Lab Exercise Report on

Working of Bag-of-Words Model

Adikar Bharath N V S (171IT202)

Under the Guidance of,

Dr. Sowmya Kamath S.

Department of Information Technology, NITK Surathkal

Date of Submission: 15th October 2020

in partial fulfillment for the award of the degree

of

Bachelor of Technology

In

Information Technology

At

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

October 2020

Table of Contents

Chapter 1

Results

1.1 Introduction

Whenever an algorithm in NLP is applied, it works on a collection of numbers. Thus, we cannot feed a set of words, paragraphs, text documents etc. to the algorithm as input directly, and must be converted into a representation in the form of numbers. Hence, the Bag-of-Words model is used to convert any text into its namesake, which keeps a count of the total occurrences of the most frequently used words.

The working of the model can be visualized using a table, with a count of each word displayed.

Before displaying the working of the model, some preprocessing of the text is required.

1.2 Preprocessing

1.2.1 Stopword removal

Stopwords are words in a language that do not add any meaning or semantic information to a sentence. These can be removed while keeping the meaning of the sentence intact. For example, the English words

- and
- the
- but

are stopwords. Since the selection of stopwords is highly subjective, the 'stopwords' package in the nltk library is downloaded for use and applied on a paragraph as demonstrated below:

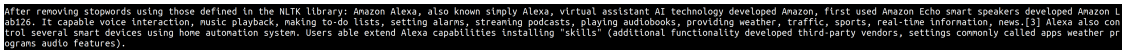

Figure 1.1: Stopword removal

1.2.2 Tokenization

Tokenization refers to the splitting up of words in a text into smaller words (for English), or even the creation of new words (in other languages). The NLTK library offers several ways of tokenization, such as:

- Sentence Tokenization: Splits text into sentences
- Word Tokenization: Splits text into words
- WordPunctuationTokenizer: Splits text into words as well as punctuation marks

Their working is demonstrated below:

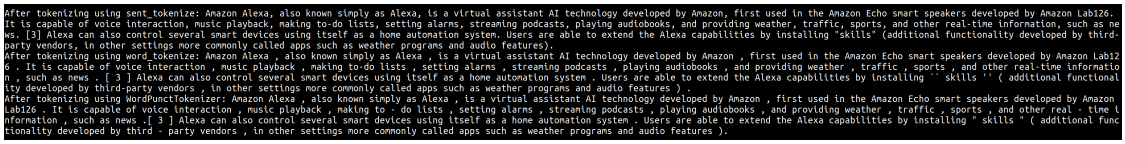

Figure 1.2: Tokenization

1.2.3 Stemming

Stemming refers to reduction of words into their root form. For example, the words do, doing, done all have the same root form 'do'. While processing text containing these words using NLP algorithms, they should be treated the same as they have the same meaning. Thus, it is imperative to reduce all words in a text to their root forms. The NLTK library again offers various stemmers such as:

- Porter Stemmer
- Lancaster Stemmer
- Snowball Stemmer (Supports multiple languages)

Their working is demonstrated below:

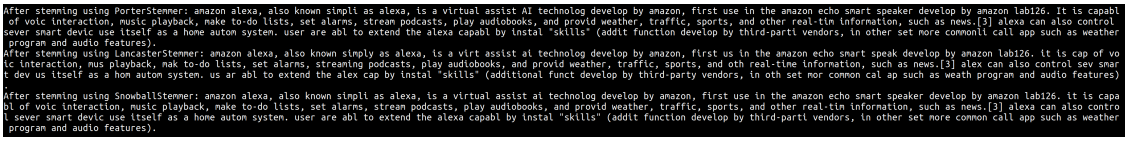

Figure 1.3: Stemming

1.2.4 Lemmatization

Lemmatization, similar to stemming, is the reduction of words into their base form. This base form is achieved by removing morphological differences and utilizing the vocabulary of the language. For example, the words good, better, best could all have the same base form 'good'. The same reasons for utilizing stemming apply to lemmatization, and stemming may not reduce words to forms that lemmatization does, as stemming uses the root form which is always part of the word. NLTK offers the WordNet Lemmatizer, whose working is demonstrated below:

Figure 1.4: Lemmatization

1.3 Vocabulary Construction

The BoW algorithm builds a model using a document-term matrix, constructed by considering the number of occurrences of each word in a document. Using this, each document can be represented as a weighted combination of various words in it. By setting a threshold and choosing words that carry more meaning, a histogram of all the words present in the documents in the form of a feature vector can be made.

Using the Count Vectorizer in the Scikit-learn library, a BoW model is constructed as demonstrated below:

Figure 1.5: BoW construction

1.4 N-gram modelling

An n-gram is a contiguous sequence of n items from a given sample of text or speech. These items can either be phonemes, syllables, letters, words, or base pairs depending on the application. N-grams are generally collected from a corpus containing text or speech. N-grams are used to generate language models that can predict which word comes next given a history of words.

Bigram, N-gram and Everygram modelling as part of the NLTK library is demonstrated below:

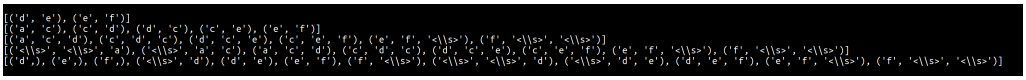

Figure 1.6: N-gram modelling

1.5 Text Corpora

Practical work in the field of NLP generally uses large bodies of linguistic corpora or data. Corpora are designed to carefully balance material across one or more genres.

Some of the corpora present in the NLTK library are (along with analysis demonstrated):

1.5.1 Gutenberg Corpus

NLTK includes a small selection of texts from the Project Gutenberg electronic text archive, which itself contains about 25,000 electronic books including established literature for free.

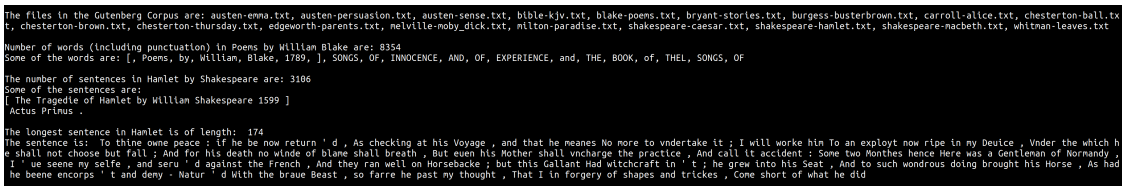

Figure 1.7: Gutenberg Corpus

1.5.2 Brown Corpus

NLTK also includes the Brown corpus, which was the worlds-first million word electronic corpus in English, created in 1961 at Brown university. The corpus contains material from over 500 sources, each categorized by genre such as news, editorial etc.

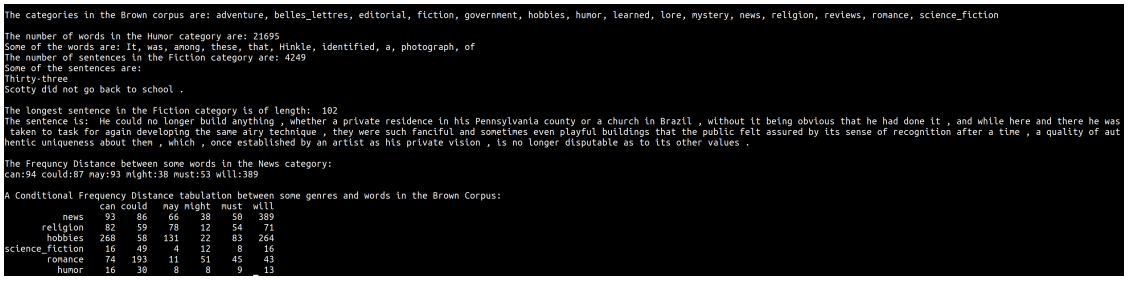

Figure 1.8: Brown Corpus