AI6122 Text Data Management & Analysis

Topic: Exercise 2 Discussions

Q2.1 Basic Processing

- Write a program to do the following tasks:
  - Download the Web page of a given link and extract the text content of the page
  - Split the text into sentences and count the number of sentences
  - Split the text into tokens, and count the number tokens and number of unique tokens (i.e., token types)
  - Find lemmas (or stems) of the tokens and count lemma types
  - Do stemming on the tokens and count unique ‘stemmed’ tokens

- You may use any tools, including nltk, LingPipe, and Stanford NLP software.

Sample code based on NLTK

```
import urllib.request
import nltk
from bs4 import BeautifulSoup

print ('Ready to collect pages....')
with urllib.request.urlopen ('https://en.wikipedia.org/wiki/Natural_language_processing')
as response:
    html=response.read()
print ('HTML page downloaded.')

text = BeautifulSoup(html, "lxml").get_text()
print('Clean text extracted from HTML')
```

Sample code based on NLTK

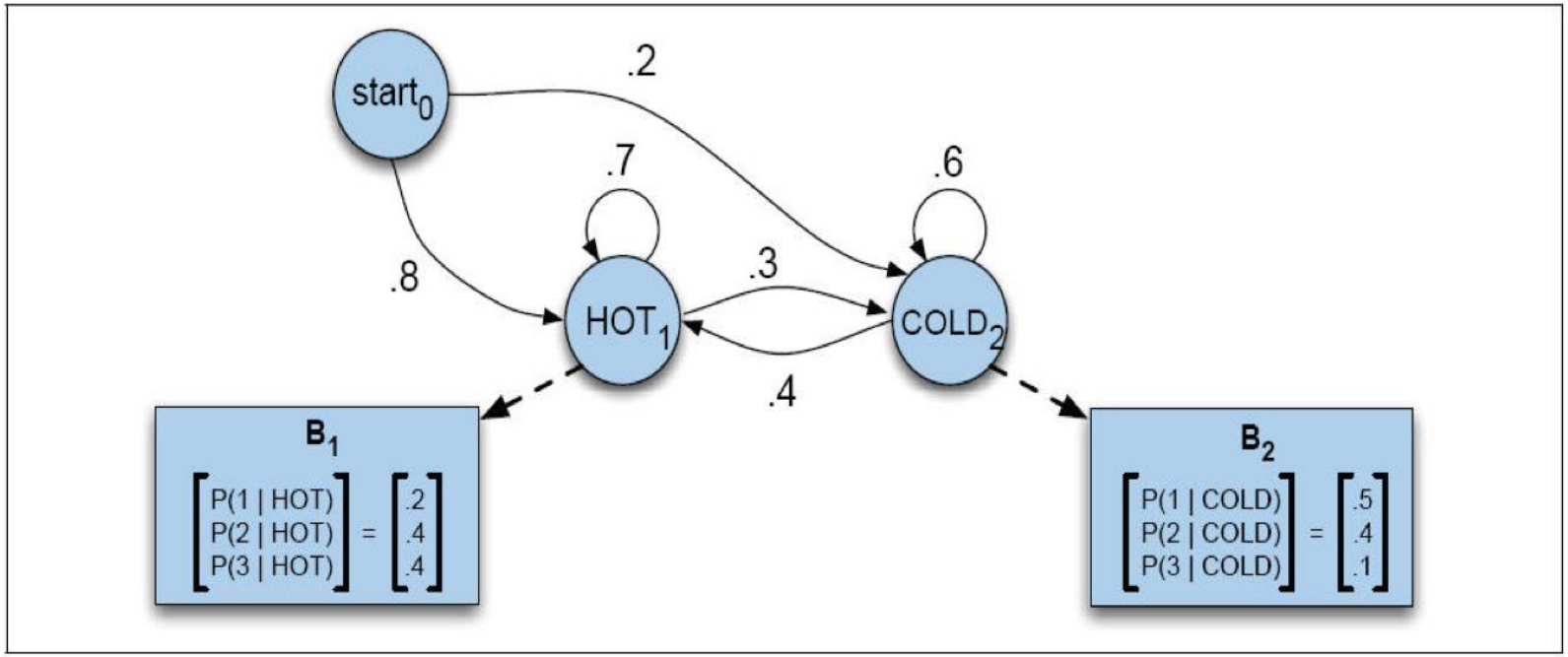
```
sentences = nltk.tokenize.sent_tokenize(text)
print ('Number of sentences: '+ str(len(sentences)))

tokens= nltk.tokenize.word_tokenize(text)

print ('Number of tokens: '+ str(len(tokens)))

token_types = list(set(tokens))

print ('Number of token types: '+ str(len(token_types)))
```

Sample code based on NLTK

```
wnl=nltk.stem.WordNetLemmatizer()

stemmer = nltk.stem.porter.PorterStemmer()
lemma_types= set()
stemmed_types= set()

for token_type in token_types:
    lemma_types.add(wnl.lemmatize(token_type))
    stemmed_types.add(stemmer.stem(token_type))

print ('Number of lemma types: '+ str(len(lemma_types)))
print ('Number of stemmed types: '+ str(len(stemmed_types)))
```

Q2.2 HMM and Viterbi

- Write a program (you may use third-party APIs), using Viterbi algorithm with the given HMM model, to compute the most likely weather sequences for each of the two following observation sequences. Sequence (A): 312312312 Sequence (B): 311233112



Sample code in Python

```
import numpy as np
from hmmlearn import hmm

states = ["-Hot-", "-Cold-"]
#array index:  0,   1

observations = ["1", "2", "3"]
#array index:   0, 1, 2

model = hmm.MultinomialHMM(n_components=2)
model.startprob_ = np.array([0.8, 0.2])
model.transmat_ = np.array([[0.7, 0.3],
                            [0.4, 0.6]])
model.emissionprob_ = np.array([[0.2, 0.4, 0.4], [0.5, 0.4, 0.1]])
```

Sample code in Python

```
obs_sqn_1 = np.atleast_2d([2, 0, 1, 2, 0, 1, 2, 0, 1, 0, 0, 1, 1, 0, 1]).T

#print(model.decode(obs_sqn_1))
logprob, decode_states_1 = model.decode(obs_sqn_1, algorithm="viterbi")
print ("Seq 1 decoded states: " + "".join(map(lambda x: states[x], decode_states_1)))

obs_sqn_2 = np.atleast_2d([2, 0, 0, 1, 2, 2, 0, 0, 1]).T
logprob, decode_states_2 = model.decode(obs_sqn_2, algorithm="viterbi")
print ("Seq 2 decoded states: " + "".join(map(lambda x: states[x], decode_states_2)))
```