CFCS1: Assignment 4

Yansong Feng

March 17, 2010

Handed out:	12 March 2010
Due date:	22 March 2010

Introduction

In this assignment, you will construct your own language models according to a small corpus and use your language model to help with a real world problem. We collected 178 sentences from 8 short stories, which are prepared for ESL(English as a Second Language) learners. These stories are short but useful for non-native English speakers to get started for reading practice.

The files needed for the assignment have been packaged into `as4_material.tar.gz`. Download this file to your working directory, and unpackage its contents using the command `tar -zxvf as4_material.tar.gz` at a shell prompt.

This will create several text files described below:

`sentence_tokens.txt`	The text of the original stories (all words are changed into lower case, punctuations are separated with space; in this assignment, we refer **word** to both normal words and punctuations, numbers, etc).
`vocab.txt`	The vocabulary file contains all the words that appear in the dataset (including words, punctuations, numbers, etc).

To make it easier to process the data in Matlab, each word has been assigned an ID number, and there are two further files:

`vocab_mapping.txt`	A list showing the mapping of ID numbers to words.
`sent_IDs.txt`	The stories again, with all words replaced by their IDs.

We have already extracted unigram counts for each word appearing in the dataset, and the counts of pairs of adjacent words in the dataset. **The most useful data have been converted into Matlab forms.** Load the file `as4_data.mat` into Matlab. This will creates the following variables:

- `vocab`, an array containing all the words in order, and the 12th symbol, `<s>`, indicating the start of a sentence (So that there are always enough previous words to predict the first actual word, e.g., sentence: `it was cool` will be written as: `<s> it was cool`. If you use a bigram LM, you can write the sentence probability as $P(it| < s >) * P(was|it) * P(cool|was)$). This symbol should be treated like any other words.
- `unigram_counts`, a vector containing the number of occurrences in the dataset of each word. For example, `unigram_counts(1)=2` is the number of times *!* occurs in the dataset.
- `bigram_counts`, a matrix containing counts of pairs of adjacent words in the dataset. For example, `bigram_counts(1,2)=2` is the number of times *!* is followed by " in the dataset, `bigram_counts(467,485)=5` is the number of times *the* is followed by *top* in the dataset.

1 Language Analysis

1. In Matlab, list five most frequent pairs of adjacent words in this dataset.

[2]

2. Compute the probability of observing each word in the vocabulary, assuming successive words in a sentence are independent. Show the results of $P(is)$ and $P(for)$.

[3]

3. Calculate the entropy of the distribution. What is the expected number of bits per word needed to code a sentence?

[3]

4. If we expect a higher entropy for this distribution, how different should the word distribution look like compared to the current one?

[2]

2 Using Language Model

A ESL beginner who just starts learning English has written 4 sentences, and does not know which are better:

```
<s> he going to take a shower
<s> he were going take a shower
<s> he was going to a shower
<s> they were going to take a shower
```

In this Section, you will construct language models to help the beginner evaluate his sentences.

1. Based on the dataset, create a *unigram* language model (You have done this in previous Section!), and compute the probabilities of these sentences.

[2]

2. Use the data in `bigram_counts` to compute the full set of bigram probabilities for words, $p(w_i \,|w_{i-1})$, where w_i is any word and w_{i-1} is the preceding word.

[3]

3. Use this *bigram* language model to compute the probabilities of these sentences. What kind of results do you get? Why does this happen ?

[4]

4. Do you have any ideas to avoid the situation happened in Question 3 ? Explain your suggestions in details, implement one of them, and re-compute the probabilities. Do you satisfy with your results?

[4]

5. In your opinion, which language model is better to help the beginner evalute his sentences? Why?

[2]

To submit

For the assignment, you should submit:

- Written answers to the required questions.
- Printouts of all the Matlab functions you wrote.
- A (brief) transcription of your Matlab session, together with the output that you used to answer the questions.

Please hand in a hard copies of the required material by 4:00pm on the due date to the Informatics Teaching Organisation, Level 4, Appleton Tower. If you have questions regarding the assignment, please contact Yansong Feng at `yansong.feng@ed.ac.uk`.