CS 1.2: Intro to Data Structures & Algorithms

Histogram & Markov Chain Worksheet

Name: Mark Frazier

Text: *"I like dogs and you like dogs. I like cats but you hate cats."* (ignore all punctuation)

Histograms

Q1: How many distinct word *types* are present in this input text? How many total word *tokens*?

Distinct word types: 8 Total word tokens: 14

Q2: What data structure would be appropriate to store a histogram counting *word frequency*? Why did you choose this data structure? In other words, what makes this data structure ideal?

A dictionary or hashmap. I chose this data structure because it is very efficient allowing for quick retrieval of data. As the hashmap grows, the lookup time remains constant.

Q3: Write the data structure you would create to store this histogram counting *word frequency* (as it would look if you printed it out with Python).

word_histogram = { "I": 2, "like": 3, "dogs": 2, "and": 1, "you": 2, "cats": 2, "but": 1, "hate": 1}

Markov Chains

Q4: Draw a conceptual diagram of the *Markov chain* generated from analyzing the text above. Label each *state transition arc* with the count of how many times you observed that word pair.

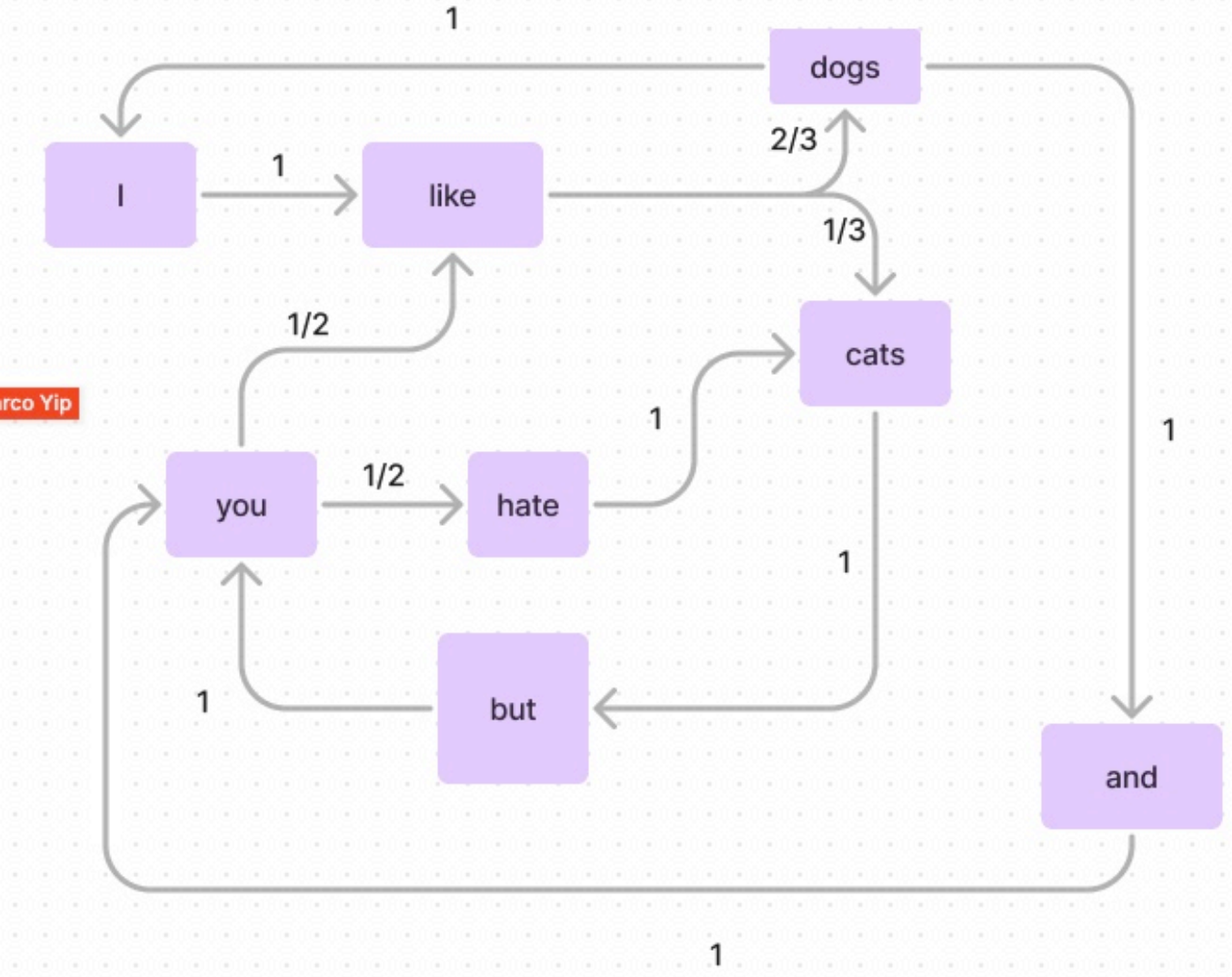
(On the next Page.)

Q5: Write the data structure you would create to store the word transitions out of the state that represents the word "*like*" in this Markov chain (as it would look if you printed it out with Python).

like_transitions = {'like' : ['dogs', 'cats']}

Q6: Write a *new sentence* that can be *generated* by doing a *random walk* on this Markov chain.

I like cats and you like cats but you hate dogs.


1
dogs
I
1
like
2/3
1/3
1/2
cats
1
1
you
1/2
hate
1
1
but
and
1