ECE469 – Pset 3

Jonathan Lam

December 14, 2020

1. *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.*

   The sentence can be represented with the following parse tree:

```
[S
  [NP
    [Article The]
    [Adjs
      [Adjective quick]
      [Adjs [Adjective brown]]]
    [Noun fox]]
  [VP
    [VP [Verb jumps]]
    [PP
      [Prep over]
      [NP
        [Article the]
        [Adjs [Adjective lazy]]
        [Noun dog]]]]]
```

   (NB: this parse tree is valid Lisp (Chez Scheme) syntax (except perhaps that the word literals should be quoted)! Shows how well Lisp works with grammars.)

This uses the CFG given in the class example, with a different lexicon. The subset of the rules used in this example is summarized in the table below.

S	$\rightarrow$ NP VP | ...
NP	$\rightarrow$ Article Adjs Noun | ...
VP	$\rightarrow$ Verb | VP PP | ...
Adjs	$\rightarrow$ Adjective | Adjective Adjs
PP	$\rightarrow$ Prep NP
Article	$\rightarrow$ the | ...
Noun	$\rightarrow$ fox | dog | ...
Adjective	$\rightarrow$ quick | brown | lazy | ...
Verb	$\rightarrow$ jumps | ...
Prep	$\rightarrow$ over | ...

2. *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.*

   The naïve Bayes formulation for text categorization is as follows:

$$\mathrm{P}(\mathrm{C}|w_{1:N}) = \alpha \mathrm{P}(\mathrm{C}) \prod_{i=1}^{N} \mathrm{P}(w_i|\mathrm{C})$$

   The naïve assumption is that each word is conditionally independent of all of the other words given the category; this is false if two words are often seen together. E.g., the words "Volkswagon" and "beetle" should not be conditionally independent of the category "vehicle": $\mathrm{P}(\text{beetle}|\text{vehicle})$ is probably much lower than $\mathrm{P}(\text{beetle}|\text{vehicle}, \text{Volkswagon})$, which will make the product of probabilities lower than it should be.

3. *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?*

   Without word embeddings, there would likely be V input nodes, where each input node represents the presence or frequency of a single word in the vocabulary. (D is irrelevant because one sentence is being classified at a time, and limiting the number of input nodes to T and U would not allow the network to use the whole vocabulary.)

4. *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?*

   A FF NN makes its inference based only on its inputs (it has no sense of persistent state or context), so it requires sentence context from the surrounding words (the rest of a sentence); since they usually have a fixed number of inputs, they require a padded input. A RNN does have an internal state that persists over a series of inputs, so it can handle dynamic-length sentences, and only requires one input at a time.

5. *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?*

   In a HMM using POS tagging, the current (hidden) state represents the part of speech. Thus, we need to learn two parameters: the transitions ("transition probabilities") between different states (the next likely part of speech), as well as the most likely word given the current POS ("emission probability").

6. *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 RNN's internal state is a function of the current input word's embedding and the previous state. It has to learn the function (weights) to transform the current state to the next state (similar to the transition probability), as well as the function (weights) to estimate the POS given the current state (similar to the emission probability).