ACCELERATED NATURAL LANGUAGE PROCESSING

Assignment 1

S1818699, S1884908

July 2018

Contents

1 Introduction

2 Random Sequence Generation - Question 4

2.1 Raw Application of Language Model

After having generated and built a language model, the model itself can be used as is to generate new sequences.

2.2 Random generation

Algorithm 1 Random Generation

```
procedure GENERATE_FROM_LM(Num_Chars, Model, Valid_Char_List)
    sequence = empty
    bigram_in ← '#' + random(Valid_Char_List)
    chars_left ← Num_Chars − 1
loop:
    if chars_left > 0 then
        pos_tris ← [bigram_in + Valid_Char_List].
        distribution ← model[pos_tris].
        bins ← cumulativesum(distribution)
        seq_pos ← random_bin_select(bins)
        new_seq ← pos_tris[seq_pos]
        bigram_in ← new_seq[0 : 1]
        sequence ← sequence + new_seq[2]
        chars_left ← chars_left − 1
        goto loop.
    return sequence
```

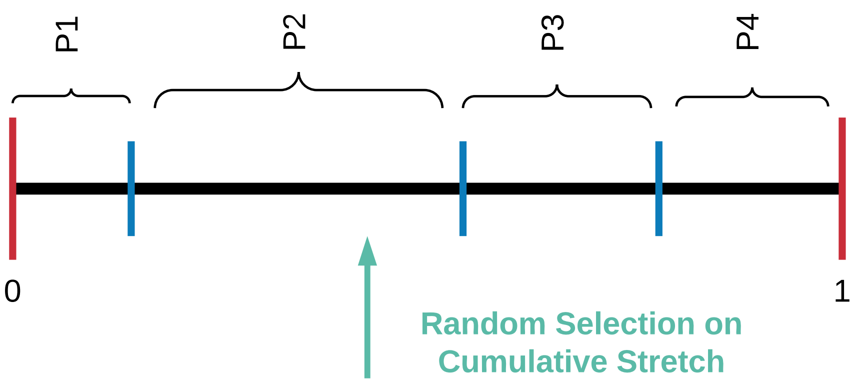

Figure 1: Random selection from a probability distribution

2.3 Results

3 Perplexity Computation - Question 5

This part of the assignment dealt with utilising the generated language models to assess the content of a given text.

3.1 Perplexity Computation

A perplexity measure attempt to measure how well a given model predicts a selected text sample. A low perplexity indicates the model is well suited to predicting the selected text, whilst a high perplexity indicates the model is unsuited for the text selected. The general equation for perplexity computation is as follows:

$$PP_M = P_M\left(w_1....w_n\right)^{-\frac{1}{n}} \tag{1}$$

Taking logs:

$$log\left(PP_M\right) = log\left(P_M\left(w_1....w_n\right)^{-\frac{1}{n}}\right) \tag{2}$$

$$log\left(PP_M\right) = -\frac{1}{n} \times log\left(P_M\left(w_1....w_n\right)\right) \tag{3}$$

$$log\left(PP_M\right) = -\frac{1}{n}\sum_{i=1}^{n}\left(log(P_M(w_1)) + ... + log(P_M(w_n))\right) \tag{4}$$

3.2 Comparative Results

4 Conclusions