Lab 1

**Word Embedding:**

* Word embedding is a means of turning texts into numbers.
* We do this because machine learning algorithms can only understand numbers, not plain texts.
* In order for a computer to be able to read texts, they have to be encoded as a continuous vector of numeric values
* The simplest method is called one-hot encoding, also known as “1-of-N” encoding (meaning the vector is composed of a single one and a number of zeros).

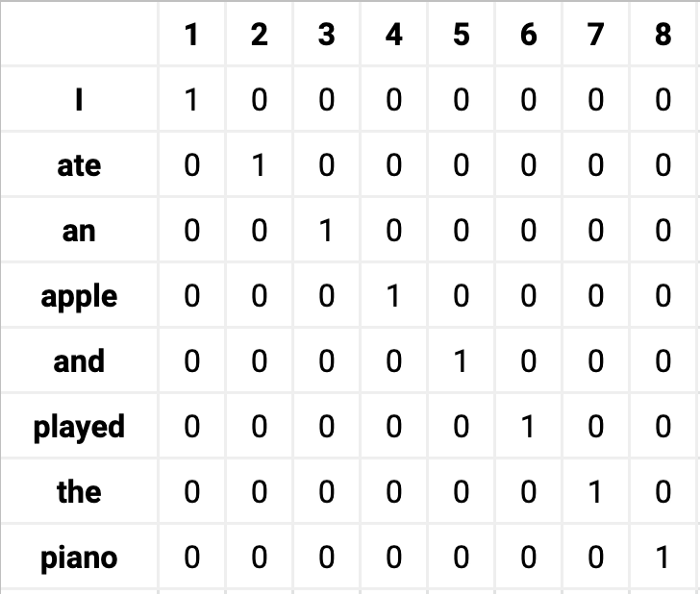
**One-Hot Encoding**

* Let’s take a look at the following sentence: “I ate an apple and played the piano.”
* We can begin by indexing each word’s position in the given vocabulary set.



* The word “I” is at position 1, so its one-hot vector representation would be [1, 0, 0, 0, 0, 0, 0, 0].
* Similarly, the word “ate” is at position 2, so its one-hot vector would be [0, 1, 0, 0, 0, 0, 0, 0].

The one-hot embedding matrix for the example text would look like this:



**Ex 1. Represent following sentence in one-hot encoding**

“ I will eat the Pizza”.

Using Numpy

***Steps to follow:***

1. Convert Text to lower case
2. Tokenize the text
3. Get unique words
4. Sort the word list
5. Get the integer/position of the words
6. create a vector of each word by marking its position as 1 and rest as 0
7. Create a matrix of the found vectors.

**Bag of Words**

* Bag of words is a [Natural Language Processing](https://www.mygreatlearning.com/blog/natural-language-processing-tutorial/) technique of text modeling
* it is a method of feature extraction with text data.
* **What**
* A bag of words is a representation of text that describes the occurrence of words within a document.
* It is called a “bag” of words because any information about the order or structure of words in the document is discarded
* **Why**
* One of the biggest problems with text is that it is messy and unstructured,
* [machine learning](https://www.mygreatlearning.com/blog/what-is-machine-learning/?highlight=what%20is%20machine%20learning) algorithms prefer structured, well defined fixed-length inputs
* Bag-of-Words technique we can convert variable-length texts into a fixed-length **vector.**
* **Example(1) without preprocessing:**

Sentence 1:  ”Welcome to Great Learning, Now start learning”

Sentence 2: Learning is a good practice”

* Step 1: Go through all the words in the above text and make a list of all of the words in our model vocabulary.
* Welcome
* To
* Great
* Learning
* ,
* Now
* start
* learning
* is
* a
* good
* practice

Sentence scoring

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sentence** | Welcome | to | Great | Learning | , | Now | start | learning | is | a | good | practice |
| **Sentence1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **Sentence2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

* **Example(2) with preprocessing:**
* **Step 1**: Convert the above sentences in lower case as the case of the word does not hold any information.
* **Step 2**: Remove special characters and stopwords from the text. Stopwords are the words that do not contain much information about text like ‘is’, ‘a’,’the and many more’.
* **Step 3**: Go through all the words in the above text and make a list of all of the words in our model vocabulary.
* **Sentence Scoring**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sentence** | welcome | great | learning | now | start | good | practice |
| **Sentence1** | 1 | 1 | 2 | 1 | 1 | 0 | 0 |
| **Sentence2** | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

**Ex 2 Create a Bag of Words Model with Sklearn**

use the **CountVectorizer()** function from the [Sk-learn library](https://www.mygreatlearning.com/blog/21-open-source-python-libraries/?highlight=python)

## 3. **Tf-Idf (term frequency-inverse document frequency)**