UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR11061 NATURAL LANGUAGE UNDERSTANDING (LEVEL 11)

Thursday 3rd May 2018

14:30 to 16:30

INSTRUCTIONS TO CANDIDATES

Answer QUESTION 1 and ONE other question.

Question 1 is COMPULSORY. If both QUESTION 2 and QUESTION 3 are answered, only QUESTION 2 will be marked.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

MSc Courses

Convener: G. Sanguinetti
External Examiners: W. Knottenbelt, M. Dunlop, M. Niranjan, E. Vasilaki

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. THIS QUESTION IS COMPULSORY

The following questions are short answer questions. Answer as concisely as possible. Often a sentence or two is sufficient; never write more than a paragraph. Longer answers will not receive more credit.

(a) Consider a language model, which, given a sentence $x = x_1, ..., x_{|x|}$, defines its probability as:

$$Pr(x) = \prod_{i=1}^{|x|+1} Pr(x_i \mid x_0, ..., x_{i-1})$$

where $x_0 = \langle \textsc{Start} \rangle$ and $x_{|x|+1} = \langle \textsc{Stop} \rangle$ are beginning- and end-of-sequence tokens not in the vocabulary.

i. An n-gram language model, feedforward language model, and recurrent neural network (RNN) language model are three ways of defining a language model. What are the key ways in which these models differ from each other? *[3 marks]*

ii. Estimating an n-gram language model usually requires some method for smoothing. Do feedforward or RNN language models require this? Why or why not? *[2 marks]*

iii. Is the model used in word2vec a language model as defined above? Why or why not? *[2 marks]*

(b) Suppose that you want to build a part-of-speech tagger. An example input x and output y is:

Input $x =$	All	mimsy	were	the	borogoves
Output $y =$	Adv	Adj	Verb	Det	Noun

Your goal is to build a probabilistic model $Pr(y \mid x)$.

i. How would model this problem with a single RNN? Draw the unrolled RNN for this example. *[2 marks]*

ii. Write down the probability distribution that your RNN models. *[2 marks]*

iii. How would you use two RNNs to model this problem? *[1 mark]*

iv. How would this change the probability distribution? *[1 mark]*

QUESTION CONTINUES ON NEXT PAGE

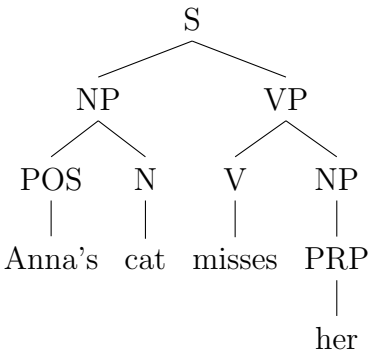
(c) Word2vec is widely used to produce representations for words, but it has some shortcomings.

i. Word2vec suffers from out-of-vocabulary problems. Explain a technique that you might use to mitigate them. [*2 marks*]

ii. Suppose that you are tasked with building a job recommendation system. You could use pre-trained word embeddings to match applicants to job postings based on a personal profile. What ethical problems might arise if you did this? [*2 marks*]

iii. Explain a technique that you might use to mitigate these problems. [*2 marks*]

(d) Explain how a recurrent neural network grammar uses up to three RNNs to encode the information used to predict the word "cat" in the following parse tree. You may find it helpful to illustrate.



[*4 marks*]

(e) Recall that a frame semantic parser takes as input a sentence with a target predicate. For example, with the predicate in **bold**:

The San Francisco Examiner **issued** a special edition yesterday .

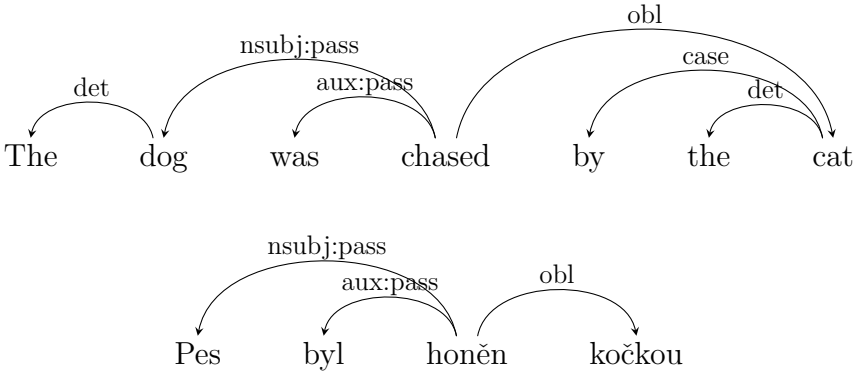
The parser should label spans of the sentence that serve as arguments to the predicate. Given the above, it should output:

$$\underbrace{\text{The San Francisco Examiner}}_{\textsc{Arg0}} \text{ issued } \underbrace{\text{a special edition}}_{\textsc{Arg1}} \ \underbrace{\text{yesterday}}_{\textsc{Argm-Tmp}} \ .$$

Explain how this can be done using an RNN. You should clearly describe the input and output of the model. [*2 marks*]

2. ANSWER EITHER THIS QUESTION OR QUESTION 3

Universal dependencies is a dependency annotation scheme that uses the same inventory of dependency labels for all languages. For example, the English and Czech translations are annotated in this scheme, and they share the dependency labels `nsubj:pass`, `aux:pass`, and `obj`.



Universal dependency annotations are available in over sixty languages, which opens the door to an intriguing possibility: could we train a single parser that works on all of these languages? In fact, there has been some successful work on this problem that synthesizes many ideas from the course, which we will focus on in the questions below.

(a) A major difficulty with training a cross-lingual parser is that the set of input words will be different for each language. This problem might be solved with word embeddings, if all words in all languages could be embedded in the same space. For simplicity, assume that you have only two languages, English and Czech. You can train English word embeddings on English monolingual data, and you can train Czech word embeddings on Czech monolingual data. Because the models are learned independently, there is no reason to expect that the embedding of "dog" in the English model will relate to the embedding of "Pes" in the Czech model, even though these words are translations of each other. Nevertheless, we might expect that the two embedding spaces have similar structure: for example, "dog" and "cat" should have similar embeddings in English, since they share many characteristics; likewise, their respective translations "Pes" and "kočkou" should also have similar embeddings in Czech. If we could find a function f that maps the Czech embedding of "Pes" to the English embedding of "dog", then f might also be a good mapping of "kočkou" to "cat". How would you use such a function in a single model that can parse both Czech and English? *[3 marks]*

QUESTION CONTINUES ON NEXT PAGE

(b) Now suppose that you also have a Czech-English dictionary, containing word pairs like ⟨"Pes","dog"⟩. How could you could use this information to learn a function f that converts a Czech word embedding into an English word embedding? Describe the input and output of your model, the model itself, and how you would train your model. [*9 marks*]

(c) Do you think your approach would work equally well if you instead learned a mapping from English embeddings to Czech embeddings? Justify your answer. [*3 marks*]

(d) Once you have f, you have a way to learn the multilingual parser from a consistent input representation. Although the output dependencies have a common set of labels, the behavior of dependencies in different languages can differ in systematic ways. So, the parses learned for one language might not help with parsing another language. Describe two general ways in which this might happen. [*6 marks*]

(e) These systematic differences in syntax relate to differences in typology. These typological differences are well-studied, and each language's typology can be (crudely) represented as a set of features with categorical values. For example, we can use the World Atlas of Language Structures (WALS) to look up the "Order of Adjective and Noun" feature, which is "Ajective-Noun" for both Czech and English, though it is "Noun-Adjective" for most languages, and "No dominant order" for many others. We could attempt to account for this knowledge by conditioning each parsing decision not only on the words of the sentence, but also on typological features. How would you modify the architecture of a stack-LSTM to account for this, and what effect do you think you would observe? [*4 marks*]

3. ANSWER EITHER THIS QUESTION OR QUESTION 2

Natural language inference (NLI) is a difficult problem that has attracted significant attention recently. An NLI system is given a pair of sentences as input—one called the *premise*, and one called the *hypothesis*. The system must decide if the premise *entails* the hypothesis—that is, if the hypothesis is necessarily true, given the hypothesis. It signals its decision with one of three labels: `entailment`, `contradiction`, or `neutral`. For example, given this sentence pair:

Premise. A girl in a red coat, blue head wrap and jeans is making a snow angel.

Hypothesis. A girl outside is playing in the snow.

The hypothesis is true whenever the premise is true, so the output label should be `entailment`. This is a difficult problem: it requires understanding that "a girl in a red coat, blue head wrap and jeans" must also be "a girl"; that someone who "is making a snow angel" must also be "playing in the snow"; and that someone who "is playing in the snow" must also be "outside".

The system should return `contradiction` when the premise contradicts the hypothesis, and it should return `neutral` when the premise neither entails nor contradicts the hypothesis, as in the following examples:

Premise. A black race car starts up in front of a crowd of people.

Hypothesis. A man is driving down a lonely road.

Label. `contradiction`

Premise. An older and younger man are smiling.

Hypothesis. Two men are smiling and laughing at the cats playing.

Label. `neutral`

A large set of examples for this problem called the Stanford Natural Language Inference (SNLI) dataset became available a few years ago.

(a) How would you use a single LSTM to model this problem? Sketch the architecture of your model, showing its input and output, and explain how you would train it. [*9 marks*]

(b) What are some problems of your model? [*3 marks*]

QUESTION CONTINUES ON NEXT PAGE

(c) Explain how you would design an alternative architecture that might alleviate some of the problems with using a single LSTM architecture, explain how you would train it, and explain why you think this will help. [*9 marks*]

(d) Models trained on the SNLI dataset have recently reached accuracies of almost 90%. We still don't yet have a clear idea of what many of these models learn. The intent is that they must learn logical relationships between the premise and hypothesis, though this must require a great deal of lexical learning—for example, that "a crowd" is not "lonely". What sorts of associations might such a model learn that are not simply logical? How would you test for this? [*4 marks*]