The University of Sheffield

DEPARTMENT OF COMPUTER SCIENCE

Autumn Semester 2003-2004 **2 hours**

TEXT PROCESSING

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.**

1. Provide short answers to the following questions.

   (a) What is mutual information?

   [10%]

   (b) What is cross entropy? Describe one way in which cross entropy is used in statistical text processing.

   [10%]

   (c) Probabilistic language models are very popular in text processing. Using the notation of conditional probabilities, write down an expression for the exact probability of a sentence containing l words. Now write down an approximation to this. The approximation should use only unigram and bigram probabilities (i.e., those having the form $P(w_k)$ where $k \in 1 \ldots l$ or $P(w_i|w_i - 1)$ where $i \in 2 \ldots l$). Briefly explain what information is used in the first expression but ignored in the second. Why might one choose to use the second form of the expression in preference to the first?

   [10%]

   (d) What is smoothing? Why is it important? Name and briefly describe two distinct methods of smoothing probability distributions which arise in probabilistic language models.

   [10%]

   (e) What is the purpose of the following Unix commands?

   ```
   cat genesis.txt | tr -cs ’[A-Za-z]’ ’[\012*]’ | sort | uniq -c | sort
   -nr
   ```

   What is the definition of a word which this command is implicitly adopting?

   [10%]

(f) Specify a pipeline of Unix tools which takes as input a plain text file and produces as output a list of words occurring in the file 20 times.

[10%]

(g) State clearly what the following Perl program would print out when run, giving reasons.

```
@y = ('a', 'b', 'c');  # line 1
print @y;
$x = @y;               # line 2
print $x;
$y[$x] = $y[@y - 1];   # line 3
print @y;
```

[20%]

(h) What is printed out by each of the following Perl programs?

i.
```
$_ = 'bananarama';
s/(an)*a/$1;
print;
```

ii.
```
$_ = 'uggawuggahuggamugga';
s/((a)(.)(u))/$2$4/g;
print;
```

[10%]

(i) The Perl regular expression

```
/^((a|b)+)\1$/
```

will match which of the following strings?

```
aa
ab
ba
aba
abba
abab
aababba
babbab
baaa
baab
```

[10%]

2. Write a Perl program which reads standard input (up to the end of file) and prints out an integer representing the number of occurrences in the input data of the most frequently occurring word in that data, along with a note of the word.

   For example, if the input data were:

```
the man with the glasses
saw the woman with the saw
```

   then the most frequently occurring word would be "the", occurring 4 times, so the output should be:

```
the : 4
```

   If there are several equally frequent words in the text, they should all be printed out with the number of occurrences. For example if the input data were:

```
the glasses with the glasses
saw the glasses with the glasses
```

   then the most frequently occurring words would be "the" and "glasses", occurring 4 times, so the output should be:

```
the : 4
glasses : 4
```

   Take a word to be any consecutive sequence of alphabetic characters separated by white space from any adjacent word. Assume the input contains only alphabetic characters, spaces, and linebreaks, i.e., there is no punctuation or numeric data.

   [100%]

3. In order for documents to be retrieved efficiently from an information retrieval system, information is stored in an inverted index. Suppose that you are given the two documents below:

   Document 1
   She sells sea shells at the shore.

   Document 2
   Many types of sea shells sell at the fish market.

   (a) Explain what an inverted index is. Show how Documents 1 and 2 would be stored in an inverted index using a stoplist of your choice and stemming. Justify the types of words chosen to include in the stoplist.

   [25%]

(b) Create a document-by-word matrix for Documents 1 and 2 after stemming and removal of stopwords. Assume the matrix cells represent the number of times a word appears in Documents 1 and 2.

[25%]

(c) Calculate the similarity of Documents 1 and 2 using a measure of your choice and justify why you have chosen this measure. (Useful information: $\sqrt{4} = 2$; $\sqrt{6} \approx 2.45$).

[25%]

(d) Write a Perl program which takes as input Documents 1 and 2 and for each word present in the two Documents, it prints out the name of the document the word has been found in and its position in the sentence. So the output of your program should be a table that looks like this:

```
document1: She 1
document1: sells 2
document1: sea 3
document1: shells 4
document1: at 5
document1: the 6
document1: sea 7
document1: shore 8
document2: Many 1
document2: types 2
document2: of 3
document2: sea 4
document2: shells 5
document2: sell 6
document2: at 7
document2: the 8
document2: fish 9
document2: market 10
```

The first column shows the document name, the second column the words found in the document and the third the position of the word in the sentence. Assume that a word is any consecutive sequence of alphabetic characters separated by white space from any adjacent word. Assume your documents have been pre-processed so that each sentence occupies a new line (i.e., you don't have to segment your documents into sentences, this has been already done for you).

[25%]

4. (a) Describe the main features of the approach to statistical machine translation adopted by the IBM research group. Justify your answers.

[50%]

(b) To what extent is this approach purely statistical and to what extent does it rely on linguistic intuitions? Justify your answers.

[25%]

(c) Do you believe that the approach is limited to closely related language pairs such as English and French, or will it be applicable to any language pair for which a large enough corpus of text is available? Justify your answers.

[25%]

5. You work for a large company where there are many meetings, both of internal staff and between staff and external clients. Meetings are recorded in form minutes. The company's files of minutes are large and the material has to be kept for many years since it may be necessary to check back on decisions taken early in large projects.

   You are asked to design a retrieval system so that company staff can locate minutes on a particular topic. The company wants the system to be reliable and effective.

   (a) Outline the design of your system, indicating the particular features it will have that are intended to meet the company's requirements (you can assume that minutes are always clearly dated and have explicit lists of participants).

   [50%]

   (b) The company is willing to allow the installation of a pilot system so your approach can be evaluated under realistic conditions. Describe, in detail, your design for the evaluation: what data will you consider, what aspects of your system will you evaluate, and why?

   [25%]

   (c) Suppose you are running a pilot evaluation using a nine-document collection and the results of two information retrieval systems about this collection shown below. The presence of the symbol √ indicates that the document is relevant to the query.

System 1 Results		
Docs	Ranking	Relevant
d8	1	√
d2	2	√
d7	3	√
d3	4	
d5	5	√
d1	6	
d4	7	
d6	8	
d9	9	

System 1 Results		
Docs	Ranking	Relevant
d8	1	√
d2	2	
d5	3	
d1	4	
d4	5	√
d6	6	√
d3	7	
d9	7	
d7	9	√

   Explain how you would evaluate the two information retrieval systems on the basis of these results. Discuss the evaluation measures you would use and illustrate your answer using the examples above.

   [25%]

END OF QUESTION PAPER

Solutions for COM3110 Exam 2003-2004
Setter: Mirella Lapata

1. (a) Mutual information is the reduction in uncertainly of one random variable due to knowing about another, or in other words, the amount of information one random variable contains about another. Mutual information can be calculated as follows.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(X) + H(Y) - H(X,Y) \\ &= \sum_x p(x) \log \tfrac{1}{p(x)} + \sum_y p(y) \log \tfrac{1}{p(y)} + \sum_{x,y} p(x,y) \log p(x,y) \\ &= \sum_{x,y} p(x,y) \log \tfrac{p(x,y)}{p(x)p(y)} \end{aligned}$$

(b) The cross entropy is useful when we don't know the actual probability distribution p that generated some data. It allows us to use some m which is a model of p (i.e., an approximation to p). The cross entropy of m on p is defined by:

$$H(p,m) = \lim_{n\to\infty} \frac{1}{n} \sum_{W\in L} p(w_1,\ldots,w_n) \log m(w_1,\ldots,w_n)$$

Cross entropy is used in language modeling to evaluate how well a given language model performs on unseen data. The lower the cross entropy, the better the model.

(c) Here's the exact expression for the probability of a sentence containing l words:.

$$P(w_1 \ldots w_l) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1,w_2) \ldots P(w_l|w_1 \ldots w_{l-1})$$

The above can be approximated as follows:

$$P(w_1 \ldots w_l) \approx P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_2) \cdots P(w_l|w_{l-i})$$

Contextual information is ignored in the second expression. We might chose the second expression over the first one, because the second one will be easier to estimate (i.e., it has less parameters).

(d) The task of reevaluating some of the zero-probability and low-probability n-grams, and assigning them non-zero values, is called smoothing. Smoothing techniques are used in a variety of statistical natural language processing applications as a means to address data sparseness, an inherent problem for statistical methods which rely on the relative frequencies of word combinations. The problem arises when the probability of word combinations that do not occur in the training data needs to be estimated. Methods for smoothing probability distributions are Adding-one smoothing (or Laplace's law), absolute discounting, or Good-Turing smoothing.

Add-one smoothing:

$$P(w_2|w_1) = \frac{C(w_1,w_2)+1}{C(w_1)+V}$$

where $C(w_1,w_2)$ and $C(w_1)$ is the number of times w_1,w_2 and w_1 appear in the corpus and V the size of the vocabulary.

Absolute discounting:

$$Pabs(w_n|w_1 \dots w_{n-1}) = \tfrac{r-\delta}{N} \qquad \text{if } r > 0$$

$$Pabs(w_n|w_1 \dots w_{n-1}) = \tfrac{(V-N_0)\delta}{N_0 N} \quad \text{otherwise}$$

where r is $C(w_1 \dots w_n)$, N is the total number of times $w_1 \dots w_{n-1}$ has been seen, V is the size of the vocabulary, and N_0 is the number of word types that were unseen after this context.

Good-Turing smoothing:

$$r^* = (r+1)\frac{N_{r+1}}{N_r}$$

where r^* is the corrected estimate for word occurring r times. The proportion of unseen words is estimated by proportion of singletons, the proportion of N_1 singletons is estimated by proportion of N_2 doubletons, etc.

(e) The Unix commands tokenise the text, translate all alphabetic characters to lower case, print each word in a separate line, sort the words alphabetically, squeeze and count repetitions, and finally sort numerically in a descending order. A word is any sequence of letters and numbers separated by white space.

(f) `cat genesis.txt | tr -cs '[A-Za-z]' '[\012*]' | sort | uniq -c | grep '^20'`

(g) `abc3abcc`

The y-array has 3 entries, a, b, c, which are printed unspaced. x tasks on the value of the length of y, i.e., 3, which is printed. The update on y assigns the element of y indexed by 2 (length of y minus 1) to the position indexed y x (3), extending the array to have a further copy of c, total length 4 elements.

(h) i. Prints `banrama`

ii. Prints `uggauggauggaugga`

(i) It matches `aa`, `abab`, `babbab` only. (Any non-empty string consisting of a string followed by an exact copy).

2.
```
while ($line = <>) {
    chop $line;
    @words = split(/\s+/,$line);
    foreach $member (@words) {
        $hash_words{$member} += 1;
    }
}

$counter = 0;

foreach $k (keys %hash_words) {
    if ($hash_words{$k} > $counter) {
        $counter = $hash_words{$k};
    }
}

 foreach $k (keys %hash_words) {
     if ($hash_words{$k} == $counter) {
         print "$k : $hash_words{$k}\n";
     }
 }
```

50% for using a hash to store the words and their frequencies, 25% for looping through the hash to find the words with the highest frequency, and 25% for printing these words correctly.

3. (a) An inverted index is a data structure that lists for each word in a document collection all documents that contain it and the frequency of occurrence in each document. An inverted index makes it easy to search for 'hits' of a query word. One just goes to the part of the inverted index that corresponds to the query word and retrieves the documents listed there. A more sophisticated version of an inverted index also contains position information. Instead of just listing the documents that a word occurs in, the position of all occurrences in the document are also listed. A position of occurrence can be encoded as a byte offset relative to the beginning of the document. An inverted index with position information lets us search for *phrases*. Here's an inverted index for Documents 1 and 2:

Number	Words	Documents
1	fish	2
2	market	2
3	sea	1, 2
4	sell	1, 2
5	shell	1, 2
6	shore	1
7	type	2

The following words have been included in the stoplist: {she, at, the, many, of}. Words which are too frequent among the documents in a collection are not good discriminators and are usually filtered out as potential index terms. Articles, prepositions, and conjunctions are natural candidates for a list of stopwords. Words like *many* and *she* can also be used as stop words given that they do not have a lot of semantic content and one would expect to find them in a lot of documents.

(b) This is the document-by-word matrix for Documents 1 and 2.

	fish	market	sea	sell	shell	shore	type
Document 1	0	0	1	1	1	1	0
Document 2	1	1	1	1	1	0	1

(c)

$$cos(\vec{Doc}\ 1, \vec{Doc}\ 2) = \frac{0\cdot 1 + 0\cdot 1 + 1\cdot 1 + 1\cdot 1 + 1\cdot 1 + 1\cdot 0 + 0\cdot 1}{\sqrt{0^2+0^2+1^2+1^2+1^2+1^2+0^2}\sqrt{1^2+1^2+1^2+1^2+1^2+0^2+1^2}} = \frac{3}{\sqrt{4}\sqrt{6}} = 0.61$$

(d)
```
foreach $member (@ARGV) {
    open(FILE,$member);

    $counter = 0;

    while ($line = <FILE>) {
        chop $line;
        $line =˜ s/\.//g;
        @words = split(/\s+/,$line);

        foreach $w (@words) {
            ++$counter;
            print "$member: $w $counter\n";
        }
    }
    close(FILE);
}
```

4. This is an essay-type question. These are general points that the students should discuss:

(a) Machine translation can be cast in terms of the noisy-channel model. The approach was pioneered by IBM in the early 90's. If we want to translate French into English, we shall assume that someone has E in his head but by the time he writes it down it's *corrupted* and becomes F. To recover the original E, need to reason about: (a) the kinds of things people say in English and (b) how English gets turned into French. Given a French sentence F the approach searches for English E that maximises $P(E|F)$.

$$\begin{aligned} \text{argmax}_E\, P(E|F) &= \text{argmax}_E\, \tfrac{P(E)\cdot P(F|E)}{P(F)} \\ &= \text{argmax}_E\, P(E)\cdot P(F|E) \end{aligned}$$

We choose not to model $P(E|F)$ directly. Instead, we break things apart to get good translations even if the probability numbers are not that accurate. $P(F|E)$ will ensure that a good E will have words that generally translate to words in F. $P(E)$ will help save us from bad English and $P(F|E)$ only needs to say whether a bag of English words corresponds to a bag of French words. $P(E)$ and $P(F|E)$ can be trained independently.

In order to model $P(F|E)$ we need a sentence sentence-aligned parallel (bilingual corpus) and a model of translation $\sim P(F|E)$. The latter assumes word alignments which we don't have. We can approximate those by using EM, an unsupervised learning algorithm that can learn alignment probabilities.

(b) The approach as described above does not rely on linguistic knowledge about the two languages.

(c) The approach is limited to closely related language pairs. It will be difficult to translate languages that are structurally very different (e.g., languages that have different notions of what counts as a word or languages that have radically different word orders).

5. This is an essay-type question. These are general points that the students should discuss:

(a) The idea is to build an IR system for the company, to enable staff retrieve minutes for meetings. The user will type in a query and the system will retrieve a list of documents. It should be described here what is the IR model of choice (boolean or vector-based), whether stemming and a list of stopwords will be used, what is the data structure that will hold the documents, how the system will be updated, etc.

(b) Information retrieval systems are typically evaluated using precision, recall and the F-measure. Since this system is built for a company with real users, a more detailed evaluation can be conducted, where user satisfaction is taken into account.

(c) In terms of precision, recall, and F-measure the two systems perform identically. By using precision at a cutoff we will get an impression of how a system ranks relevant documents before non relevant ones. With precision at 5 we see that System 1 returns 100% of the relevant documents, whereas System 2 returns only 50%.