Homework 4

11-411/11-611: Natural Language Processing

Due Thursday, October 11th at 11:59 PM Eastern Time

1 Introduction

In this homework, you will be building your first Language Model. You will be expected to build a N-gram Language model. You will also implement Laplace Smoothing (a Lazy version) to account for unknown words. The assignment is broken down into following subsections

2 Programming (60 points)

Refer to the notebook that is provided as part of the handout. You will be downloading the required data files, along with the **utils.py** and **main.py**. **Do not edit these files**. After you are done coding, paste the functions and the classes you implemented in the **lm.py** that is part of the handout and then upload the **utils.py** and **lm.py** to the HW-3 Programming Submission, without zipping them.

3 Written (40 points)

Answer the following questions based on the code you've written. You can use the latex file in the handout to answer the questions. Upload the PDF to the HW-3 Written Submission

3.1 N-Gram counts(10 points)

Train the language model on the data/bbc/business.txt dataset for n = 2 and n = 3. Then do the same for data/bbc/sports.txt datset

1. *[2.5 points]* How many unique 2-grams are present in the business data-set?
   Answer: 83819

2. *[2.5 points]* How many unique 3-grams are present in the business data-set?
   Answer: 141221

3. *[2.5 points]* How many unique 2-grams are present in the sports data-set?
   Answer: 77398

4. *[2.5 points]* How many unique 3-grams are present in the sports data-set?
   Answer: 135645

3.2 Most Discriminant N-grams(10 points)

A virus has ravaged the servers of BBC and all metadata about news articles has disappeared. Luckily, you have agreed to help BBC with your superior N-gram modelling skills and have proposed to build a N-gram classifier to tell the different news domains apart.

From each data set, select the two most-discriminant tri-grams from the list of top n-grams sorted in descending order by frequency.

Hint: The more rare a tri-gram is to a data-set, the more discriminant it is for that data-set

1. Entertainment:
   (a) ('fi', 'series', 'firefly')
   (b) ('sci', 'fi', 'series')
2. Politics:
   (a) ('get', 'another', 'chance')
   (b) ('would', 'get', 'another')
3. Sport:
   (a) ('who', 'twisted', 'his')
   (b) ('melzer', 'who', 'twisted')
4. Technology:
   (a) ('the', 'days', 'lol')
   (b) ('was', 'the', 'days')
5. Business:
   (a) ('to', 'be', 'confirmed')
   (b) ('still', 'to', 'be')

3.3 Song Attribution (8 points)

You are scrolling through the top hits playlist on Spotify when you notice a new unknown song at the top. It's recorded by an anonymous artist but the lyrics sound uncannily similar to some other songs you have heard. You have narrowed it down to three artists but are unable to choose one: *it could be any of them!*

You go along the rest of the day thinking who could it be. You reach Posner Hall to attend an NLP 11-411/611 lecture and David is teaching language models. Wait: language models. It suddenly hits you: language models can help in this task!

Train tri-gram ('n=3', 'smoothing= 0.1') language models on collections of song lyrics from three popular artists ('data/lyrics/') and use the model to score a new unattributed song.

Note In reality, perplexity should only be used to compare language models when they have the same vocabularies but we will relax that condition for this question.

1. *[6 points]* What are the perplexity scores of the test lyrics against each of the language models?
   (a) Taylor Swift: 138.00663307990817
   (b) Green Day: 522.5401188730924
   (c) Ed Sheeran: 521.2574891234094
2. *[2 points]* Who is most likely to be the lyricist?
   Answer: Taylor Swift

3.4 Introduction to Decoding and Text Generation(8 points)

Run the code provided in the notebook and fill in the answers below

1. *[6 points]* For each of these phrases ‘s1‘ to ‘s3‘, what are the top five word candidates after the sequence? Remember to not include, the special tokens we added during training to indicate end-of-sentence and start-of-sentence.

   (a) s1: ('during', 'to, 'taking', 'one', 'was')

   (b) s2: ('the', 'hit', 'a', 'jean', 'collateral')

   (c) s3: ('a', '2004')

2. *[2 points]* Report any one of the generated sentence here. Which generation mode do you think is better and why? I think "Max-probability decoding" mode would be great because it can always generate the sentence with maximum probability and it looks like really working. I report the example made by random and max mode like below.

   Input: "number", "three"
   random output: 'number three during its run to date breach ofcom s programme code'
   max output: 'number three during its first week on home entertainment release the film s director richard eyre issued a warning earlier in his'

3.5 Comparision to a GPT (4 points)

Run the code provided in the notebook and fill in the answers below.

1. *[2 points]* What is the perplexity of your LM model?
   Answer: 4889.176510567337

2. *[2 points]* What is the preplexity of the GPT-2 model?
   Answer: 50.56884002685547