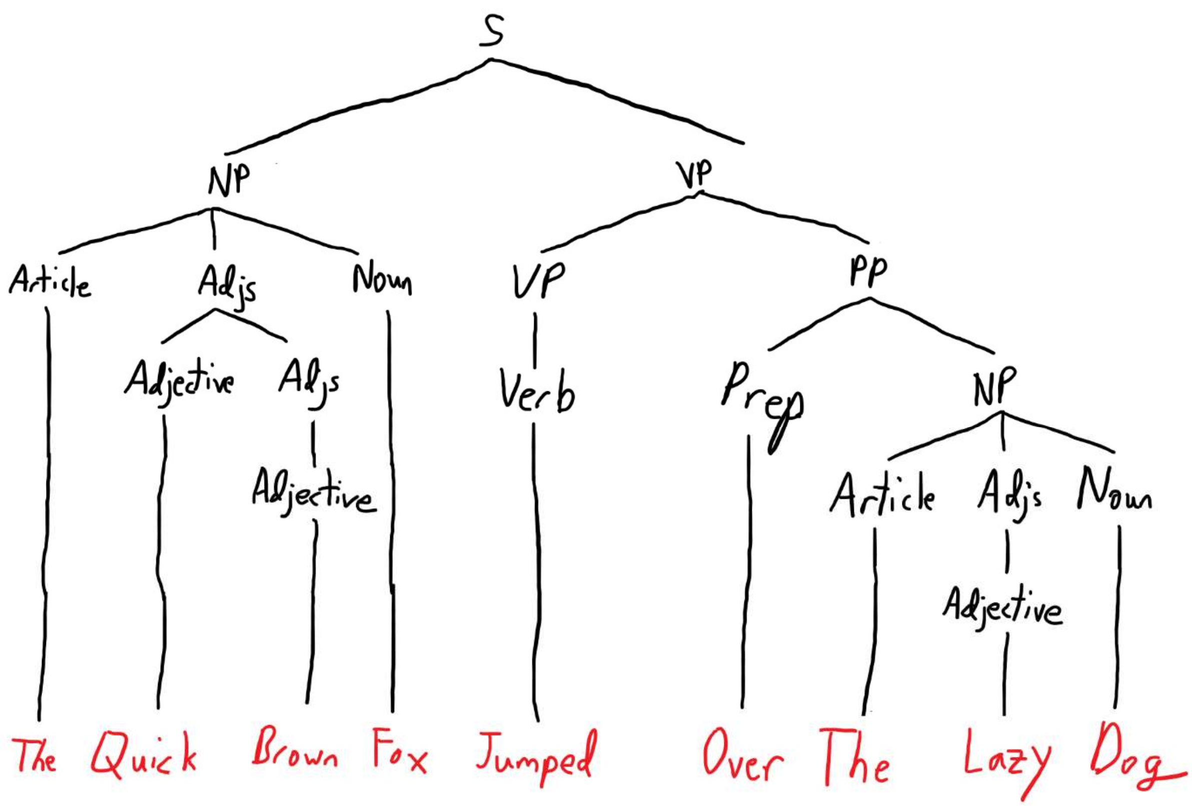
ECE 469: Artificial Intelligence
Fall 2020
Problem Set #3

1) Computational Linguistics

Consider the famous sentence, "The quick brown fox jumps over the lazy dog."

Draw a reasonable parse tree for the sentence (assuming the existence of reasonable grammar rules). The root of the tree should be S, representing a sentence, and the leaves should be the words of the sentence.

Also express the CFG rules, including the lexical rules, that are implied by the tree.



S	→	NP VP
NP	→	Article Adjs Noun
Article	→	The
Noun	→	Fox
	|	Dog
Adjs	→	Adjective Adjs
	|	Adjective

ECE 469: Artificial Intelligence

Fall 2020

Problem Set #3

Adjective	→	Quick
	|	Brown
	|	Lazy
VP	→	VP PP
	|	Verb
Verb	→	Jumped
PP	→	Prep NP
Prep	→	Over

S	
NP VP	S → NP VP
Article Adjs Noun VP	NP → Article Adjs Noun
Article Adjective Adjs Noun VP	Adjs → Adjective Adjs
Article Adjective Adjective Noun VP	Adjs → Adjective
The Adjective Adjective Noun VP	Article → **The**
The Quick Adjective Noun VP	Adjective → **Quick**
The Quick Brown Noun VP	Adjective → **Brown**
The Quick Brown Fox VP	Noun → **Fox**
The Quick Brown Fox VP PP	VP → VP PP
The Quick Brown Fox VP Prep NP	PP → Prep NP
The Quick Brown Fox Verb Prep NP	VP → Verb
The Quick Brown Fox Jumped Prep NP	Verb → **Jumped**
The Quick Brown Fox Jumped Over NP	Prep → **Over**
The Quick Brown Fox Jumped Over Article Adjs Noun	NP → Article Adjs Noun
The Quick Brown Fox Jumped Over Article Adjective Noun	Adjs → Adjective
The Quick Brown Fox Jumped Over The Adjective Noun	Article → **The**
The Quick Brown Fox Jumped Over The Lazy Noun	Adjective → **Lazy**
The Quick Brown Fox Jumped Over The Lazy Dog	Noun → **Dog**

ECE 469: Artificial Intelligence
Fall 2020
Problem Set #3

2) Statistical NLP (conventional and modern)

Briefly answer the following questions (with *one or two phrases or sentences for each question*) related to statistical natural language processing.

(a) Naïve Bayes systems work well for some text categorization tasks, even though the "naïve" assumption is clearly false. Explain what it means for the assumption to be false for this task, and give a specific example that demonstrates it is false.

Naïve Bayes assumes that each word occurs independently from every other word, which is clearly false. The order of the words does not matter. For example, a review that states, “That movie was incredibly bad.” is clearly negative, but it has one positive word (incredibly) and one negative word (bad), so a Naïve Bayes system may not classify this correctly as a negative review.

(b) Consider a conventional, feedforward neural network applied to the task of text categorization, and one sentence is being classified at a time. Assume it has been trained on a corpus with D labeled sentences, and the total size of the vocabulary is V. It is now being used to classify a document with T total tokens and U unique, or distinct, tokens. If a conventional feedforward neural network is being used for the task, what would typically be the number of input nodes? What would be represented by each input node?

There are V input nodes. Each input node represents a target word.

(c) Now consider text categorization involving d-dimensional word embeddings and a recurrent neural network (either a simple RNN, or a variation such as an LSTM). We have learned that it shouldn't be necessary to pad sentences to ensure they have equal length. When using other types of deep neural networks with word embeddings (such as a feedforward neural networks or a CNN), it typically is necessary to pad the input sentence. Why isn't it generally necessary to pad sentences when using an RNN for text categorization?

RNNs can take inputs sequentially, unlike other neural networks. An RNN can take a sentence word-by-word as an input, rather than having to take the entire sentence as an input. Other neural networks have a fixed input size, so it is necessary to pad a sentence to that fixed size.

(d) Now consider a hidden Markov model being used for part-of-speech (POS) tagging. If the tagger is trained using a treebank (a corpus containing labeled examples of POS), what parameters need to be learned?

It needs to learn the transition probabilities and the emission probabilities.

(e) Now consider a simple RNN being used as a POS tagger (in practice, a variation such as an LSTM would more likely be used). If the tagger is trained using a treebank, what parameters need to be learned?

The weights of the RNN are learned, which includes the weights for the next state (like the transition probabilities) and the weights for the output (like the emission probabilities).