

The Department of Computer Science

**CIS4515**

**Practical Data Analysis**

Level 7

Coursework 1

Task 2

Question 5 (Report)

2023/2024

### Student: Joseph Z. Moyo

**Student No: 25792334**

**1. Introduction**

Amazon’s Data Analysis department ran a mini project to analyse the positive customer reviews of gaming products that were sold. The task involved the use of the NLTK library in Python for natural language processing (NLP). This task was important to identify the common products that were referenced by customers in their positive reviews, enabling Amazon to note potential items to focus their stock budget on. In this exercise it was revealed that the top five best reviewed games were Resident Evil, Grand Theft Auto, Tony Hawk, Eternal Darkness and Mario Smash Bros. The X box gaming console was also highly referenced in the reviews, implying perhaps the gaming platform is the most popular for the customers.

**2. Portfolio Task**

The task involved importing text data of Amazon reviews, pre-processing it to query the collocations found in the data, revealing important information that could give Amazon clear understanding of its target market.

**2.1. Question 1**

The initial step in the exercise was to import the reviews from the text file into the python jupyter notebook workspace. Several modules were imported from the NLTK python library to read and pre-process the reviews text.

The regular expression tokenisation model ‘regexp\_tokenize’ was used to tokenise the words in the reviews. It was called with an argument to ensure that words that have an apostrophe were not split (NLTK, 2023). The ‘corpus()’ module and its methods were used to remove common English language stop words. The tokens were subsequently tagged using the ‘pos\_tag()’ module to identify the part-of-speech they belonged to. The tokens were further grouped in pairs called bigrams as they appeared in the reviews. This pairing helps to reduce the dimensions of the data while conveying more information (Medium, 2019). The first exercise’s aim was to produce a collection of bigrams of token pairs with their respective part of speech tags.

The screenshots of the documented code are shown:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**2.2. Question 2**

The follow up task involved extracting the 40 most important bigrams in the reviews text using the co-occurrence frequency algorithm introduced in class tutorial exercises. The data was imported and pre-processed in the same manner as in question 1. A function called ‘freq\_of\_bigrams()’ was created and it took as input the pos\_tagged bigrams produced by the code from question\_1.

The screenshot for the code can be seen below:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

The code produces the top 40 most frequently observed bigrams based on frequency of mentions. The top 10 mostly include common expressions for gaming experience such as ‘great game’, ‘game play’, ‘fun game’, ‘replay value’, and more. The top 10 also included actual video game titles such as ‘Resident Evil’, ‘Grand theft’ and ‘Super Smash’. There were also frequent references to a particular gaming platform in the form of the ‘x box’ console.

**2.3. Question 3**

The bigrams dictionary produced in question 1 was filtered using a list of pos\_tag combinations given in the tutorials to remove pairs that aren’t considered as collocations.

The screenshots below present the code.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The observed collocations and their frequencies are different for some of the entries compared with the dictionary obtained in question 2. There is noticeable change in frequency for some of the pairs that were retained after filtration, e.g., the collocation ‘game play’ is still number two on the list but the frequency dropped from 240 to 186.

**2.4. Question 4**

The collocations were alternatively evaluated using the mutual information statistical metric which determines the importance of a pair from the statistical combination of its probability with the probabilities of the two member words, all normalised based on the total number of words in the tokens list (Icalem, 2018).

Two methods were explored:

* using the Mutual Information formula given in the tutorial to mechanically calculate the MI values from the bigram frequency function from question 3 along with a custom function to calculate token frequencies.
* utilising the NLTK library’s modules for computing MI values.

The code used for the first method is shown below.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

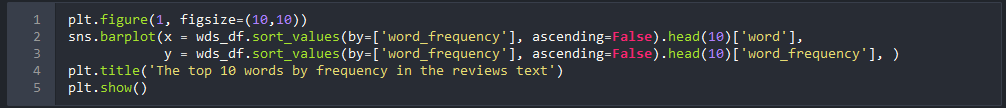


Figure 1 below shows the top 10 most common words in the reviews text.

A graph of a bar

Description automatically generated with medium confidence

**Figure 1:** The most common words in the reviews text, showing that the word ‘game’ was highest. Positive words such as ‘great’, ‘fun’, ‘like’ and ‘good’ also appeared in the top 10.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The first method produced the top 40 collocations with the highest MI values. These pairs had a frequency of 1. They also had the same MI value, implying the method did not produce meaningful results.

The second method enlisted the modules in the NLTK library.

The code is presented in the screenshots below.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen shot of a code

Description automatically generated

The second method’s results gave the top 10 collocations presented in Figure 2 below. Seven of the top 10 collocations were actual video game titles. One of the popular collocations was the X Box gaming console.

A graph of different colored bars

Description automatically generated

**Figure 2:** The top 10 collocations/bigrams by Mutual Information.

An alternative to the MI calculation in the form of the Pointwise Mutual Information (PMI) produced different values, giving values like those observed in method 1.

The screenshots below reveal the steps taken.

A screenshot of a computer program

Description automatically generated

The PMI calculation was further modified to filter out collocations that appear less than three times, resulting in the top PMI values of 16.685. The screenshot below shows the calculation.

This tweak produced slightly meaningful collocations with examples, ‘Hulk Hogan’ and ‘Ted Diase’ who are actually WWF wrestlers, and ‘Flight Simulator’ and ‘Striker 1945’ which are game titles.

A screenshot of a computer

Description automatically generated

**3. Evaluation**

The application of NLP in this exercise aimed to generate customer insight by mining reviews text to give Amazon a commercial edge through a clear picture of the products that are popular. The first technique explored in question 3 produced a top 40 collocations list that included general expressions used in gaming and identified very few specific product names. The top 12 observed in the provided screenshot had only three actual gaming products, ‘X box’, ‘Grand theft’ and ‘Super Smash’. The MI technique used in question 4 was approached in two ways; mechanically looping through the data using the MI formula, and by using NLTK library modules. The former produced results that were not informative consisting of random paired noun collocations that appeared only once. The latter avenue gave the best outcome with a top 40 mainly consisting of specific video games and platforms alongside common expressions. This approach would be ideal for identifying which products Amazon should focus its stock budget on.

**4. References**

Icalem, 2018. *Mutual Information.* [online]. Available from: <https://lcalem.github.io/blog/2018/10/17/mutual-information> [Accessed 15 February 2024]

Medium, 2019. *Feature Engineering with NLTK for NLP and Python*. [online]. Available from: <https://towardsdatascience.com/feature-engineering-with-nltk-for-nlp-and-python-82f493a937a0> [Accessed 15 February 2024].

NLTK, 2023. *NLTK Documentation.* [online]. Available from: <https://www.nltk.org/_modules/nltk/tokenize/regexp.html> [Accessed 14 February 2024]