CE314/887 Assignment 1
Language Models and Smoothing

Annie Louis

24 October, 2016

Introduction

In this assignment, you will build unigram and bigram language models, implement Laplace smoothing and use the models to compute the perplexity of test corpora. Part of the assignment asks you to do the computation by hand and the latter part asks you to write code for the same.

- Carefully read the instructions and details about the data.
- Pay attention to what the vocabulary of each model is. The vocabulary to use for unigram and bigram models is explicitly mentioned. Be sure to read carefully.
- When the phrase "Show your work" appears in this assignment, you should write down the formula, plug in the numbers and write the output probability.
- You can write the code in any programming language you are comfortable with.
- Read the *What to Submit* section before you submit your work.
- For the report, be concise. Lengthy explanations and derivations are not required.
- This assignment is worth 10% of your module marks

Important note on plagiarism

The assignment should be completed **individually**. Collaboration is not allowed. You may discuss the lecture material and concepts, and clarify the questions asked in the assignment with other students. But you may not discuss the solutions or approaches. Regarding code, it is okay to help someone with a language's syntax or library functions but you may not give them the actual program statements or show them your code.

Carefully read the guidelines here. `http://www.essex.ac.uk/ldev/resources/plagiarism/default.aspx`

All plagiarism cases will be referred to an academic offenses procedure. No allowance regardless of whether the act was intended or unintended.

Input data

There are two datasets given to you. Uncompress `a01_data.zip` and look at the files.

Toy dataset: The files `sampledata.txt, sampledata.vocab.txt, sampletest.txt` comprise a small toy dataset. `sampledata.txt` is the *training corpus* and contains the following:

<s> a a b b c c </s>
<s> a c b c </s>
<s> b c c a b </s>

Treat each line as a *sentence*. <s> is the start of sentence symbol and </s> is the end of sentence symbol. To keep the toy dataset simple, characters `a-z` will each be considered as a *word*. i.e. The first sentence has 8 tokens, second has 6 tokens, and the last has 7.

The file `sampledata.vocab.txt` contains the vocabulary of the training data. It lists the 3 word types for the toy dataset:

a
b
c

`sampletest.txt` is the *test corpus*.

Actual data: The files `train.txt`, `train.vocab.txt`, and `test.txt` form a larger more realistic dataset. These files have been preprocessed to remove punctuation and all words have been converted to lower case. An example sentence in the train or test file has the following form:

<s> the anglo-saxons called april oster-monath or eostur-monath </s>

Again every space-separated token is a word. The above sentence has 9 tokens. The `train.vocab.txt` contains the vocabulary (types) in the training data.

Important: Note that the <s> or </s> are not included in the vocabulary files. In this assignment, the term UNK will be used to indicate words which have not appeared in the training data. UNK is also not included in the vocabulary files but you will need to add UNK to the vocabulary while doing computations. While computing the probability of a test sentence, any words not seen in the training data should be treated as a UNK token.

Important: You do not need to do any further preprocessing of the data. Simply split by space you will have the tokens in each sentence.

Q1. Computing a unigram model (10 marks)

Use the **Toy dataset**. The vocabulary is the words in the `sampledata.vocab.txt` plus the UNK token. **Do not** include <s> and </s> in the vocabulary.

a) Compute the probabilities *by hand* in a unigram language model without smoothing. Show your work for $P(a)$, $P(c)$, $P(\text{UNK})$. Create a table in the following format and list all of the probabilities in the unigram model.

X	**P(X)**
a	
b	
...	

b) Smooth the model using Laplace smoothing. Show your work for $P(a)$, $P(c)$, $P(\text{UNK})$. Show all the smoothed probabilities in a table.

Q2. Computing a bigram model (20 marks)

Use the **Toy dataset**. The vocabulary is the words in `sampledata.vocab.txt`, plus the UNK token and </s> symbols. <s> should be included only in the context or history. </s> should not be included in the history but only as the following word. The table below should clarify for you.

a) Compute the probabilities *by hand* in a bigram language model without smoothing. Show your work for $P(b|a)$, $P(\text{UNK}|\ \text{<s>})$, $P(\text{UNK}|\text{UNK})$. Create a table in the following format and list all of the probabilities in the model.

	$P(w_i\|w_{i-1})$				
w_{i-1} \ w_i	a	b	c	UNK	</s>
a					
b					
c					
UNK					
<s>					

b) Smooth the model using Laplace smoothing. Show your work for $P(b|a)$, $P(\text{UNK}|\ \text{<s>})$, $P(\text{UNK}|\text{UNK})$. Show all the smoothed probabilities in a table.

Q3. Computing sentence probabilities (10 marks)

Use the **Toy dataset**. There are 5 sentences in `sampletest.txt`. Using the *smoothed* models above, compute the probability of each sentence by hand. For unigram probability, you should ignore the <s> and </s> symbols.

a) Show your work for sentence numbers 3, 4, 5, for each model: unigram and bigram.

b) Fill in the probabilities of all the sentences in a table.

S	$P_{uni}(S)$	$P_{bi}(S)$
<s> a b c </s>		
<s> a b b c c </s>		
...		

Q4. Compute corpus perplexity (10 marks)

Use the **Toy dataset**. Treat `sampletest.txt` as the test corpus. All 5 sentences together comprise the test corpus.

a) Compute perplexity of the test corpus using a smoothed unigram model. Show your work. Ignore the <s> and </s> symbols. Remember that the correct number of tokens in the test corpus will be different if you do not exclude correctly.

b) Compute the perplexity of the test corpus using a smoothed bigram model. Show your work. Here you will include <s> and </s> while computing the probability of the test corpus. However, while computing the number of tokens in the test corpus, include </s> but not the <s> token.

c) Which perplexity is higher? Explain why in a sentence or two.

Q5. Implementation of the models (30 marks)

a) Write a function to compute unigram unsmoothed and smoothed models. Print out the unigram probabilities computed by each model for the **Toy dataset**.

b) Write a function to compute bigram unsmoothed and smoothed models. Print out the bigram probabilities computed by each model for the **Toy dataset**.

c) Write a function to compute sentence probabilities under a language model. Print out the probabilities of sentences in **Toy dataset** using the smoothed unigram and bigram models.

d) Write a function to return the perplexity of a test corpus given a particular language model. Print out the perplexities computed for `sampletest.txt` using a smoothed unigram model and a smoothed bigram model.

Check: Do your numbers agree with those computed by hand. If not, go back and check if you have the right computation.

Q6. Run on large corpus (20 marks)

Now use the **Actual dataset**. Train smoothed unigram and bigram models on `train.txt`. Print out the perplexity under each model for

a) `train.txt` i.e. the same corpus you used to train the model.

b) `test.txt`

c) Compare the perplexities computed for (a) and (b). Which one is higher and why? Explain in one or two sentences.

What to submit: Report and Code

A single `registration_number.zip` file. The uncompressed folder should contain a report and a code directory.

Report. A file containing the answers for Q1, Q2, Q3, Q4 and Q6. For Q6, write the perplexity values and your explanation.

Code. Your code should run without any arguments. It should read files in the same directory. Absolute paths must not be used. When downloaded, you code should run with a simple command such as `java LangModel` or `python LangModel.py`. A README.txt file should have a single line giving the command to run your code. Check your code by downloading your .zip file into a different machine and testing that it runs without modification.

When your code is run, it should print values in the following format.

```
---------------- Toy dataset ----------------

=== UNIGRAM MODEL ===
- Unsmoothed -
 a:0.0  b:0.0  ...

- Smoothed -
 a:0.0  b:0.0  ...
```

```
=== BIGRAM MODEL ===
- Unsmoothed -
       a    b    c  UNK  </s>
 a     0.0  ...
 b     ...
 c     ...
 UNK   ...
 <s>   ...

- Smoothed -
       a    b    c  UNK  </s>
 a     0.0  ...
 b     ...
 c     ...
 UNK   ...
 <s>   ...

== SENTENCE PROBABILITIES ==
 sent                 uprob  biprob
 <s> a b c </s>       0.0    0.0
 <s> a b b c c </s>   ...
 ...

== TEST PERPLEXITY ==
unigram:  0.0
bigram:  0.0

---------------- Actual dataset -----------------

PERPLEXITY of train.txt
unigram:  0.0
bigram:  0.0

PERPLEXITY of test.txt
unigram:  0.0
bigram:  0.0
```

Assessment criteria

What we are looking for in your answers:

Clear understanding of the concepts demonstrated by taking the right approach, giving the right formula, correct substitution and accurate answers

Ability to use concepts learned in class demonstrated by clear answers to questions which ask to analyze numbers and output

Delivering the requested software solutions demonstrated by code which satisfies the criteria, outputs the required solutions cleanly, runs without dependencies, contains proper comments. You will not be evaluated on the efficiency of your code or algorithms as long as they run in reasonable time.