The University Of Sheffield.

Data Provided: None

DEPARTMENT OF COMPUTER SCIENCE **Autumn Semester 2010-2011**

TEXT PROCESSING **2 hours**

Answer THREE questions.

All questions carry equal weight. Figures in square brackets indicate the percentage of available marks allocated to each part of a question.

Registration number from U-Card (9 digits) — to be completed by student

1. a) What is *part-of-speech-tagging*? Explain why this task is difficult, i.e. why it requires something more than just simple dictionary look-up. [20%]

   b) Describe and explain the TF.IDF term weighting scheme used in information retrieval. Include the formula (or formulae) for computing TF.IDF values as part of your answer. [20%]

   c) Text compression techniques are important due to the growth in volume of textual data that must be stored and transmitted. Briefly explain the differences between:

      (i) **symbolwise** and **dictionary** methods for text compression [10%]

      (ii) **modelling** versus **coding** steps in text compresson [10%]

   d) Write a short program in Python, which reads a text file and prints out all occurrences of URL ('web address') expressions found, satisfying certain requirements. The requirements are that the expression should:

      i. fall within a single line of the file
      ii. appear within double quotes (")
      iii. start with `http://`
      iv. end with `.htm` or `.html`
      v. between this start and end, allow any characters to appear *except* ", < and > [20%]

   e) We want to compute the similarity of documents using a measure, which is a *count-sensitive* variant of a *Jaccard coefficient*, that is defined in the following way. Let w_A and w_B denote the counts of word w in documents A and B, respectively. The measure is then computed as follows, where Σ_w implicitly ranges over the full set of terms appearing in either document:

$$\frac{\Sigma_w \, \mathtt{min}(w_A, w_B)}{\Sigma_w \, \mathtt{max}(w_A, w_B)}$$

   Assume we have Python code that will count the term occurrences in a document, and store this information in a dictionary, i.e. in which each key is a term that appears in the document, and the associated value (an integer) is its count in the document. Define a Python function which takes two arguments, each a dictionary storing the count information for some document, and which computes and returns the similarity score for those two documents according to the above measure. [20%]

2. a) Explain the differences between *direct*, *transfer* and *interlingual* approaches to Machine Translation. [15%]

   b) When applied to translating from French to English, the IBM approach to statistical machine translation might be expressed by the following equation:

$$E^* \quad = \quad \underset{E}{\text{argmax}}\ P(E) \cdot P(F|E)$$

   Explain what this equation means, and indicate the role played by the components $P(E)$ and $P(F|E)$ in the process of translation. [15%]

   c) Show how the equation given in (b) is derived using the Bayes Rule. What is the benefit of this approach as compared to one attempting to use the probability $P(E|F)$ directly? [20%]

   d) We want to compress a large corpus of text of the (fictitious) language *Fontele*. The writing script of Fontele employs only the six letters found in the language name (`f,o,n,t,e,l`) and the symbol ␣, used as a 'space' between words. Corpus analysis shows that the probabilities of these seven characters are as follows:

Symbol	Probability
`e`	0.3
`f`	0.04
`l`	0.26
`n`	0.2
`t`	0.04
`o`	0.1
␣	0.06

      (i) Show how to construct a Huffman code tree for Fontele, given the above probabilities for its characters. [25%]

      (ii) Use your code tree to encode the message "`telefone␣noel`" and show the resulting binary encoding. For this string, how much length does your codetree encoding save as compared to a minimal fixed length binary character encoding for a 7 character alphabet? [10%]

   e) Explain the difference between **static**, **semi-static** and **adaptive** techniques for text compression, noting their key advantages and disadvantages. How would the Huffman approach that you used in (d) be categorized in these terms? [15%]

3. a) Explain what is meant by *summarisation*. Identify and explain some of the ways in which different kinds of summary have conventionally been classified. [15%]

   b) Discuss the criteria that have been used by *shallow* approaches to automated summarisation for identifying text segments in a document containing significant information. Discuss how multiple features may be combined in deciding which text segments to select. [25%]

   c) Discuss alternative approaches to evaluating the performance of automatic summarisation systems. [20%]

   d) Write a Python program to generate simple sentence-extraction summaries of documents. Your program should take two command line arguments, as in the following example call:

      ```
      python mysumm.py document.txt 30
      ```

      Here, the first argument is the file containing the document to be summarised, whilst the second argument specifies the 'compression ratio', indicating that the summary should be not more than 30% of the length of the original document, computed in terms of the *number of words* that each contains. You can assume that document files have been preprocessed so that *each sentence appears on a separate line*.

      For any document, your program should print a summary that consists of the first N sentences of the document, where N is indirectly determined by the compression ratio. Thus, summaries should contain as many whole sentences as is possible *without* exceeding the length limit set by the compression ratio. You will need code to compute the number of words within a sentence, for which you can assume a simple model of tokenization, e.g. that the words of a sentence are just the maximal alphabetic sequences to be found within the line. [20%]

   e) Consider a simplistic measure of the 'information content' of sentences that calculates the *proportion* of words within a sentence that are *not stopwords*. Show how your Python program of (d) can be modified so as to generate summaries consisting of the top-scoring sentences of the document according to this measure. Assume that a stopword file is supplied to your program as a third command line argument, e.g. as in the command line call:

      ```
      python mysumm.py document.txt 30 stoplist.txt
      ```

      You will need code to read and store the stopwords, and to use this information in computing scores for sentences. These scores can in turn be used for sorting the sentences, so that the top-scoring sentences can then be printed. [20%]

4. a) Explain what is meant by the use of (i) a *stoplist* and (ii) *stemming* in the context of information retrieval. Discuss the potential benefits of using these methods for an information retrieval system. [20%]

b) Explain what is meant by an *inverted index*, and why such indices are important in the context of information retrieval. Show how Documents 1 and 2 below would be stored in an inverted index, using stemming and the following stoplist: {`and, does, is, my, not, to, will, your`} [25%]

Document 1: Your data is corrupt. Corrupted data does not hash.

Document 2: Your data system will transfer corrupted data files to trash.

c) What is the similarity measure used by the *vector space model* of information retrieval? Compute the similarity score between the following query and the two documents given in (b), to determine which (if either) of them would be ranked more highly as relevant to the query. Assume that term frequency values are used for the term weights in document vectors.

Query: my corrupted data files will not hash [30%]

d) Two information retrieval systems, System 1 and System 2, each return a ranked list of 10 documents from a given collection for a particular query. It is known that there are 12 documents relevant to the query within the collection. The following table shows whether the document returned at each rank position is relevant (✓) or not (×), for each system.

Document rank	System 1	System 2
1	✓	×
2	×	✓
3	✓	×
4	✓	✓
5	✓	✓
6	×	×
7	×	×
8	✓	✓
9	×	×
10	×	✓

Discuss the metrics that have been used within information retrieval to compare the performance of alternative information retrieval systems. Illustrate your answer with reference to the above table of results. [25%]

END OF QUESTION PAPER