Sample Questions

Multiple Choice

Circle all that apply.

1. Which of the following are true about a valid dependency tree?
a. There must be a path from 0 to every node in the tree
b. Every node must have at least one parent
c. Each arc is labeled with a word
d. Nodes must not form a cycle.

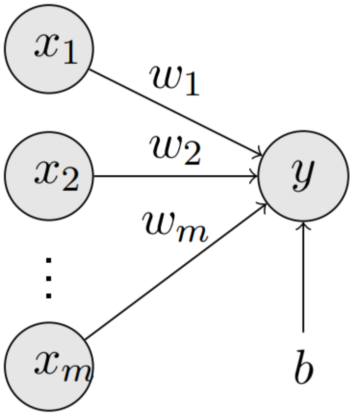
Answer: A, C, D. B is not true because the root node cannot have a parent. A and D are true by the definition of a tree. C is true in that arcs have labels such as “NSUBJ”.

2. In a fully-connected, feed-forward neural network
a. Some neuron in layer i is connected to every neuron in layer i+1
b. Each neuron in layer i is connected to some neuron in layer i+1
c. Each neuron in layer i is connected to every neuron in layer i+1
d. Some neuron in layer i is connected to some neuron in layer i+1

Answer: C is most correct. The other options are not complete.

3. A single layer perceptron
a. has many input neurons but only one output neuron
b. has one hidden layer
c. has no hidden layers
d. can only learn functions where the input is linearly separable

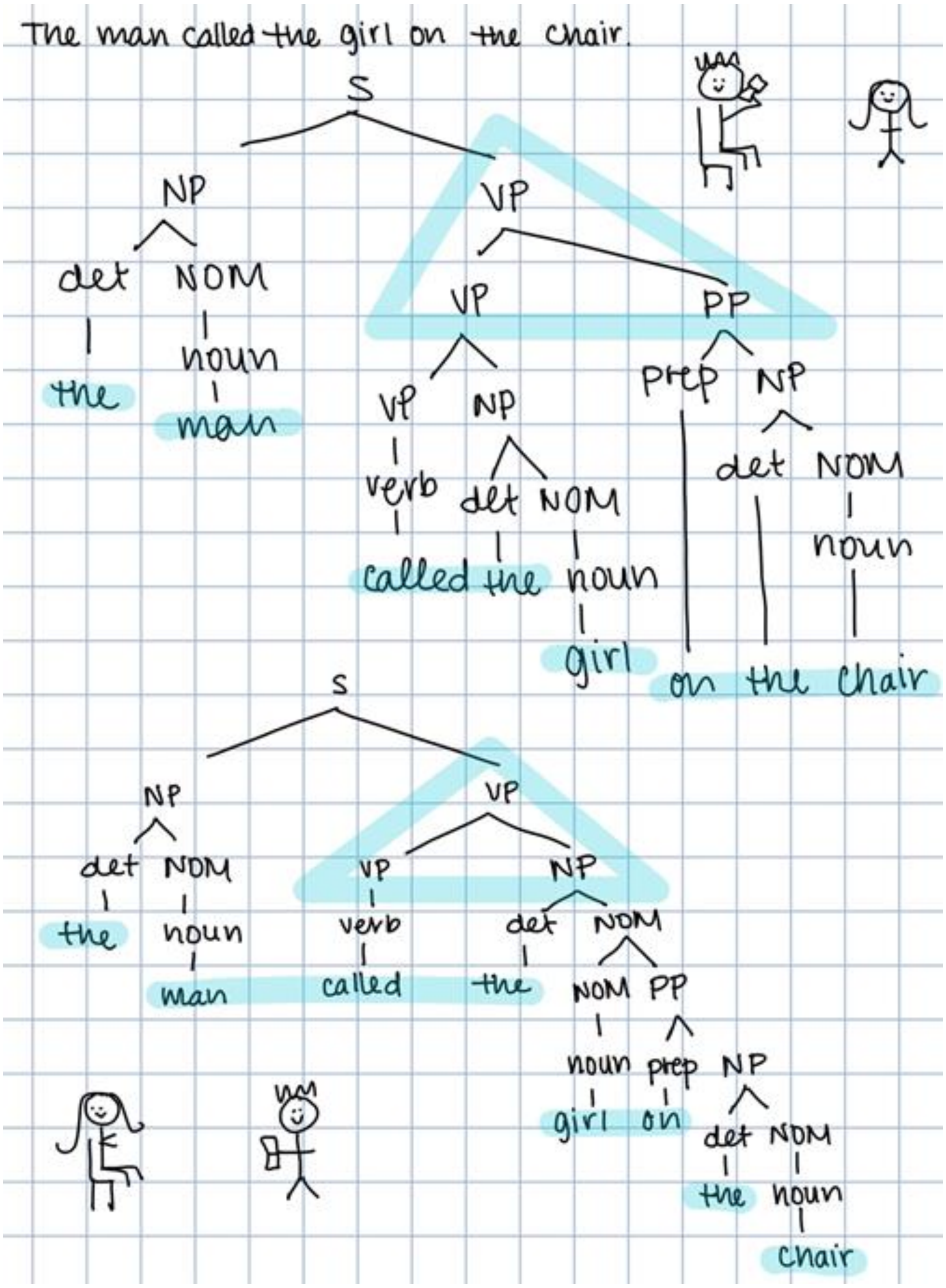
Answer: A, C, D. A single-layer perceptron is pictured below. Recall that a single-layer perceptron is essentially a linear classifier and cannot solve problems like the XOR problem studied in class.



4.True or False: The grammar shown below is ambiguous for the sentence "The man called the girl on a chair."

S -> NP VP
NP -> Det NOM
NOM -> noun
NOM -> NOM PP
VP -> VP PP
VP -> verb
VP -> VP NP
PP -> prep NP

Answer: True.
"On a chair" can modify the VP (rule 5) or can modify the NOM (rule 4), so two different parse trees can be constructed for this input. (That is, either the man is doing the calling while on a chair, or the girl is on a chair.)



5. A good source of seed patterns to bootstrap information extraction for biographies is:

a. Google search results
b. Manual patterns
c. Wikipedia entries for people
d. WordNet

Answer: C. Wikipedia has an information box that lists important attributes about a person, giving structured data with which to start to collect patterns. Manual patterns would be highly precise, but would be expensive to create.

6. Which of the following word pairs is an example of a hypernym?

a. *animal* is a hypernym of *dog*
b. *black* is a hypernym of *white*
c. *big* is a hypernym of *large*
d. *bank* is a hypernym of *building*

Answer: A.

Short Answer

1. Provide two sentences that could be generated by the following AMR

```
(r / read-01
            arg0 (b1 /boy)
            arg1 (b2/ book
              :poss (a /amr-unknown))
```

Answers: There are multiple correct answers, including

- Whose book did the boy read?
- The boy read which book?

2. Name two sources of disambiguation for POS tagging and describe how they are used in hidden markov modeling.

Answers: (Note these are sources that come from the data, which would be other parts of the sentence surrounding the word to be tagged)

- current word lemma (emission probabilities)
- POS of the previous word (transition probabilities)

The probabilities of the usual pos for this word play a role in selecting the current POS. The current tag also depends on the tag of the previous word.

3. How many gates does an LSTM have? State what they are and what each gate does (one sentence/gate).

Three. Input, forget, output. For each gate the LSTM learns which elements of the vector are most important and should be passed on. The input gate learns which vector elements of the input word are most important. The forget gate learns which elements of the previous context vector are most important. The output gate learns which elements of the current state are most important.

4. State two different kinds of information that would be useful for learning a dependency parser and explain why (1 sentence each).

- POS tags. POS tags of the current word help determine what dependency it can participate in.
- Previous arcs (for the sentence parsed so far). If the parser has so far learned arcs representing a subject and a verb, then it is likely the next arc may point to an object (for example).
- Words to the left and the word on the buffer. Words to the left of the current word can indicate what arcs should be added between those words and the current word. The next word on the buffer can indicate what right arc should be added to the parse.
- Lemmas (in place of words). Sometimes lemmas are more informative than words.

5. What is the difference between distributional representations and distributed representations?

Distributional representations can include sparse vectors. They are derived from the distributional hypothesis and encode some information about the frequency of the words in the context of the word being expressed.

Distributed representations distribute meaning across the vector but as a consequence are typically not interpretable. Words with similar meanings or contexts will share some of their distributed representation. A typical word embedding is an example of a distributed representation.

Word2vec embeddings are both distributional and distributed.

6. State two ways in which BERT is different from Word2Vec.

Possible answers include

1. BERT can provide contextual embeddings for each word, e.g. the representation for “bark” would be different for the sentences “i love tree bark” and “i love to bark like a dog”.
2. BERT is trained using a transformer architecture, whereas word2vec is trained with a more standard dense neural network.
3. BERT creates representations at the subword level, whereas word2vec creates them for whole words.

4. BERT can be used as a full language model, whereas word2vec only creates word embeddings.

6. In the following text, circle the mentions. What does the term *wikification* mean and why is it useful to develop a program that does this?

[Donald John Trump] (born June 14, 1946) is the 45th and current [President of the United States], in office since January 20, 2017. Before entering politics, [he] was a businessman and television personality. [Trump] was born in the [New York City] borough of [Queens]. [He] earned an economics degree from the [Wharton School] of the [University of Pennsylvania] and followed his [grandmother Elizabeth] and [father Fred] in running the [family real estate company].

Mentions are *words which refer to entities in the real world.*
Wikification means *to link mentions with the Wikipedia pages (or entries in some other knowledge base) on their respective entities.*

Wikification is useful in information extraction systems, summarization systems, and any systems in which you or your users may want to know information about the entities which is not contained explicitly in the text. For example, linking mentions of Paris the city to “Paris, France” and Paris the American media personality to “Paris Hilton” may help you decide what documents are related in information extraction/retrieval.

7. A phrase-based MT system uses a language model, a phrase-based model and a distortion model. Give a one sentence definition of each in the context of MT, referring to how they are used.

A language model is a model of fluent word order in the target language. It is used to help arrange the words of the translation.
A phrase based model represents the statistics for translating phrases in the source to phrases in the target.
A distortion model represents the likelihood that the word order of the output will be different from the input and how.

8. What is the vanishing gradient problem and when does it arise in neural networks? What are two techniques that can be used to combat it? (Your answers can apply to any kind of network you want, e.g., a recurrent network.)

The vanishing gradient problem has to do with training neural networks with backpropagation. The updates for weights can get very small (or even 0), effectively preventing learning because weights are never changed. The update for a weight in the network is proportional to the partial derivative of the error function; because of the chain rule if there are many layers or many steps between the output and the weight, this update can get very small for the earlier steps (multiplying many very small values together), and may literally turn to 0 in a computer representation (underflow).

Multiple possible answers.) To combat this:

1. Use a different activation function, like ReLU or another rectifier that doesn't squish gradient values to be close to zero.
2. Use an LSTM or GRU that allows information from earlier steps to be added together, rather than multiplied.
3. Use a resnet, that adds residual connections directly to earlier layers.

Problem solving

3. [20 points]. Summarization. Suppose you want to develop a machine learning approach to summarization that extracts phrases rather than full sentences and puts together the phrases to form a sentence for the summary.

a. You need to create training data that your learner can use. You have access to the Zipf-Davis corpus, which consists of news articles associated with a human abstract.

1. Describe how you might use a parser and what code you would have to write to access phrases below the sentence level.

   Answer: You would use a parser to parse the sentences of the input article. You would need to write code to find valid sub-phrases within the parse tree. For training, you would use this to find phrases in the input text that match phrases within the gold standard summary. These phrases would be selected (e.g., assigned a value of 1). Then, all phrases that do not match the phrases of the summary would get the value 0 (i.e., not be selected).

2. Describe how the following methods of determining when an extracted clause from the summary matches a clause from the article would result in different kinds of abstraction:

   a. Using word overlap only

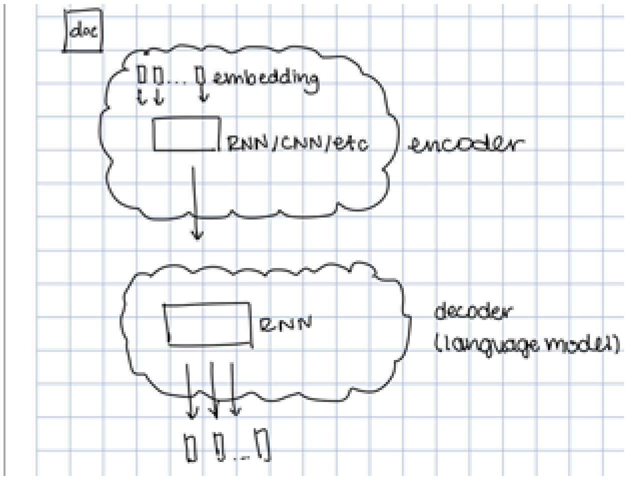
      Answer: If you could only use word overlap then you'd be developing a summarizer that does compression.

   b. Using WordNet also

      Answer: This would allow you to generate summaries with paraphrases and fusion (since fusion would typically involve some paraphrasing when putting together clauses from different sentences.)

3. Many summarization systems use language models. Explain how they are used as part of a neural net summarization system. Provide a diagram of the summarization system architecture.

A summarization system should produce text that is fluent—it reads like it was written by a human writer. It should be grammatically correct and flow well. A language model, which predicts the most likely next word given past words (like bigrams or trigrams) can help produce fluent text. The neural architecture for a summarization system would use a sequence-to-sequence or encoder-decoder model. The input to the encoder would be the document to be summarized. It might consist of an embedding layer that is fed into an RNN, CNN, bag-of-words, etc. The output of the encoder would be given to the decoder, which is a language model that generates the final output (the summary).



3. [14 points]. Information Extraction and bootstrapping. Suppose you are building an information extraction system to identify the city and state in which a person was born. You want to use bootstrapping to do this.

a. [8 points] You know where Barack Obama was born (Honolulu, Hawaii) but you don't know where any other famous person was born. Describe how you could use this information, along with a combination of data from Google and Wikipedia, to find patterns that could be used in general to determine place of birth. Be specific. Show the algorithm and some examples of the algorithm in operation (you can make up data that you think would be available).

You might first look at Barack Obama's Wikipedia page, or search for documents on the web, to find phrases in which the entities "Barack Obama" and "Honolulu, Hawaii" both occur. (For example, on Obama's Wikipedia page, you can find "Obama was born in Honolulu, Hawaii" and "Obama was born on August 4, 1961, at Kapiolani Medical Center for Women and Children in Honolulu, Hawaii."

Then you might remove the known entities and transform those phrases into patterns. (For example, "<PERSON> was both in <PLACE>" or "<PERSON> was born on <DATE>, at <PLACE2> in <PLACE>".)

Next, you might search for instances of those same templates in documents on the web. (Perhaps you find something like "George W. Bush was born in New Haven, Connecticut" or "Biden was born on November 20, 1942, at St. Mary's Hospital in Scranton, Pennsylvania".)

Then, by matching entities to slots in your templates, you can gain new information. (In this example, now you know the birthplaces of both George W. Bush and (Joe) Biden.)

b. [4 points] Why is it better to use both Google and Wikipedia rather than one corpus alone? What advantage does each corpus have?

The two corpora contain different types of information and are able to complement each other. Wikipedia is full of reasonably clean, structured data with convenient links to other articles and entities. It is formal and often consistent, making it easy to find patterns. Meanwhile, Google searches are far broader in scope and contain information about many more entities, but often in a less structured and noisier way.

Questions from In-Class Review

What are three problems for which we use a sequence-to-sequence decoder?

1. machine translation
2. dialog generation
3. generating description from meaning representation

What is the problem that attention is trying to address?

Allowing the decoder to focus on earlier parts of the input.

How does attention work?

Attention develops a filtered context vector, so the decoder can access the hidden states of any of the encoder steps.

What is the difference between abstractive and extractive summarization?

Extractive pulls text verbatim from source; abstractive can paraphrase.

What are three different encoder models we have encountered?

1. bag of words
2. convolutional encoder
3. attention

In the context of word sense disambiguation, what are collocational features?

A vector consisting of the token and pos for words within a window of the target word (i.e. the one you're trying to disambiguate).

Given a set of reference summaries and a generated summary, how do you calculate the pyramid score?

Select important Summary Content Units (SCUs) and build a 'pyramid' of the SCUs based on how many of the reference summaries they occur in. e.g. if an SCU occurs in 4 summaries, then it has weight 4 and would be 'high up' in the pyramid. (Low weighted SCUs are on the bottom of the pyramid, bc there are more of them.) Select the SCUs that would be in a "maximally informative summary" and calculate the score by summing the weights of the SCUs (this is MAX). Then sum the weights of the SCUs in the generated summary (this is D). The pyramid score is D/MAX.