >' after␣
 ↪replacement is "+str(unknowns)+" ........\n")
```

… The selected threshold for unknown words replacement is 2 …

… Total size of vocabulary is 23183 …

… The total occurrences of the special token '< unk >' after replacement is 20011 …

[5]:
```
#writing into vocab.txt file

with open('vocab.txt', 'w') as f:
    for idx, (word, count) in enumerate(vocab.items()):
        f.write(str(word)+'\t'+str(idx+1)+'\t'+str(count)+'\n')
```

1.2 Task 2 Model Learning

[6]:
```
prev = "^"
pos_dict = {prev:sentences, **pos_dict}
transmission_data = defaultdict(int)
emission_data = defaultdict(int)
with open('data/train', newline='') as tsvfile:
    text_data = csv.reader(tsvfile, delimiter='\t')
    for row in text_data:
        try :
            transmission_data[(prev,row[2])]+=1
            if row[1] in vocab:
                emission_data[(row[2], row[1])] += 1
            else:
                emission_data[(row[2],'<unk>')] += 1
            prev = row[2]
```

```
        except:
            prev = "^"
            pass
```

[7]:
```
#Here we find the transmission and emission probabilities

transmission_prob = defaultdict()
emission_prob = defaultdict()
for key,value in transmission_data.items():
    transmission_prob[key] = (value+alpha)/
 ↪(pos_dict[key[0]]+alpha*(len(pos_dict)-1))

for key,value in emission_data.items():
    emission_prob[key] = value/pos_dict[key[0]]


for p1 in pos_dict.keys():
    for p2 in pos_dict.keys():
        if p2 == '^':
            continue
        if (p1, p2) not in transmission_prob:
            transmission_prob[(p1, p2)] = alpha/
 ↪(pos_dict[p1]+alpha*(len(pos_dict)-1))
```

[8]:
```
print("........ The number of emission parameters is ",len(emission_data),".....
 ↪...\n")
print("........ The number of transmission parameters is␣
 ↪",len(transmission_data),".......\n")
```

```
… The number of emission parameters is  30303 …

… The number of transmission parameters is  1392 …
```

[9]:
```
##Output learned model into model file in json format - hmm.json

import json

transmission_dict = dict((','.join(k), v) for k,v in transmission_prob.items())
emission_dict = dict((','.join(k), v) for k,v in emission_prob.items())

hmm = {'transmission': transmission_dict, 'emission': emission_dict}
with open("hmm.json", "w") as outfile:
    json.dump(hmm, outfile, indent=4)
```

1.3 Task 3 Greedy Decoding with HMM

[10]:
```
data_dev = []
words = []
pos_tags = []

sent_count = 1
with open('data/dev', newline = '') as tsvfile:
    csv_reader = csv.reader(tsvfile, delimiter = '\t')
    for row in csv_reader:
        try:
            words.append(row[1])
            pos_tags.append(row[2])
        except:
            data_dev.append([sent_count, words, pos_tags])
            sent_count += 1
            words = []
            pos_tags = []
#     data_dev.append([sent_count, words, pos_tags])
df_dev = pd.DataFrame(data_dev,columns = ['sentence_number', 'words',␣
↪'pos_tags'])
df_dev
```

[10]:

	sentence_number	words	pos_tags
0	1	[The, Arizona, Corporations, Commission, autho…	[DT, NNP, NNP, NNP, VBD, DT, CD, NN, NN, NN, I…
1	2	[The, ruling, follows, a, host, of, problems, …	[DT, NN, VBZ, DT, NN, IN, NNS, IN, NNP, NNP, ,…
2	3	[The, Arizona, regulatory, ruling, calls, for,…	[DT, NNP, JJ, NN, VBZ, IN, $, CD, CD, IN, JJ, …
3	4	[The, company, had, sought, increases, totalin…	[DT, NN, VBD, VBN, NNS, VBG, $, CD, CD, ,, CC,…
4	5	[The, decision, was, announced, after, trading…	[DT, NN, VBD, VBN, IN, NN, VBD, .]
…	…	…	…
5521	5522	[The, pilot, union, is, vowing, to, pursue, an…	[DT, NN, NN, VBZ, VBG, TO, VB, DT, NN, WDT, DT…
5522	5523	[But, if, the, board, rejects, a, reduced, bid…	[CC, IN, DT, NN, VBZ, DT, VBN, NN, CC, VBZ, TO…
5523	5524	[The, pilots, could, play, hardball, by, notin…	[DT, NNS, MD, VB, NN, IN, VBG, PRP, VBP, JJ, T…
5524	5525	[If, they, were, to, insist, on, a, low, bid, …	
5525	5526	[Also, ,, because, UAL, Chairman, Stephen, Wol…	

```
5524  [IN, PRP, VBD, TO, VB, IN, DT, JJ, NN, IN, ,, …
5525  [RB, ,, IN, NNP, NNP, NNP, NNP, CC, JJ, NNP, N…

[5526 rows x 3 columns]
```

[11]:
```
df_dev
```

[11]:
```
      sentence_number                                              words  \
0                   1  [The, Arizona, Corporations, Commission, autho…
1                   2  [The, ruling, follows, a, host, of, problems, …
2                   3  [The, Arizona, regulatory, ruling, calls, for,…
3                   4  [The, company, had, sought, increases, totalin…
4                   5  [The, decision, was, announced, after, trading…
…                 …                                                  …
5521             5522  [The, pilot, union, is, vowing, to, pursue, an…
5522             5523  [But, if, the, board, rejects, a, reduced, bid…
5523             5524  [The, pilots, could, play, hardball, by, notin…
5524             5525  [If, they, were, to, insist, on, a, low, bid, …
5525             5526  [Also, ,, because, UAL, Chairman, Stephen, Wol…

                                              pos_tags
0     [DT, NNP, NNP, NNP, VBD, DT, CD, NN, NN, NN, I…
1     [DT, NN, VBZ, DT, NN, IN, NNS, IN, NNP, NNP, ,…
2     [DT, NNP, JJ, NN, VBZ, IN, $, CD, CD, IN, JJ, …
3     [DT, NN, VBD, VBN, NNS, VBG, $, CD, CD, ,, CC,…
4                    [DT, NN, VBD, VBN, IN, NN, VBD, .]
…                                                    …
5521  [DT, NN, NN, VBZ, VBG, TO, VB, DT, NN, WDT, DT…
5522  [CC, IN, DT, NN, VBZ, DT, VBN, NN, CC, VBZ, TO…
5523  [DT, NNS, MD, VB, NN, IN, VBG, PRP, VBP, JJ, T…
5524  [IN, PRP, VBD, TO, VB, IN, DT, JJ, NN, IN, ,, …
5525  [RB, ,, IN, NNP, NNP, NNP, NNP, CC, JJ, NNP, N…

[5526 rows x 3 columns]
```

[12]:
```
def accuracy(predictions, ground_truth):
    correct_matches = 0
    total_words = 0
    for pred, gt in zip(predictions, ground_truth):
        correct_matches += np.sum(np.array(pred) == np.array(gt))
        total_words +=  len(pred)
    accuracy = correct_matches/total_words
    return accuracy*100
```

[13]:
```
import numpy as np
import random
```

```
def greedy_decoding(df):
    predicted_tags_list = []
    accuaracy_list = []

    for idx, (_, words) in df.iterrows():
        predicted_tags = []
        prec_pos_tag = '^'
        for word in words:
            max_proba = 0
            best_tag = random.choice(list(pos_dict.keys()))
            for pos_tag in pos_dict.keys():
                if word not in vocab:
                    word = '<unk>'

                if pos_tag == '^':
                    continue

                if (pos_tag, word) in emission_prob:
                    tp =  transmission_prob[(prec_pos_tag, pos_tag)]
                    ep = emission_prob[(pos_tag, word)]
                    proba = tp*ep
                    if proba > max_proba:
                        max_proba = proba
                        best_tag = pos_tag


            predicted_tags.append(best_tag)
            prec_pos_tag = best_tag

        predicted_tags_list.append(predicted_tags)

    return predicted_tags_list

df_dev_greedy = df_dev.copy()
df_dev_greedy['predicted_tags'] = greedy_decoding(df_dev_greedy.loc[:
 ↪,['sentence_number', 'words']])
```

[14]:
```
df_dev_greedy
```

[14]:

	sentence_number	words
0	1	[The, Arizona, Corporations, Commission, autho…
1	2	[The, ruling, follows, a, host, of, problems, …
2	3	[The, Arizona, regulatory, ruling, calls, for,…
3	4	[The, company, had, sought, increases, totalin…
4	5	[The, decision, was, announced, after, trading…
…	…	…
5521	5522	[The, pilot, union, is, vowing, to, pursue, an…

```
5522             5523  [But, if, the, board, rejects, a, reduced, bid…
5523             5524  [The, pilots, could, play, hardball, by, notin…
5524             5525  [If, they, were, to, insist, on, a, low, bid, …
5525             5526  [Also, ,, because, UAL, Chairman, Stephen, Wol…

                                                  pos_tags  \
0     [DT, NNP, NNP, NNP, VBD, DT, CD, NN, NN, NN, I…
1     [DT, NN, VBZ, DT, NN, IN, NNS, IN, NNP, NNP, ,…
2     [DT, NNP, JJ, NN, VBZ, IN, $, CD, CD, IN, JJ, …
3     [DT, NN, VBD, VBN, NNS, VBG, $, CD, CD, ,, CC,…
4                    [DT, NN, VBD, VBN, IN, NN, VBD, .]
…                                                   …
5521  [DT, NN, NN, VBZ, VBG, TO, VB, DT, NN, WDT, DT…
5522  [CC, IN, DT, NN, VBZ, DT, VBN, NN, CC, VBZ, TO…
5523  [DT, NNS, MD, VB, NN, IN, VBG, PRP, VBP, JJ, T…
5524  [IN, PRP, VBD, TO, VB, IN, DT, JJ, NN, IN, ,, …
5525  [RB, ,, IN, NNP, NNP, NNP, NNP, CC, JJ, NNP, N…

                                           predicted_tags
0     [DT, NNP, NNS, NNP, VBD, DT, CD, NN, NN, NN, I…
1     [DT, NN, VBZ, DT, NN, IN, NNS, IN, NNP, NNP, ,…
2     [DT, NNP, JJ, NN, VBZ, IN, $, CD, CD, IN, VBN,…
3     [DT, NN, VBD, VBN, NNS, VBG, $, CD, CD, ,, CC,…
4                    [DT, NN, VBD, VBN, IN, NN, VBD, .]
…                                                   …
5521  [DT, NN, NN, VBZ, VBG, TO, VB, DT, NN, WDT, DT…
5522  [CC, IN, DT, NN, NN, DT, JJ, NN, CC, VBZ, TO, …
5523  [DT, NNS, MD, VB, JJ, IN, VBG, PRP, VBP, JJ, T…
5524  [IN, PRP, VBD, TO, VB, IN, DT, JJ, NN, IN, ,, …
5525  [RB, ,, IN, NNP, NNP, NNP, NNP, CC, JJ, NNP, N…

[5526 rows x 4 columns]
```

[15]:
```
acc = accuracy(df_dev_greedy['predicted_tags'].tolist(),␣
 ↪df_dev_greedy['pos_tags'].tolist())
acc
```

[15]:
```
93.52718385439199
```

[16]:
```
print("........ Accuracy of dev data is",acc,"........\n")
```

```
… Accuracy of dev data is 93.52718385439199 …
```

[17]:
```
data_test = []
words = []
pos_tags = []
```

```
sent_count = 1
with open('data/test', newline = '') as tsvfile:
    csv_reader = csv.reader(tsvfile, delimiter = '\t')
    for row in csv_reader:
        try:
            words.append(row[1])
        except:
            data_test.append([sent_count, words])
            sent_count += 1
            words = []
            pos_tags = []

df_test = pd.DataFrame(data_test,columns = ['sentence_number', 'words'])
```

[18]:
```
df_test_greedy = df_test.copy()
df_test_greedy['predicted_tags'] = greedy_decoding(df_test_greedy)
with open('greedy.out', 'w') as tsvfile:
    for _, (_, words, predicted_tags) in df_test_greedy.iterrows():
        idx = 0
        for word, pos_tag in zip(words, predicted_tags):
            print ("%d\t%s\t%s" % (idx+1, word, pos_tag), file=tsvfile)
            idx += 1
        if idx != len(df_test_greedy.index):
            print(file=tsvfile)
```

1.4 Task 4 Viterbi Decoding with HMM

[19]:
```
def viterbi_decoding(df):

    prec_pos_tag = '<POS>'
    predicted_tags_list = []
    accuaracy_list = []

    for idx, (_, words) in df.iterrows():
        predicted_tags = []
        T = len(words)
        N = len(pos_dict.items())
        viterbi = [[0]*(T) for _ in range(N)]
        backpointer = [[0]*(T) for _ in range(N)]

        for t, word in enumerate(words):

            if word not in vocab:
                word = '<unk>'
            for s, pos_tag in enumerate(pos_dict.keys()):
                if s == 0:
```

```
                    pass
                elif t == 0:
                    tp =  transmission_prob[('^', pos_tag)]
                    ep = emission_prob[(pos_tag, word)] if (pos_tag, word) in␣
↪emission_prob else 0

                    viterbi[s][0] = tp * ep
                    backpointer[s][0] = 0
                else:
                    backpointer[s][t] = random.randint(1,len(pos_dict)-1)
                    for s1, prec_pos in enumerate(pos_dict.keys()):
                        if s1 == 0:
                            continue

                        if (pos_tag, word) in emission_prob:
                            tp =  transmission_prob[(prec_pos, pos_tag)]
                            ep = emission_prob[(pos_tag, word)]
                            cur_proba = viterbi[s1][t-1]*tp*ep

                            if cur_proba > viterbi[s][t]:
                                viterbi[s][t] = cur_proba
                                backpointer[s][t] = s1


        bestpathprob = -1
        bestpathpointer = 0
        for s in range(1, N):
            if viterbi[s][T-1] >= bestpathprob:
                bestpathprob = viterbi[s][T-1]
                bestpathpointer = s

        bestpath = []
        t = T-1
        pos_list = list(pos_dict.keys())
        while bestpathpointer != 0:
            bestpath.append(pos_list[bestpathpointer])
            bestpathpointer = backpointer[bestpathpointer][t]
            t -= 1

        bestpath = bestpath[::-1]

        predicted_tags_list.append(bestpath)

    return predicted_tags_list

df_dev_viterbi = df_dev.copy(deep = True)
```

```
df_dev_viterbi['predicted_tags'] = viterbi_decoding(df_dev_viterbi.loc[:
↪,['sentence_number', 'words']])
```

[20]:
```
df_dev_viterbi
```

[20]:
```
      sentence_number                                              words  \
0                   1  [The, Arizona, Corporations, Commission, autho…
1                   2  [The, ruling, follows, a, host, of, problems, …
2                   3  [The, Arizona, regulatory, ruling, calls, for,…
3                   4  [The, company, had, sought, increases, totalin…
4                   5  [The, decision, was, announced, after, trading…
…                 …                                                …
5521             5522  [The, pilot, union, is, vowing, to, pursue, an…
5522             5523  [But, if, the, board, rejects, a, reduced, bid…
5523             5524  [The, pilots, could, play, hardball, by, notin…
5524             5525  [If, they, were, to, insist, on, a, low, bid, …
5525             5526  [Also, ,, because, UAL, Chairman, Stephen, Wol…

                                                 pos_tags  \
0     [DT, NNP, NNP, NNP, VBD, DT, CD, NN, NN, NN, I…
1     [DT, NN, VBZ, DT, NN, IN, NNS, IN, NNP, NNP, ,…
2     [DT, NNP, JJ, NN, VBZ, IN, $, CD, CD, IN, JJ, …
3     [DT, NN, VBD, VBN, NNS, VBG, $, CD, CD, ,, CC,…
4                     [DT, NN, VBD, VBN, IN, NN, VBD, .]
…                                                   …
5521  [DT, NN, NN, VBZ, VBG, TO, VB, DT, NN, WDT, DT…
5522  [CC, IN, DT, NN, VBZ, DT, VBN, NN, CC, VBZ, TO…
5523  [DT, NNS, MD, VB, NN, IN, VBG, PRP, VBP, JJ, T…
5524  [IN, PRP, VBD, TO, VB, IN, DT, JJ, NN, IN, ,, …
5525  [RB, ,, IN, NNP, NNP, NNP, NNP, CC, JJ, NNP, N…

                                       predicted_tags
0     [DT, NNP, NNS, NNP, VBD, DT, CD, NN, NN, NN, I…
1     [DT, NN, VBZ, DT, NN, IN, NNS, IN, NNP, NNP, ,…
2     [DT, NNP, JJ, NN, VBZ, IN, $, CD, CD, IN, JJ, …
3     [DT, NN, VBD, VBN, NNS, VBG, $, CD, CD, ,, CC,…
4                     [DT, NN, VBD, VBN, IN, NN, VBD, .]
…                                                   …
5521  [DT, NN, NN, VBZ, VBG, TO, VB, DT, NN, WDT, DT…
5522  [CC, IN, DT, NN, VBZ, DT, JJ, NN, CC, VBZ, TO,…
5523  [DT, NNS, MD, VB, VBN, IN, VBG, PRP, VBP, JJ, …
5524  [IN, PRP, VBD, TO, VB, IN, DT, JJ, NN, IN, ,, …
5525  [RB, ,, IN, NNP, NNP, NNP, NNP, CC, JJ, NNP, N…

[5526 rows x 4 columns]
```

[21]:
```
acc = accuracy(df_dev_viterbi['predicted_tags'].tolist(),␣
↪df_dev_viterbi['pos_tags'].tolist())
acc
```

[21]: 94.8114245812176

[22]:
```
print("........ Accuracy of dev data is",acc,"........\n")
```

… Accuracy of dev data is 94.8114245812176 …

[24]:
```
df_test_viterbi = df_test.copy()
df_test_viterbi['predicted_tags'] = viterbi_decoding(df_test_viterbi)
with open('viterbi.out', 'w') as tsvfile:
    for _, (_, words, predicted_tags) in df_test_viterbi.iterrows():
        idx = 0
        for word, pos_tag in zip(words, predicted_tags):
            print ("%d\t%s\t%s" % (idx+1, word, pos_tag), file=tsvfile)
            idx += 1
        if idx != len(df_test_viterbi.index):
            print(file=tsvfile)
```

[25]:
```
df_test_viterbi
```

[25]:

	sentence_number	words	predicted_tags
0	1	[Influential, members, of, the, House, Ways, a…	[JJ, NNS, IN, DT, NNP, NNPS, CC, NNP, NNP, VBD…
1	2	[The, bill, ,, whose, backers, include, Chairm…	[DT, NN, ,, WP$, NNS, VBP, NNP, NNP, NNP, -LRB…
2	3	[The, bill, intends, to, restrict, the, RTC, t…	[DT, NN, VBZ, TO, VB, DT, NNP, TO, NNP, NNS, R…
3	4	[``, Such, agency, `, self-help, ', borrowing,…	[``, JJ, NN, ``, NNP, POS, NN, VBZ, JJ, CC, JJ…
4	5	[The, complex, financing, plan, in, the, S&L, …	[DT, JJ, NN, NN, IN, DT, NN, NN, NN, VBZ, VBG,…
…	…	…	…
5456	5457	[The, men, also, will, be, faced, with, bridgi…	[DT, NNS, RB, MD, VB, VBN, IN, VBG, DT, NN, NN…
5457	5458	[Says, Peter, Mokaba, ,, president, of, the, S…	[VBZ, NNP, NNP, ,, NN, IN, DT, NNP, NNP, NNP, …
5458	5459	[They, never, considered, themselves, to, be, …	[PRP, RB, VBN, PRP, TO, VB, NN, RB, .]
5459	5460	[At, last, night, 's, rally, ,, they, called, …	[IN, JJ, NN, POS, NN, ,, PRP, VBD, IN, PRP$, N…
5460	5461	[``, We, emphasize, discipline, because, we, k…	

```
5460  [``, PRP, VB, NN, IN, PRP, VBP, IN, DT, NN, VB…

[5461 rows x 3 columns]
```

[ ]: