**N-Gram Language Models**

**Matthew Ruiz**

**Contents**

1. **INTRODUCTION**
   1. Objectives
2. **DATA**
3. **PROGRAM** **LOGIC**

3.1 Sentence segmentation and word tokenization guidelines

1. **RESULTS**

4.1 Top 15 N-Grams

4.2 Randomly Generated Sentences

1. **EVAULATION AND REFLECTION**

5.1 Issues

5.2 Lessons Learned

5.3 Summary

1. **Future Considerations**
2. **INTRODUCTION**

This report: -

* Explains the decisions made during the design phase of the assignment
* Details similarities and differences among two n-gram models
* Summarizes the issues faced during implementation and as a result, the lessons learned

**Objectives**

The objectives for this assignment are as follows: -

* Download and process a large text dataset in python using the csv library
* Perform sentence and word tokenization
* Calculate N-gram counts and probabilities
* Compare the characteristics of the N-grams across different models
* Generate random sentences using the models

**2. DATA**

a. There were two datasets used for this assignment:

* + 1. [winemag-data\_first150k.csv](https://www.kaggle.com/zynicide/wine-reviews/data)
    2. [Reviews.csv](https://www.kaggle.com/snap/amazon-fine-food-reviews/data)
  1. For demonstration, the first 50,000 lines of the *Reviews.csv* file and the first 100,000 lines of the *winemag-data\_first150k.csv* were processed.
     1. The reason for the disparity between the amount of data being processed is directly related to the nature of the two files. The Amazon dataset includes reviews for a vastly diverse set of products, whereas the wine-review dataset contains strictly wine-related reviews. Therefore, the difference in vocabulary, which, terms of size, is in favor of the Amazon dataset, would cause unfair n-gram counts. For example, when the same amount of reviews is processed for each dataset, the vocabulary size of the Amazon dataset is about two times larger than that of the wine-reviews dataset.
  2. The data is read in using a generator. This was used to prepare for the eventual implementation of larger datasets.

**3. PROGRAM** **LOGIC**

**3.1 Sentence segmentation and word tokenization**

Guidelines on how text is processed were established.

These include: -

* @#%^&\*()(\_)-+=";: symbols are mapped to the ‘$’ token
* HTML tags are mapped to the ‘.’ token
  + It became apparent that HTML tags usually replace end of sentence punctuation. Therefore, the decision to map these tags to the ‘.’ token made sense.
  + In some cases, HTML tags would be present in conjunction with end of sentence punctuation.
    - Therefore, all instances of two or more adjacent periods are mapped to a single ‘.’ token.
* When generating n-grams, ‘.’ tokens are replaced by the </s> token.
  + When this occurs, the bigram, (<s>, token) and the trigram (<s>, <s>, token) are generated. This is necessary for maintaining accurate n-gram counts.
* Unicode characters are encoded with ‘ascii’.
  + Since the Python 3’s default character encoding is UTF-8, we handle charsets: ascii and UTF-8. All other characters are bypassed.
* Numbers of any kind are mapped to the single token ‘0.0’.
  + Without this rule, number would appear less than anticipated,
* All remaining text is normalized to lowercase.

**4. DATA**

**4.1 Top 15 N-grams**

In this section, you will find the top 15 n-grams for the unigram, bigram, and trigram models for each dataset.

**4.1.1 Unigrams**

*V = 39,974 V = 52,638*

*N = 4,629,148 N = 4,397,762*

|  |  |  |
| --- | --- | --- |
| **wine-reviews.csv** | | |
| Wn | C (Wn) | P(Wn) |
| , | 354,973 | 0.076682 |
| </s> | 274,385 | 0.059273 |
| and | 267,167 | 0.057714 |
| the | 166,531 | 0.035974 |
| a | 141,046 | 0.030469 |
| of | 124,407 | 0.026874 |
| with | 100,627 | 0.021737 |
| this | 76,171 | 0.016454 |
| is | 73,307 | 0.015835 |
| wine | 57,367 | 0.012392 |
| flavors | 50,001 | 0.010801 |
| in | 48,254 | 0.010423 |
| it | 42,580 | 0.009198 |
| to | 41,383 | 0.008939 |
| fruit | 36,616 | 0.007909 |

|  |  |  |
| --- | --- | --- |
| **amazon-reviews.csv** | | |
| Wn | C (Wn) | P(Wn) |
| </s> | 298,763 | 0.067935 |
| the | 161,151 | 0.036643 |
| , | 145,038 | 0.032979 |
| i | 128,944 | 0.029320 |
| and | 109,795 | 0.024966 |
| a | 102,758 | 0.023365 |
| to | 87,139 | 0.019814 |
| it | 79,674 | 0.018116 |
| of | 68,342 | 0.015540 |
| is | 63,329 | 0.014440 |
| this | 57,612 | 0.013000 |
| for | 47,265 | 0.010747 |
| in | 45,958 | 0.010450 |
| $ | 44,386 | 0.010092 |
| my | 38,538 | 0.008763 |

We see that, although the *amazon-reviews.csv* file had a smaller dataset, albeit less than a 5% difference, it’s vocabulary size included almost thirteen thousand more unigrams. This should be expected given that the *wine-reviews.csv* file pertains to strictly wine; whereas the amazon dataset contained reviews for a much more diverse set of products.

Another interesting observation is the significant difference in the use of ‘i’ in each dataset. Comparing the 128,944 times it occurred in the amazon dataset to the measly 273 times it occurs in the wine-review dataset, it becomes apparent that customer reviews aren’t always driven by the same intent.

**4.1.2 Bigrams**

*V = 408,271 V = 637,350*

|  |  |  |
| --- | --- | --- |
| **wine-reviews.csv** | | |
| Wn-1 | Wn | C(Wn|Wn-1) |
| , | with | 40,271 |
| <s> | the | 33,625 |
| , | this | 24,048 |
| on | the | 22,904 |
| with | a | 18,921 |
| this | is | 18,693 |
| , | but | 17,310 |
| is | a | 16,130 |
| , | and | 16,066 |
| <s> | it | 15,932 |
| <s> | it’s | 15,595 |
| the | palate | 14,507 |
| in | the | 14,312 |
| and | a | 14,002 |
| finish | </s> | 13,807 |

|  |  |  |
| --- | --- | --- |
| **amazon-reviews.csv** | | |
| Wn-1 | Wn | C(Wn|Wn-1) |
| <s> | i | 45,386 |
| <s> | the | 18,128 |
| , | and | 16,945 |
| , | but | 16,919 |
| of | the | 13,108 |
| <s> | it | 12,680 |
| , | i | 11,114 |
| in | the | 10,919 |
| i | have | 10,875 |
| it | is | 10,457 |
| this | is | 9,669 |
| it | </s> | 9,331 |
| $ | </s> | 8,700 |
| is | a | 8,010 |
| <s> | this | 7,830 |

Once again, the use of the ‘i’ indicates the possessive nature of the reviews inside of the amazon dataset. Alternatively, the bigrams (, with) and (with a), which ultimately form the trigram (, with a), give us some insight into one of the common structures of sentence predicates. For roughly 22% of the times that the bigram (with a) appears, the following word is an adjective of quantity: touch (~5%), hint (~4%), and etc.

Knowing this should help with us in determining the writer’s attitude towards the product they are viewing.

**4.1.2 Trigrams**

*V = 1,225,704 V = 1,822,022*

|  |  |  |  |
| --- | --- | --- | --- |
| **wine-reviews.csv** | | | |
| Wn-2 | Wn-1 | Wn | C(Wn|Wn-2Wn-1) |
| this | is | a | 9,507 |
| <s> | <s> | and | 9,220 |
| , | with | a | 8,934 |
| <s> | <s> | is | 8,394 |
| , | this | is | 7,276 |
| on | the | finish | 7,010 |
| the | finish | </s> | 6,526 |
| <s> | <s> | , | 6,420 |
| <s> | the | palate | 5,496 |
| <s> | the | wine | 5,391 |
| in | the | mouth | 5,295 |
| on | the | palate | 5,008 |
| <s> | drink | now | 4,867 |
| the | wine | Is | 4,639 |
| drink | now | </s> | 4,590 |

|  |  |  |  |
| --- | --- | --- | --- |
| **amazon-reviews.csv** | | | |
| Wn-2 | Wn-1 | Wn | C(Wn|Wn-2Wn-1) |
| <s> | it | is | 3,810 |
| <s> | i | have | 3,489 |
| <s> | <s> | is | 3,341 |
| <s> | if | you | 2,712 |
| <s> | this | is | 2622 |
| <s> | they | are | 2576 |
| , | but | i | 2515 |
| <s> | <s> | have | 2501 |
| This | is | a | 2454 |
| <s> | i | was | 2193 |
| <s> | i | am | 2172 |
| <s> | <s> | love | 1966 |
| this | is | the | 1900 |
| <s> | i | would | 1897 |
| a | lot | of | 1822 |

**4.2 Randomly Generated Sentences**

In this section, we will compare three randomly generated sentences using the unigram, bigram, and trigram models for each dataset. Output is hand-corrected for capitalization to improve readability.

**Unigrams**

*wine-reviews.csv*

–Bergamot the controlled are $ can with Montagne butterscotch rate smoothly

–Bolstered aromas with offering intensity drink

–Evolved

*amazon-reviews.csv*

–At, far make out rather juices one the need made money the many of usually claiming at many sharing, the both are plant am were the are than large burn in from > these the not doctor

–The best by they - basically a about, candies differently stinky a and I'm previous

–Imune with for tea

**Bigrams**

*wine-reviews.csv*

–The next 0.0 years easily around the flavors of feline smells unexpectedly dry core of a balanced alcohol $0.0

–It's too full-bodied and cabernet sauvignon blanc, wild berry and delicious wine offers abundant raspberry and white chocolate

–Winery's regular

*amazon-reviews.csv*

–We were a pastrami many years I think you'll forgive me headaches in it, because mine had in your dog when I prefer plain ol' t, I taste, and smooth cup

–The recipient of king syrup, over-seasoned products with both spoke to have been drinking hot water hits a kid with a crumbly state it has helped me, nor in it just organic, what went on to apply any other than other important to about 0.0 cents more concentrated juices

–It but he wolfed it was thrilled about the apple or no worries here for a similar yourself on the years now

**Trigrams**

*wine-reviews.csv*

–A touch of sweetness which gives it immediate appeal

–Little by little rocks, lemon and lime show here, but that mature fruit, developed fruit flavors mingle on the palate in dense minerality as well

–In fact, that promises a long-aged wine

*amazon-reviews.csv*

–Package of lake champlain hot cocoa I have dh and I was a little more like the jalapeno ones

–Is it, then again - I sure hoped I could make my childs play dough recipe

–Enjoy, and I like that strong, something has changed but I would really be cool if they measure up to every meal, and he gobbled it right away and offered to kids, no problems with them

**Summary**

As expected, the more context on which we train the model, the more intelligible the sentences. The unigram sentences show little to no relation between tokens. In the bigram sentences, we see some local word-to-word coherence. Lastly, the trigram sentences display more English-like sentences.

**5. EVAULATION AND REFLECTION**

**5.1 Issues**

When generating n-grams from a given list of tokens, it is important that the appropriate n-grams are generated at the beginning of each sentence. Since we’re processing the text on an as-is basis, it was difficult to determine the necessary steps that were required to accurately generate these n-grams.

Example: -

* Given the text “My dog came home. I was happy.”, then,

tokens = [‘My, ‘dog’, ‘came, ‘home’, ‘.’, ‘I’, ‘was’, ‘happy’, ‘.’]

* Therefore, as stated in section 3.1, to account for punctuation:
  + the unigram (</s>) must be generated twice, one for each period,
  + the bigrams (home, </s>), (<s> i), and (happy, </s>), and
  + the trigrams (came, home, </s>), (<s>, <s>, i), and (was, happy, </s>)

must be generated to maintain an accurate n-gram count.

**5.2 Lessons Learned**

* The first lesson I learned from this assignment was the importance of handling Unicode characters. Not only did this realization open my eyes to the importance of character handling, it also led me down a path towards understanding the importance of creating software that works internationally.
* Now, the latter was not necessarily implemented into this assignment, but as a result of the different Unicode characters portrayed in both datasets, it became apparent that this was a consideration that I should take into account moving forward.

**5.3 Summary**

Overall, I believe that this assignment was a success. Although I could have put more thought into the design phase of the assignment, before writing a single piece of code, due to the lack of time I was able to allocate for it, the program delivers on its requirements.

**6. FUTURE CONSIDERATION**

Looking over the source code, I have violated of the DRY principle. The files: *unigram.py*, *bigrams.py* and *trigrams.py* have a lot of the same methods. With more time spent on the initial design, I would have been able to catch this. Therefore, although slightly different in the way the n-grams are accessed, processed, and distributed, the three files provide extremely similar functionality. After reviewing the necessary steps to refactoring, there was not enough time to do so before deadline.

In the future, I would like to add some functionality, such as: determining the probability of a sentence entered by the user and sentiment analysis based off of sentence generation. This is the first assignment in college that I enjoyed exploring and look forward to working with in the future.