Aysel Aydin

Summary

The article explains the Bag of Words (BoW) model in Natural Language

4 — Bag of Words Model in NLP

have previously covered in the field of NLP, [illegible]

Bag of Words (BoW) is a Natural Language Processing strategy for converting a text document into numbers that can be used by a computer program. This method involves converting text into a vector based on the frequency of words in the text, without considering the order or context of the words.

Let's examine the example we gave for tokenization in our previous article with BoW.

Imagine a social media platform that aims to analyze customer reviews and understand the popularity of services among users. This platform decides to employ the **Bag of Words** method for processing customer reviews.

Data Collection: The first step involves collecting and storing customer reviews, which consist of text written by customers about various services.

Preprocessing: Text data is cleaned by removing punctuation marks, numbers and unnecessary whitespace.

Creating a Word List: A word list is created for BoW. This list includes all the unique words in the dataset.

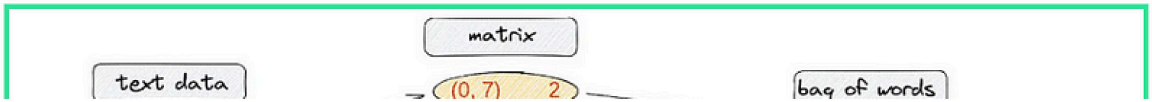
Text Representation: Each customer review is represented using the BoW method. The frequency of each word is recorded within a vector based on its position in the word list. For example, the BoW representation for the phrase **“great service”** could be as follows: [service: 1, great: 1, other_words: 0].

Analysis and Classification: With this representation method, the platform can analyze how popular services are among customers and identify which services receive positive or negative reviews. For instance, if a service’s BoW representation frequently includes positive terms like “**high quality**” and “**affordable**,” it can be inferred that the service receives positive feedback.

Improvement: Based on the results obtained, the platform can take steps to optimize its service portfolio and enhance the overall customer experience.

In this way, the BoW method enables the social media platform to analyze customer reviews, monitor service performance and make improvements

Let’s apply the above steps to a sample text.



Firstly, let’s write a function to pre-process our text. We removed stop words, emojis, numbers, punctuation marks, and excess spaces from the sentence and converted all characters to lowercase with this function.

```
import nltk
from nltk.stem import WordNetLemmatizer
import re
from nltk.corpus import stopwords

def preprocessing_text(text):
    lemmatizer = WordNetLemmatizer()
    emoji_pattern = r'^(?:[\u2700-\u27bf]|(?:\ud83c[\udde6-\uddff]){1,2}|(?:\

    text = text.split()
    text = [lemmatizer.lemmatize(word) for word in text if not word in set(st
```

```
    text = re.sub(emoji_pattern, '', text)
    text= re.sub(r'\s+', ' ', text)
    text= text.lower().strip()

    return text

paragraph = """I am really disappointed this product. I would not use it agai
I love this product! It has some good features"""

sentences_list = nltk.sent_tokenize(paragraph)

corpus = [preprocessing_text(sentence) for sentence in sentences_list]

print(corpus)
```

Output:

```
[
'i really disappointed product',
'i would use again',
'it really bad feature',
'i love product',
'it good feature'
]
```

Now, we will create a Bag of Words model using the **count vectorizer** function available in **sklearn**.

```
vectorizer = CountVectorizer()

X = vectorizer.fit_transform(corpus)

feature_names = vectorizer.get_feature_names_out()

X_array = X.toarray()

print("Unique Word List: \n", feature_names)
print("Bag of Words Matrix: \n", X_array)
```

```
Unique Word List:
['again' 'bad' 'disappointed' 'feature' 'good' 'it' 'love' 'product'
 'really' 'use' 'would']

Bag of Words Matrix:
[[0 0 1 0 0 0 0 1 1 0 0]
 [1 0 0 0 0 0 0 0 0 1 1]
 [0 1 0 1 0 1 0 0 1 0 0]
 [0 0 0 0 0 0 1 1 0 0 0]
 [0 0 0 1 1 1 0 0 0 0 0]]
```

Let’s create a dataframe and show the result visually.

```
import pandas as pd

df = pd.DataFrame(data=X_array, columns=feature_names, index=corpus)
```

	again	bad	disappointed	feature	good	it	love	product	really	use	would
i really disappointed product	0	0	1	0	0	0	0	1	1	0	0
i would use again	1	0	0	0	0	0	0	0	0	1	1

Conclusion

Through this article, we have learned about the bag of words.

In summary, Bag of Words used to convert words in a text into a matrix representation by extracting its features, it shows us which word occurs in a sentence and its frequency, for use in modeling such as machine learning algorithms.

In the next article, we will cover the **TF-IDF** topic.

I hope it will be a useful article for you. If you stayed with me until the end, thank you for reading! Happy coding ✊

Contact Accounts: Twitter, LinkedIn

Recommended from ReadMedium

Dr. Walid Soula

Bag-of-Words

Explore the fundamentals of the Bag-of-Words model in natural language processing

4 min read

Okan Yenigün

NLTK #2: Text Corpora

Accessing Text Corpora and Lexical Resources

9 min read

Emad Dehnavi

Understanding Sentence Similarity in NLP: Top 3 Models You Should Know

Sentence Similarity, is a specific task within field of Natural Language Processing (NLP) that involves assessing how similar two sentences...

2 min read

6—Creating a Word Cloud using TF-IDF in Python

In this article, we will cover creating a word cloud using TF-IDF in Python. Before we start, I recommend you read the article I have...

3 min read

Yanwei Liu

The Evolution of Natural Language Processing: From N-Grams to GPT

Comprehensive Analysis of Key Techniques and Architectures Driving Advances in NLP, including Word Embeddings, Recurrent Neural Networks...

10 min read

Anmol Talwar

CBOW—Word2Vec

Continuous Bag of Words (CBOW) is one of the architectures used in the Word2Vec framework for learning word embeddings. CBOW is designed...

4 min read

J

Jo Wang

Deep Learning Part 5 -How to prevent overfitting

Techniques used to prevent overfitting in deep learning models:

Aysel Aydin

9—Understanding Word Embeddings in NLP

In this article, we will talk about word embedding and techniques, their usage areas.

3 min read

Rahul Kumar

NLP Hands-On with Text Classification

This post is a part of the NLP Hands-on series and consists of the following tasks: 1. Text Classification 2. Token Classification 3...

3 min read

Mdabdullahalhasib

A Complete Guide to Embedding For NLP & Generative AI/LLM

Understand the concept of vector embedding, why it is needed, and implementation with LangChain.

11 min read

Ajay Halthor

Word2Vec, GloVe, and FastText, Explained

How computers understand words

John Vastola

10 Must-Know Machine Learning Algorithms for Data Scientists

Machine learning is the science of getting computers to act without being explicitly programmed.” — Andrew Ng

7 min read