Final Exam

COMS W4705: Natural Language Processing

December 9, 2019

Version A

Directions

This exam is closed book and closed notes. You may use a non-graphing calculator. It consists of three parts, each labeled with the amount of time you should expect to spend on it. If you are spending too much time on a question, skip it and come back if you have time.

The first part is multiple choice. The second part is short answer **(select four of six)**. The third part is problem solving. **Read the instructions at the top of each section carefully before beginning to work on the problems.**

Important: Answer Part I by **circling the letters** of your answers. Answer Part II and Part III in the provided spaces. **For Part II, circle the numbers of the four problems you select. If you do not select four, we will grade the first four you attempt.** Turn in all test sheets at the end of the exam. If you need extra paper, write CONTINUED at the end of the space and label each extra sheet with the corresponding question number.

VERY IMPORTANT: If you are still writing by the time we come to collect your exam, we will mark your paper as late and you will receive a deduction.

Name: _____ UNI: _____

UNI of the person to your left: _____

UNI of the person to your right: _____

☐ **Check this box if you are a PhD student taking this exam as a comp**

1 Multiple choice (30 points total)

Circle the letter of the choice you select.
For all questions you should select exactly one choice.

Recommended time: 20 minutes.

1. (3 points) In an encoder-decoder sequence to sequence model, like that used in homework 4, what are the two inputs for the *first* decoder step?

A. The first word of the gold standard output and the last hidden state of the encoder.

B. The start-of-sequence token and the last hidden state of the encoder.

C. The first word of the gold standard output and a randomized initial hidden state.

D. The start-of-sequence token and a randomized initial hidden state.

Answer: B [A in other version]

2. (3 points) Which of the following describes the attention mechanism as used in sequence to sequence models?

A. Attention introduces a dynamic context vector for each *decoder step*; this context vector is a weighted average over all encoder hidden states.

B. Attention introduces a dynamic context vector during each *encoding step*, which allows the encoder to look both forward and backwards while encoding.

C. Attention introduces a dynamic context vector for each *decoder step*; this context vector is a weighted average over all past decoder hidden states, allowing it to look backwards at past decoding steps.

D. Attention introduces a dynamic context vector for each *encoding step*; this context vector is a weighted average over all the past encoder steps.

Answer: A [B in other version]

3. (3 points) Which of the following is an example of a hyponym relation?

A. "tired" is a hyponym of "weary".

B. "bank" is a hyponym of "bank".

C. "algebra" is a hyponym of "math".

D. "meal" is a hyponym of "lunch".

Answer: C [in both versions]

4. (3 points) Given the following training sentences, estimate the probability of the sequence <SOS> Bella is cute <EOS>. Assume you are using a bigram language model, and sentence boundaries occur at the line breaks.

Begin training data:

<SOS> The dog is cute <EOS>

<SOS> Bella is her name <EOS>

<SOS> I love Bella <EOS>

End training data.

A. $\frac{1}{36}$
B. $\frac{1}{12}$
C. $\frac{1}{8}$
D. $\frac{1}{3}$

Answer: B [C in other version]

5. (3 points) Which of the following is true about summarization?

A. Abstractive summarization is a type of extractive summarization.
B. Extractive summarization reuses sentences from the source to describe it.
C. Abstractive summarization chooses salient sentences from a source to describe it.
D. Abstractive and extractive summarization are similar, but research on extractive summarization is more recent and at a less mature stage.

Answer: B [A in other version]

6. (3 points) In a bigram Hidden Markov Model POS tagger using the Viterbi algorithm, how is the final POS sequence chosen?

A. Select the most likely POS for each individual word conditioned on the word and the previous two POS tags.
B. Select the most likely POS for each individual word independently.
C. Select the most likely POS for each individual word conditioned on the word and the previous POS tag.
D. Select the most likely POS for each individual word conditioned on the word.

Answer: C [A in other version]

7. (3 points) Which problem with using recurrent neural networks to process text is the Long Short-Term Memory Network usually intended to fix?

A. The RNN architecture tends to overfit to the training data because of a lack of structure.

B. The RNN tends to forget information and dependencies over long distances.

C. The RNN can become too reliant on certain weights if they are not sometimes zeroed out during training.

D. The RNN is unable to learn to make use of subword information because it does not place more importance on character clusters that occur frequently.

Answer: B [D in other version]

8. (3 points) Which of the following is a drawback of WordNet as an inventory of word senses for all-words word sense disambiguation?

A. WordNet is noisy because it is semi-supervised.

B. Synonyms do not fully disambiguate between word senses, so WordNet does not contain sufficient information for this task.

C. WordNet senses are very fine-grained, which makes them difficult to annotate as well as to classify.

D. WordNet's coverage is very limited, so most words are out-of-vocabulary.

Answer: C [B in other version]

9. (3 points) Suppose you have the following candidate sentence and reference sentences in a language generation setting. What is the modified **bigram** precision of the candidate sentence as defined in BLEU? (Do not consider start and end tokens.)
Candidate sentence:

```
my cat chased the lizard
```

Reference sentences:

```
the cat chased a mouse
my cat ran after a lizard
the dog chased the frog
```

A. $\frac{5}{6}$

B. $\frac{3}{4}$

C. 1

D. $\frac{9}{5}$

Answer: B [D in other version]

10. (3 points) Which sentence could not be handled by the grammar below:

```
S-> NP VP
NP -> det N
VP -> verb NP
VP -> verb NP NP
```

A. The dog ate dinner.

B. The chickens escaped the coop.

C. The cat chased the dog.

D. The man gave the dog a bone.

Answer: A [D in other version]

2 Short answer (32 points total)

Provide at most 2 or 3 sentences for four out of the following six questions. Answer in the provided space after the question. If you need extra space, continue on another sheet of paper clearly labeled with the problem number.

NOTE: Choose FOUR. If you answer more than four, you will be graded on the first four answers you provide.

Recommended time: 25 minutes.

1. (8 points) What are two problems with BLEU? Why do researchers keep using it despite the problems?

Answer: BLEU has no explicit model of semantics. This means it cannot give high scores to out-puts that are semantically correct but don't word-for-word match something in a reference sentence. It tends to favor fluent outputs over correct ones, as a mixture of reference sentences will score highly even if this mixture introduces unexpected meanings.

Penalizes synonyms, paraphrasing, or inflectional variations/morphology.

Allows changes in critical words that can completely change the meaning of a sentence. BLEU treats all words equally and does not prioritize content or topic words.

It tends to over-estimate performance by prioritizing fluency.

As you collect more and more reference sentences, BLEU may begin overestimating the quality of your candidates.

The brevity penalty may not appropriately select a good length; it just restricts the model to choosing outputs of the same length as the references.

People keep using BLEU because it is automatic and thus can be used for development very easily over and over again. It also makes it easy to compare against the state of the art since everyone uses it.

2. (8 points) An extractive summarizer for single-document news summarization that produces a paragraph length summary is naturally implemented by an encoder-decoder architecture. What does the encoder encode? Describe in 1-3 sentences how it could do this. What does the decoder implement?

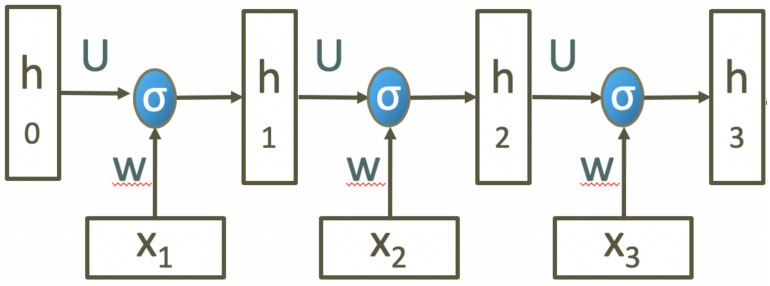
Answer: The encoder encodes the full document which must be summarized. Because this is very long, we saw how a hierarchical architecture could be used. First a CNN encoded each sentence of the input and then the sentence embeddings were passed to an LSTM which encoded the full document. The decoder used attention to look back at corresponding hidden states. For each sentence, the decoder implements a classifier that decides to keep the sentence in the summary or not.

3. (8 points) Psychological research has used the implicit association test to measure bias. In this test, the difference in mean response time is measured between two sets of target words and two sets of attribute words. For example, the targets might be "insect" and "flower" and the attribute words might be words that all imply "unpleasant" and words that all imply "pleasant".

How has this test been adapted for word embeddings? How would the modified test be used to determine gender bias as it pertains to occupation?

Answer: For word embeddings, researchers developed a test called WEAT which measures the association between a set of target words and a set of association words. It does this using cosine similarity between each word in the target set and each word in the association set and then uses a permutation statistics test for significance; if some target words are significantly more associated with some target words (e.g., arts words are predominantly female and science words predominantly male), then the test suggests the embeddings are biased. To look at gender bias as it pertains to occupation, it could be done either by having the association set contain all words related to occupation or it could actually check association with Labor Bureau statistics which show how many people of each gender are employed for each job type.

4. (8 points) Consider the RNN shown below. What is U and how are its values computed? Describe how the hidden state h_t would be calculated at time t.



Answer: U is a weight matrix and its values are computed during training using back-propagation. The hidden state at time t is calculated by taking the sigmoid of U from time t-1 and the word embedding matrix W for the input word at time t-1.

5. (8 points) Neural MT is often augmented with attention and sub-word encoding. Why is attention important for neural machine translation? What information does attention capture that is similar to information induced during phrase-based MT? Why does sub-word encoding help neural MT?
Answer: Attention is important for NMT because it encourages the neural model to align the words of the input source sentence with the words of the target sentence, thus paying attention to the aligned word or phrase when translating to target words/phrases. Attention essentially captures the phrase table that is learned from a specific alignment process for phrase-based MT. Neural MT requires a lot of parallel data to train and due to sparsity issues, some words may not be seen frequently enough in the training data to provide a reliable signal for translation. However, if we break words down into sub-units, we will increase the amount of times each unit is seen. A typical sub-unit that is used is byte-pair encoding, but one can imagine using other sub-units as well.

6. (8 points) Show the first three actions an arc-standard transition-based dependency parser would make in parsing the following sentence. Assume you begin with the root already on the stack. Show the arc set that would be created by these actions.
Two points for each action and two points for a correct arc set.
`Kathy gave Roscoe a bath .` **Answer:**

```
Stack: root
Action 1: shift
Stack: root kathy
Action 2: shift
Stack: root kathy gave
Action 3: left arc
Arc set: ((Kathy gave subj))
```

3 Problem solving (38 points)

There are two problems in this section. Do both problems. Answer in the provided space after the question. If you need extra space, continue on another sheet of paper clearly labeled with the problem number.

Recommended time: 30 minutes.

1. (19 points) **Word sense disambiguation.** The preposition "of" is semantically ambiguous and, in fact, has many different possible meanings. Consider the following noun phrases:

1. I own *a house of red brick*.
2. I ate *the whole bowl of cherries*.

(a) (4 points) Each of the italicized noun phrases above has a different meaning of "of". Describe the meaning that is intended by providing a paraphrase of each italicized phrase.
Answer: 1. A house made of red brick 2. A bowl full of/containing cherries.

(b) (8 points) We discussed several different supervised methods to learn a word sense disambiguation program: one used collocational features, and the other used decision lists with ranked learned rules.

i. (4 points) Show the collocational features that would be represented for example 1 above assuming a window size of two.
Answer: Collocational features include the word, its position and its POS. So for example 1:
house noun made verb red adj brick noun
Where position is implicit.

ii. (4 points) Give three examples of learned rules that could be used for disambiguating "of" and describe in one sentence how those rules might be learned.
Answer:

```
if cherries is to right of "of" -> contains
if brick is to right of "of" -> made of
if hypernym of word to left of "of" is "container -> contains
if "house is to left of "of" -> made of
```

The rules can be learned by first creating an exhaustive list of candidates using some corpus or word list ("cherries" to the right, "berries" to the right, "dog" to the right, "dog" to the left...) and then selecting the best rules using a metric that calculates how discriminating the rule is. One such metric is the ratio of the P(sense1|rule1)/P(sense2|rule2).

(c) (7 points) If you were to use a neural net approach to do word sense disambiguation, would it be better to use Word2Vec or BERT? Justify your response in two sentences by saying why or why not for each. Describe how you could use the embedding space to do word sense disambiguation.
Answer: It would be better to use BERT. Word2Vec generates a single embedding for a word based on all contexts in which it is used. Thus, the embedding is in some ways an average of all senses. BERT produces contextual embeddings, yielding a different representation for different types of context. Clustering using BERT shows that words with the same sense will appear within the same part of the embedding space. So in order to disambiguate a word, we will embed it and its containing sentence using BERT and see where the word's embedding falls. We may calculate similarity to synonyms of the target word, or to other embeddings of the target word with a known sense (e.g., embed "the dogs bark" and "tree bark" to get different approximate locations for "bark").

2. (19 points) **Dialogue systems and generation.** You are building a neural dialogue system that can chat about a presidential election. Given a question, your system must generate a response. Assume that you are given a training set of transcripts of recorded dialogues between voters about the upcoming election.

(a) (7 points) What kind of neural architecture would you use? Give two different approaches that have been used for dialog systems, where each takes different input. Describe why each approach would be useful.

Answer: It would be natural to use an encoder/decoder architecture where the encoder part encodes some representation of the input utterances or dialog and the decoder produces a response appropriate (hopefully) to the input. Different approaches have been tried with different kinds of input. The encoder could only encode the previous utterance, which would be the question or statement which the user just made. Alternatively, the encoder could in addition encode some portion of the dialog context, from the several utterances prior to the previous utterance to the entire preceding dialog. The latter would enable the dialog system to take into account what it already said to help avoid generating the same response twice or it could encourage the system to use the previous context to help in disambiguating the previous utterance. Or it could help the dialog system to infer speaker goals.

(b) (3 points) Name one problem that a neural chatbot faces and describe an approach that has been used to address it (2 sentences max).

Answer: Could be any of: 1. Having a consistent personality. Approach: learn personality embeddings.

2. Handling OOV entities: templatization or mechanisms that copy information like named entities from the input

3. Repetition or making responses that are too general: VAE with latent variable to learn different distributions.

(c) (9 points) Your system from part (a) receives as input "My biggest concern is honesty." To generate its response, it has access to the bigram probabilities shown in Table 1. Show how your answer generator would construct the first three words of the response using beam search with a beam size of two. Show each step of constructing the response and the paths the generator maintains.

	are	**you**	**concerned**	**I**	**want**	**a**
are	0	0.6	0.2	0	0.05	0.1
you	0.2	0	0.4	0	0.3	0.1
concerned	0	0.4	0	0.1	0	0.5
I	0	0.05	0.01	0	0.7	0.05
want	0	0.5	0	0	0	0.5
a	0	0	0.4	0	0.6	0
<SOS>	0.6	0.1	0	0.2	0	0.1

Table 1: Bigram probabilities for your generation system in Problem 2.c. Here, the probability of observing "you" given "are" is 0.6. <SOS> is the start of sequence token.

Answer:

1. Start with <SOS> and generate all possible next words and their probabilities:
are: .6, you: .1, concerned: 0, I: .2, want: 0, a: .1

2. Select top 2
are (.6)
I (.2)

3. Generate all possible next words and their probabilities
are (.6):
are: 0, you: .6. concerned: .2, I: 0, want: .05, a: .1
I (.2):
are: 0, you: .05, concerned: .01, I: 0, want: .7, a: .05

4. Choose top two paths
are (.6) * you (.6)
are (.6) * concerned (.2)
I (.2) * want (.7)
I (.2) * you (.05) or I (.2) * a (.05)

5. Generate all next words
are (.6) * you (.6):
are: .2, you: 0, concerned: .4, I: 0, want: .4, a: .1

I (.2) * want (.7):
are: 0, you: .5, concerned: 0, I: 0, want: 0, a: .5

6. Choose top 2 paths
are (.6) * you (.6) * concerned (.4) = .144
are (.6) * you (.6) * want (.3) = .108
I (.2) * want (.7) * you (.5) = .07
I (.2) * want (.7) * a (.5) = .07