***CLICKBAIT GENERATOR***

***Importing the dataset and necessary libraries:***

***A screenshot of a computer

Description automatically generated with medium confidence***

>Accessed the dataset on sample clickbaits using open method of python.

>imported torch and nltk libraries of python.

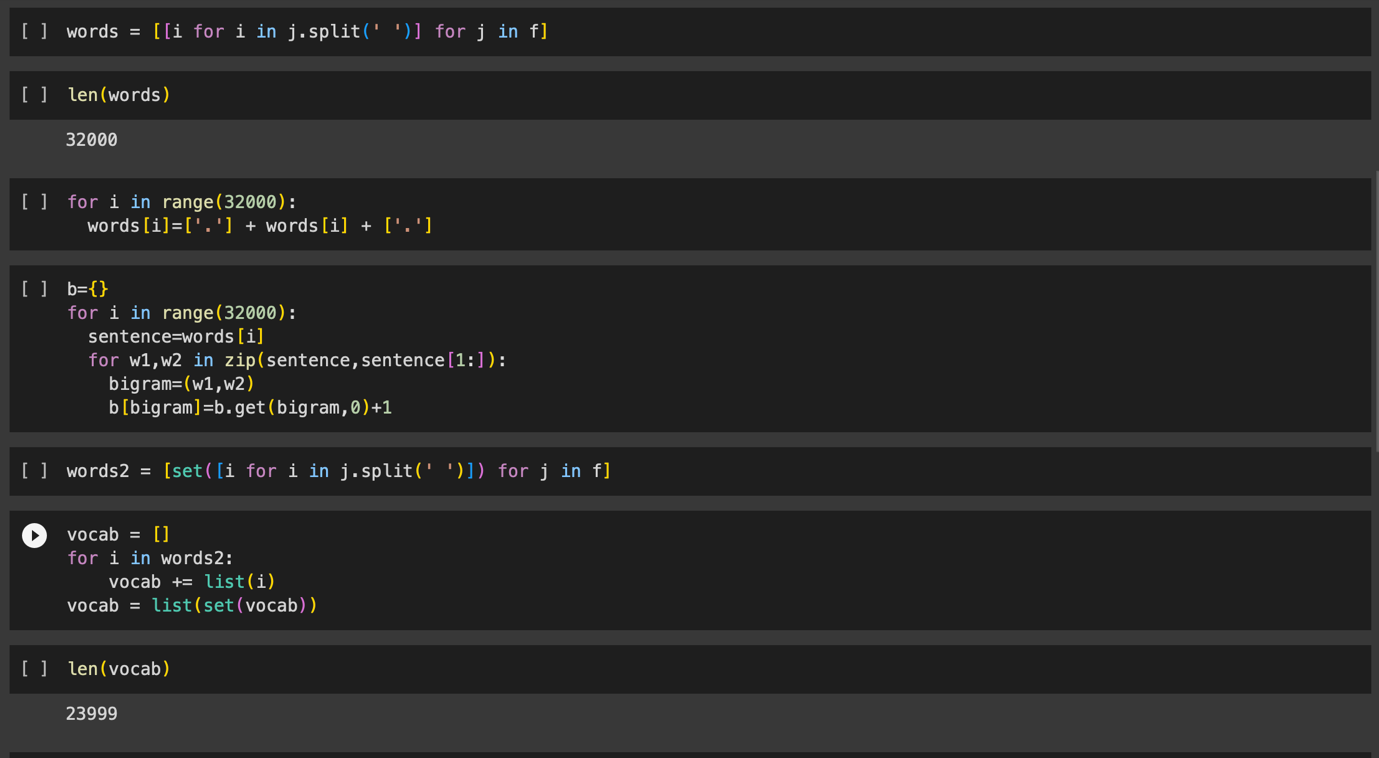
***Cleaning the dataset:***

***A screenshot of a computer program

Description automatically generated with low confidence***

>cleaned the dataset by keeping only alphabets and spaces and converted everything to lower case allowing case-insensitive analysis.

***Bigram Creation:***

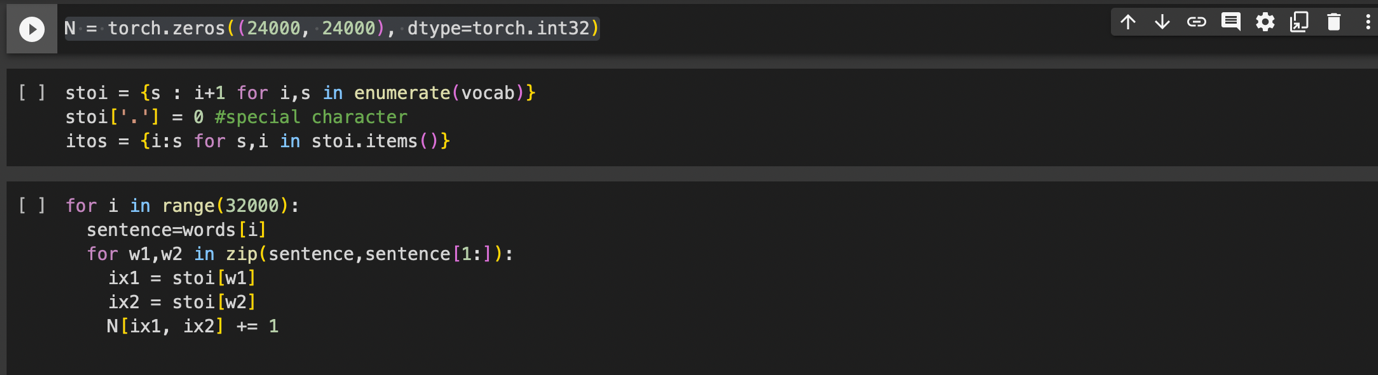


>created a list “words” containing all the sentences of f.

>created the bigrams and counted the total occurrence of each bigram.

>created a list vocab containing all the unique words present in the dataset.

***Tensor Creation:***

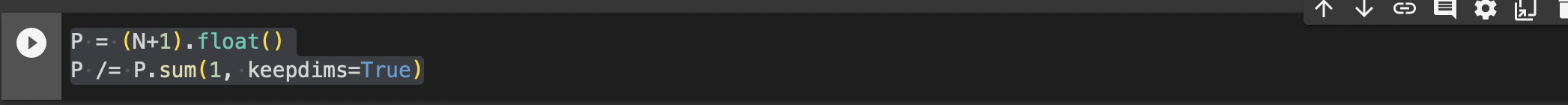


>created a tensor N with dimensions equal to the no. of unique words in dataset +1(for ‘.’ Indicating start or end of the sentence.)

>assigned an integer value to each word in vocab.

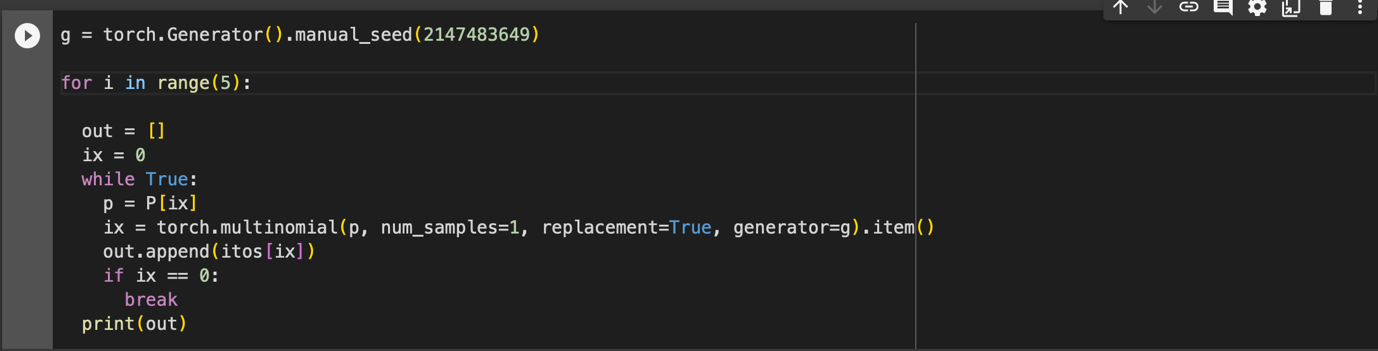
>populated the tensor with all possible bigrams and their total occurences.

***Normalization:***



>created a new tensor P which contains the normalized values where values of each row sums up to one, effectively creating a probability distribution.

***ClickBait Generation:***



>created a random number generator g with a given seed value which maintains the reproducibility of the random sequence generated.

>generates 5 sequence of words.

>initialized an empty list out to store the generated sequence of words.

>sampled a word index from the probability distribution p.

>appended the sampled word to out after converting word index to corresponding word.

>the loop breaks if the sampled word is 0 indicating the end of the sentence.