REPRINT

COMP 34412

Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Natural Language Systems

Date: Tuesday 22nd May 2018

Time: 09:45 - 11:45

Please answer all Questions.

You should show your working for all questions that require calculation or simulation of an algorithm, since marks will be awarded for answers that demonstrate the use of the correct method.

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text

[PTO]

1. a) Outline an algorithm for calculating the perceived pitch of a speech signal **3 marks**, and explain how this could be used to lengthen or shorten the signal without changing its apparent pitch. **2 marks** without altering it

   b) Define precision, recall, and F measure: **1 mark**

   Suppose that you have a collection of tweets T1, ..., T5, which a classifier has assigned to 0 or more of the classes '*love*', '*hate*', '*disgust*', '*optimism*' as in Table 1 (the labels in brackets are the Gold Standard labels in each case, so Y (Y) in the first column for T1 says that T1 was classified as expressing 'love', and the correct answer for this cell is indeed Y; and Y (N) in the last column for this row says that it was also classified as expressing optimism, but that this was not the correct answer)

	love	anger	disgust	optimism
T1	Y (Y)	N (N)	N (N)	Y (N)
T2	Y (Y)	N (Y)	N (N)	N (N)
T3	N (Y)	N (N)	N (N)	N (N)
T4	Y (N)	N (N)	Y (Y)	Y (N)
T5	Y (Y)	Y (Y)	Y (Y)	Y (N)

Table 1: Classifying tweets by sentiment

   Calculate the precision, recall and F-measure of the classifier on this set of tweets **3 marks**

   Describe a situaton in **either** precision would be more important than recall when classifying the sentiments **or** or one where recall is more important than precision for this task **1 mark**

   c) Suppose that you had the following sets of probabilities:

   Emission probabilities The entry `"love":{"noun":0.3, "verb":0.7}` means that there is a 30% chance that any occurrence of '*love*' is a noun and a 70% chance that it is a verb, and likewise for the other words in the dictionary.

```
{"love":{"noun":0.3, "verb":0.7},
 "I": {"pronoun": 1.0},
 "my": {"det": 1.0},
 ".": {"stop": 1.0}}
```

   Transition probabilities The entry `"det":{"noun":0.9, "verb":0.1}` means that there is a 90% chance that any occurrence of a determiner will be followed by a noun and a 10% chance that it will be followed by a verb. The other entries in this table should be interpreted similarly.

```
{"det":{"noun":0.9, "verb":0.1},
  "pronoun": {"noun": 0.1, "verb":0.9},
  "noun": {"det": 0.5, "stop":0.5},
  "verb": {"det": 0.3, "stop": 0.7}}
```

Show the first four steps that an HMM-based tagger would go through when using these probabilities to assign part-of-speech tags to the sentence '*I love my love* .'
5 marks

d) Words in Martian can be decomposed into several parts: a root, a prefix marking spin, a suffix marking charm,and a second suffix marking strangeness. The word '*abcdxyx*' has the properties `spin=yes, charm=no, strangeness=yes`, the word '*xbcdxxy*' has `spin=no, charm=no, strangeness=no`, and '*abcdyyx*' has `spin=yes, charm=yes, strangeness=yes`. Work out what the root and affixes that make up these words must be **3 marks**, and provide categorial descriptions of them **2 marks**.

Total marks for Q1: 20

2. a) Describe each of the components of a typical speech recogniser. **5 marks**

   b) What data do you have to supply in order to train such a recogniser? **3 marks**

   c) Describe the ways in which a grammar might be useful when training and using a speech recogniser. For each such use, say whether it is essential to the task at hand or whether it is something which might be useful but could be omitted. **7 marks**

   d) How does a concatenative unit selection synthesiser work **3 marks**, and what role would a recogniser have in developing such a synthesiser? **2 marks**

   Total marks for Q2: 20

3. a) Outline the algorithms underlying transition-based and graph-based approaches to robust dependency parsing **4 marks for each**

   b) Show the steps, and the intermediate states that these steps will produce, that the transition-based parser MALT would go through given the tagged text '*the:AT man:NN ate:VV a:AT peach:NN*' and the set of rules in Figure 1. **5 marks**

```
input:[NN, _], stack:[AT, _], relations:_ ==> leftArc
input:[NN, _], stack:[VV, _], relations:_ ==> rightArc
input:[VV, _], stack:[NN, _], relations:_ ==> leftArc
input:[§*, _], stack:_, relations:_ ==> shift
```

Figure 1: Decision rules for MALT

   c) Show the graph that the rules in Fig 2 would generate from the text '*I love my love.*' and sketch how Edmonds algorithm would eliminate any loops that the graph contains. You do **not** need to propagate exactly the same scores that Edmonds algorithm assigns to arcs as it transforms the graph, and you do **not** need to show how the tree is reconstructed once all the loops have been eliminated. **5 marks**

```
I:SUBJ (.*)* love:HD 0.5;
my:SPEC (.*)* love:HD 1.0;
love:HD (.*)* love:COMP 0.3;
love:MV (.*)* .:HD 0.4;
I:HD (.*)* love:dtr 0.2;
```

Figure 2: Rules for dependency parsing

   d) The basic MALT algorithm is linear in the length of the input text: the algorithm for obtaining a minimum spanning tree is $O(N^2)$ in the number of arcs in the graph. Why might you nonetheless choose to use the minimum spanning tree algorithm? **2 marks**

Total marks for Q3: 20

4. a) What do WordNet similarity metrics measure? Describe a strategy for using them to compare the similarity of the words '*fork*' and '*junction*'. **4 marks**

   b) Outline an entailment algorithm for dependency trees: this algorithm should allow '*I watched a black bird*' to entail '*I saw a bird*', assuming that the dependency trees for these two sentences are as in Figure 3 **6 marks**

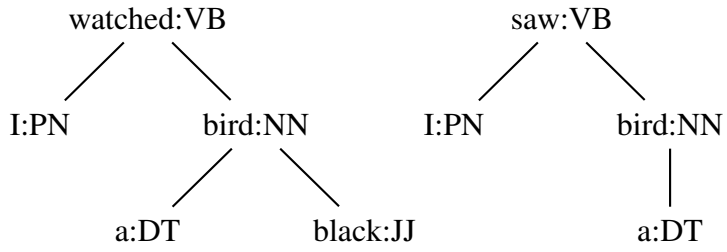

Figure 3: Dependency trees for '*I watched a black bird*' and '*I saw a bird*'

   c) What would you have to do to the trees in Figure 4 to ensure that the matching algorithm supported the inference from '*I know this is an easy question*' to '*I know this is a question*', but that for '*I doubt this is an easy question*' to '*I doubt this is a question*' the entailment is the other way round. Show any annotation that you would add to the trees in order to achieve this. **5 marks**

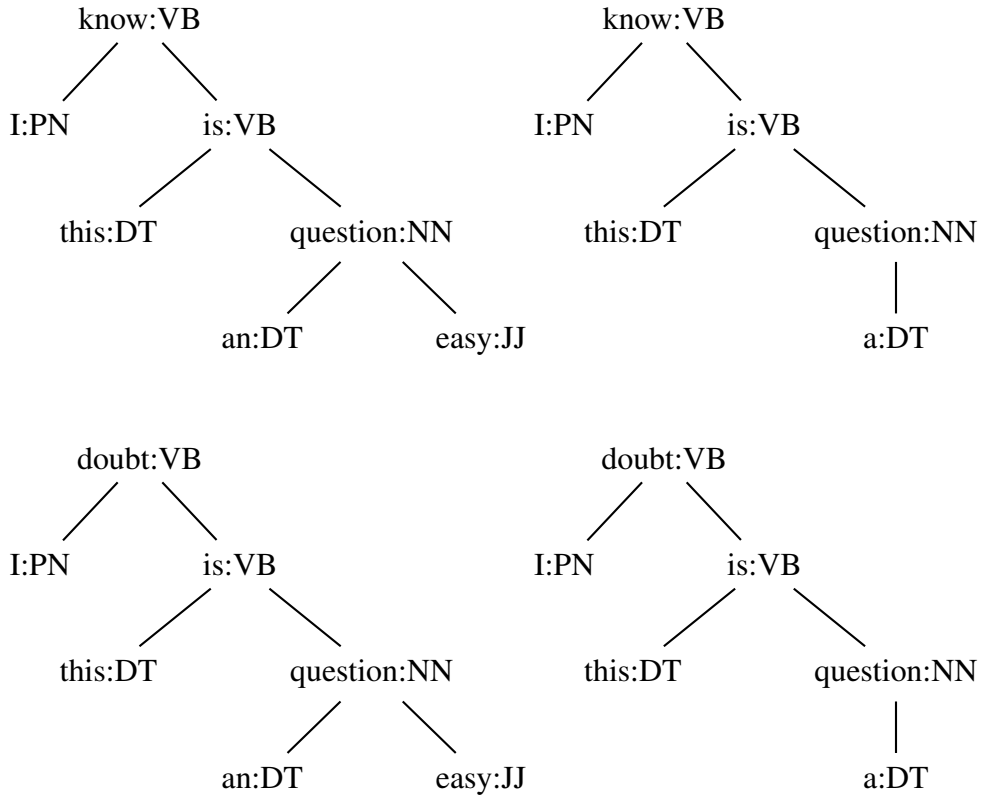

Figure 4: Dependency trees for '*I know this is an easy question*', '*I know this is a question*', '*I doubt this is an easy question*' and '*I doubt this is a question*'

d) Sketch a rule that would allow you to infer '*This is a question*' from '*I know this is a question*', based on the trees in Figure 5. How could you use such rules in a general purpose inference engine for natural language? **5 marks**

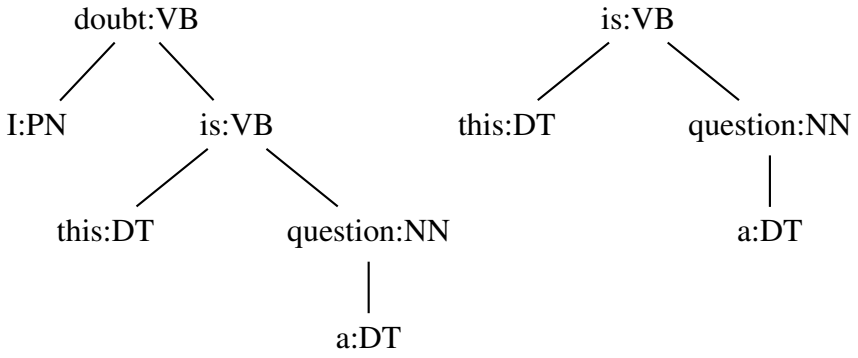

Figure 5: Dependency trees for '*I know this is a question*' and '*This is a question*'

Total marks for Q4: 20

END OF EXAMINATION