ASSIGNMENT 2

COMP 550, Fall 2019

Due: Friday, October 18^{th}, 2019, 9:00pm.

You must do this assignment individually. You may consult with other students orally, but may not take notes or share code, and you must complete the final submission on your own.

Question 1:	20 points
Question 2:	40 points
Question 3:	40 points
	100 points total

Assignment

Question 1: Sequence Models (20 points)

Prove or disprove each claim below.

1. The Viterbi algorithm returns the globally optimal state sequence for hidden Markov models, $\arg\max_{\vec{Q}} P(\vec{Q}, \vec{O}|\theta)$. You may assume a first-order Markov transition model for simplicity (i.e., a bigram transition model).
2. The negative log likelihood loss for linear-chain conditional random fields, $-\sum_i \log P(Y^{(i)}|X^{(i)})$, is equivalent to the sum of the cross entropy losses incurred over every timestep in the same sequences, $-\sum_i \sum_t \log(p_t^i)$. Here, i indexes over samples in your labelled training corpus, t indexes over timesteps in sample i, and p_t^i is the probability of the gold-standard label for sample i at time t.

Question 2: Grammar for French (40 points)

In this question, you will develop a context-free grammar for a fragment of French. Your grammar must account for various aspects of the French language, as listed below.

Basic sentence word order in the present

The basic word order in French is Subject-Verb-Object, as in English:

(1) Je regarde la télévision.
I watch the television

(2) Le chat mange le poisson.
The cat eats the fish

Subject-verb agreement

Just as in English, the subject must agree with the verb in number and person:

(3) Tu regardes la télévision.
You(2Sg) watch the television

(4) Il regarde la télévision.
He watches the television

(5) Nous regardons la télévision.
We watch the television

(6) Vous regardez la télévision.
You(2Pl) watch the television

(7) Ils regardent la télévision.
They(Masc.) watch the television

Look up the list of subject pronouns in French, as well as the verb conjugation paradigm for several common verbs using an online website. Include these in your grammar.

Reference: `http://www.wordreference.com/conj/FrVerbs.aspx`

Definite noun phrases and proper names

A definite noun phrase in French follows a similar order as in English (article + noun). However, the article must agree with the noun in number and grammatical gender. Grammatical gender is a more-or-less arbitrary categorization of nouns into either masculine or feminine.

Examples:

(8) Le chat
the(Masc.) cat

(9) La télévision
the(Fem.) television

(10) Les chats
the(Pl.) cats

(11) Les télévisions
the(Pl) televisions

As you can see, there is no distinction in the plural between masculine or feminine.

Some proper names in French do not take articles, just as in English:

(12) Jonathan
Jonathan

(13) Montréal
Montreal

Others do (e.g., *le Canada*), but you do not have to handle them.

Direct object pronouns

When a pronoun is a direct object of the verb, they precede the verb:

(14) Il la regarde.
He it(Fem.) watches.

Look up the list of direct object pronouns in French, and enhance your grammar to account for the word order with direct objects.

Attributive adjectives

Adjectives typically follow the noun that they modify in a noun phrase:

(15) Le chat noir
the(Masc.) cat black

(16) Le chat heureux
the(Masc.) cat happy

However, other adjectives precede the noun:

(17) Le beau chat
the(Masc.) beautiful cat

(18) Le joli chat
the(Masc.) pretty cat

Yet others may precede OR follow the noun, though the meaning usually changes slightly:

(19) La dernière semaine
the(Fem.) last week
the last week (e.g., of the year)

(20) La semaine dernière
the(Fem.) week last
last week (i.e., the one before this week)

In addition, adjectives must agree with the noun that they modify in number and gender:

(21) Les chats noirs
the(Pl.) cats black(Pl.)
the black cats

(22) La télévision noire
the(Fem.) television black(Fem.)
the black television

(23) Les télévisions noires
the(Pl.) televisions black(Fem. Pl.)
the black televisions

Note that adjectives do distinguish masculine from feminine in the plural.

Find several adjectives of each of the three classes above, and incorporate them into your grammar.

References
`http://french.about.com/od/grammar/a/adjectives.htm`
`http://french.about.com/od/grammar/a/adjectives_4.htm`

Examples and submission format

You already have many examples that your grammar should accept (though many of the above examples were only noun phrases, not full sentences). Here are some sentences that your grammar should **reject**:

(24) *Je mangent le poisson.

(25) *Les noirs chats mangent le poisson.

(26) *La poisson mangent les chats.

(27) *Je mange les.

Use the following nonterminals to indicate grammatical categories:

```
S           sentence/clause
NP          noun phrase
VP          verb phrase
N           noun
PN          proper noun
PR          pronoun
V           verb
DT          determiner
A           adjective
```

You may add further non-terminal categories or subdivide them (e.g., V-1Sing) as needed. Don't forget the lexical rules! Include enough lexical items such that each of the syntactic categories can be expressed in at least three different ways.

Write your grammar in a text editor using a predictable, computer-readable format. For instance, here is one possible rule:

```
S -> NP VP
```

Here is another example of a set of four rules (here, they are lexical rules):

```
V-1Sg -> mange | aime | regarde | cherche
```

These are just examples, and are not necessarily the rules you want in your grammar! Ignore punctuation and capitalization in your grammar (just use all lower-case, except for proper names). French has contractions in many cases where a word begins with a vowel (e.g.,*j'aime* rather than **je aime*). You may ignore such issues.

Submit your grammar as a plaintext .txt file. Show instances where your grammar correctly accepts and rejects some sentence. In addition, answer the following questions in your response to the question:

1. What are some advantages of modelling French grammar with a CFG?
2. What are some disadvantages of modelling French grammar with a CFG?
3. What are some aspects of French grammar that your CFG does not handle?

This question is rather open-ended; your grammar will be judged on the following points:

- Whether you followed the specifications above (e.g. names of non-terminals, minimum number of lexical entries)
- Coverage of the required grammatical constructions
- Clarity of the grammar
- The responses to the questions above

You won't get extra points for having many additional lexical items that exhibit the same type of behaviour!

Question 3: Decipherment with HMMs (40 points)

We have intercepted coded communications from our enemies, the League Against Natural Language Processing. We have obtained short samples containing both ciphertext (i.e., encrypted text) and its associated plaintext (i.e., plain English). Being good experimentalists, we have separated the data into

a training set and a test set, so that we know that we will be able to decrypt future coded messages from this despicable organization.

Actually, we have intercepted three different ciphers of the sample text:

1. The first cipher is quite archaic, first used by Julius Caesar. Each letter in the plain text is shifted by a fixed amount to produce the cipher text.
2. The second cipher is a more complex cipher, in which there are two letter substitution ciphers. When encrypting each letter, one of the two is randomly chosen, and that cipher is used to generate the ciphertext letter.
3. The third cipher was invented by Gilbert Vernam. The cipher text is produced by adding the integer values of the a key and the plain text. We also know that the key is produced by shifting characters in plain text by 3 places from right to left. For example if you need to encrypt the plain text **'nlp is awesome.'** the key you will use is **' is awesome.nlp'**. To generate the cipher, you need to add the integer values of the two strings character by character. Integer values for characters from a-z are 0-25 and for ' ', ',', '.' are '26', '27' and '28' respectively. Thus the cipher text will be **'ktexilbshqwnzpo'**.

Note: The three cipher texts are attached.

The plaintext and ciphertext alphabets are both the 26 letters of the alphabet (lowercase), plus the space, comma, and period, for a total of 29 symbols.

In this question, we will explore several HMM models for solving these ciphers. Each HMM sample will be a sentence, and each time step within a sequence will be one character. The hidden layer will consist of the plaintext characters, while the observed layer will consist of ciphertext characters.

Standard HMM

Implement a system which trains a standard HMM on the training set using MLE, and tests on the testing data. Use NLTK's `nltk.tag.hmm`. Report per-token accuracy on both ciphers. Print out the results of decoding on the test set. What do you observe about the performance of the model?

Your code should run in the following way:

```
python decipher.py <cipher_folder>
```

It should print out the deciphered test set, and report the accuracy score. Since training should not take that much time, you do not need to save the model output.

Laplace smoothing

Let's see if smoothing will help improve performance. Modify your code and add an option that implements Laplace smoothing during training. The simplest method for doing so will likely be to modify the `HiddenMarkovModelTagger` object that you got from training from the previous step. Consult NLTK's API in order to figure out how to do this. You may find the page on `nltk.probability` useful as well.

It should be possible to turn Laplace smoothing on at the command line in the following way:

```
python decipher.py -laplace <cipher_folder>
```

Improved plaintext modelling

The training set that we have in this question is very small. Maybe we can further improve performance by having a better model of character bigram transitions in English. Change your training procedure to incorporate this information by preprocessing and getting character transition counts from samples of English that you collect from webpages online. These counts should supplement, not replace the counts that you get from the original training set. You will have to deal with the following issues:

1. Sentence segmentation
2. Lower-casing the text
3. Removing any other character which is not one of the 29 symbols
4. Removing extra space from the beginning and end of each sentence

How does this change affect the results?

It should be possible to turn this option on at the command line in the following way:

```
python decipher.py -lm <cipher_folder>
```

For this step, you should save and submit the additional preprocessed corpus or your counts thereof, so that the TA can reproduce your results. In addition, it should be possible to turn on both Laplace smoothing and the improved language modelling.

Report and submission requirements

Experiment on the three ciphers, reporting accuracy for each of the settings in a table. Write a brief (max. half-page) report on the results, noting whether each change was successful in improving performance. Were there any surprises or unexpected results? Do you achieve close to perfect accuracy? If not, why not? Try to explain and speculate on the reasons for these results.

What To Submit

Your submission must be made through MyCourses, and must consist of the following three files:

1. The written portion of Questions 1, 2 (except the grammar), and 3 as a .pdf file called 'a2-written.pdf'.
2. The grammar file of Question 2, as a plaintext .txt file called 'french-grammar.txt'.
3. For the programming part of Question 3, you should submit your source code and any other files necessary to regenerate your results as described above as a single .zip file called 'a2-q3.zip'.