

Bag of Words

A good property to extract from a text corpus is the frequency of words. In Andrew Trask's lesson, he took the frequency of words for negative and positive movie reviews. This representation of the data is known as Bag of Words.

A bag of words is usually represented as a dictionary. The key is the word and the value is the number of times a word appears in the text. For example, the sentence "the fox jumps over the lazy dog" would become:

{'the': 2, 'jumps': 1, 'lazy': 1, 'over': 1, 'fox': 1, 'dog': 1} Limitations However, bag of words doesn't keep information related to the order of words which is problematic for our understanding. For example, can you derive the meaning from the following bag of words?

{'take': 2, 'I': 2, 'for': 2, 'do': 1, 'them': 1, 'neck': 1, 'head': 1, 'the': 1}

By losing the ordering of the words, we lose important context that is important to understanding. Keep this in mind as you use this technique for sentiment analysis.

Quiz

In this quiz, you'll implement the bag of words algorithm. Write your implementation in the `bag_of_words` function. Return a dictionary with the key as the word and value as the number of times the word appears in the parameter text.

In []:

```
from collections import Counter
```

```
def bag_of_words(text):  
    # TODO: Implement bag of words  
    return Counter(text.split())
```

```
test_text = 'the quick brown fox jumps over the lazy dog'
```

```
print(bag_of_words(test_text))
```

```
Counter({'the': 2, 'dog': 1, 'quick': 1, 'lazy': 1, 'over': 1, 'fox': 1, 'brown': 1, 'jumps': 1})
```