Homework 3
Due: Anytime on February 28, 2024
Submit: On Blackboard

Part 1: Hidden Markov Models

We are going to use a Hidden Markov Model to tag the part-of-speech of the words in the sentence:

Flies fly

In this setup, we have observable events, which are the observed words ("Flies fly"), which are generated by hidden events, which are the parts-of-speech. In order to build up our Hidden Markov Model, we need to define:

- A set of possible states (in our case, the set of possible part-of-speech tags). We will use the "Universal" POS tags: DET, PRON, NOUN, ADP, ADV, ., CONJ, VERB, PRT, ADJ, NUM, X. See SLP Figure 8.1 if interested in more details.
- A transition probability matrix that defines the probability of transitioning between each pair of states (in our case, P(pos_tag_of_word_w+1 | pos_tag_of_word_w); one example is P(VERB|NOUN), or the probability of a verb occurring right after a noun).
- Emission probabilities that define the probability of a particular state (part-of-speech) generating a particular observation (word). In our case, this takes the form of P(word|pos_tag); one example is P('fly'|VERB).
- An initial probability distribution over states (in our case, one example is the probability that a sentence starts with a NOUN). We will use the following initial probability distribution calculated from the tagged Brown corpus:

DET	PRON	NOUN	ADP	ADV	.	CONJ	VERB	PRT	ADJ	NUM	X
0.21	0.16	0.14	0.12	0.09	0.09	0.05	0.05	0.04	0.03	0.02	0.0

Generally, you would want to define all of these probabilities for all words in the vocabulary and all possible tags/states, but for this homework, you only need to worry about the words "flies" and "fly" and any tags that are observed with these words in the corpus (hint: as you will see, most POS tags are not observed with these words in this corpus).

For this problem, we will use the Brown corpus tagged using the universal tagset, which can be accessed using the python package nltk (info here and see attached code for starter code to access the corpus).

Important: For the purpose of this problem, please treat capital and lowercase letters identically - e.g., "Flies" and "flies" should be treated as the same word.

Answers to hand in:

1. In the Brown corpus with Universal tags, what is the set of part-of-speech tags we observe as labels for the words in our sentence: “flies” and “fly”?
2. Provide all non-zero emission probabilities for “flies” and “fly” based on the tagged Brown corpus. (Hint: refer to your answer to #1). Remember to ignore case (i.e., treat “Flies” and “flies” as the same word).
3. Give an example of an emission probability (for ‘flies’ or ‘fly’) that is zero.
4. Normally, you would want to calculate the transition probability matrix between all pairs of possible states (i.e., parts-of-speech). Given that there are 12 parts-of-speech in the universal tagset, this would involve creating a 12x12 matrix. However, for this problem, it is sufficient to create a 2x2 matrix. Provide that transition probability matrix and its values, and explain why no other transition probabilities are needed in order to correctly tag the sentence “Flies fly”.
5. Walk through the Viterbi algorithm to show how the sentence “Flies fly” would be tagged. Your answer can take the form of a 2x2 matrix where the rows correspond to the parts of speech that “flies” and “fly” can take (hint: refer to #1 and #2) and the columns correspond to each word in the to-be-tagged sentence.
6. Make a change to the emission probabilities, transition probabilities, or starting probabilities (or any combination thereof) in such a way that it changes how the sentence would be tagged (but make sure all probabilities that should sum to 1 still do). Provide your change and what the resulting part of speech tags would be. Explain the intuition for why your change to the probabilities causes a change in tagging.

Part 2: Parsing

In this part of the assignment, you will be working with the following context-free grammar.

Rules:

S → NP VP
S → Aux S
VP → V S
VP → V NP
VP → VP PP
NP → Det N
NP → NP PP
PP → P NP
PP → P S

Lexicon:

NP → Waikiki
NP → Oslo
NP → Kim
NP → snow
V → adores
V → snores
Aux → does
Aux → can
Aux → is
P → in
P → on
P → before
Det → this
Det → these
Det → the

Step through the pseudocode for the CKY algorithm to see how it would parse the following sentence given the above grammar:

Kim adores snow in Oslo.

What to hand in:

1. A diagram showing the final state of the chart
2. A list of changes/additions made to the chart in the order in which they were added. Each entry in this list should indicate the rule that licenses the addition and its span.
3. Is this sentence ambiguous? If so, what are the interpretations corresponding to the different parses?
4. We can turn the context-free grammar from above into a probabilistic context-free grammar by adding probabilities to each rule, which would allow us to calculate the probability of parse trees. Use the following probabilistic context-free grammar to calculate the probability of all output parses from (1). What does your answer tell you about the most likely meaning of the sentence?

Rules:

S → NP VP	0.8
S → Aux S	0.2
VP → V S	0.25
VP → V NP	0.45
VP → VP PP	0.3
NP → Det N	0.4
NP → NP PP	0.2
PP → P NP	0.9
PP → P S	0.1

Lexicon:

NP → Waikiki	0.1
NP → Oslo	0.1
NP → Kim	0.1
NP → snow	0.1
V → adores	0.7
V → snores	0.3
Aux → does	0.2
Aux → can	0.25
Aux → is	0.55
P → in	0.2
P → on	0.4
P → before	0.4
Det → this	0.2
Det → these	0.2
Det → the	0.6