28.4 Bag of Words (BOW)

The below techniques are used to convert a text to a vector.

1) Bag of Words (BOW)
2) Term Frequency - Inverse Document Frequency (TF-IDF)
3) Average Word2Vector
4) TF-IDF Weighted Average Word2Vector

Bag of Words (BOW)

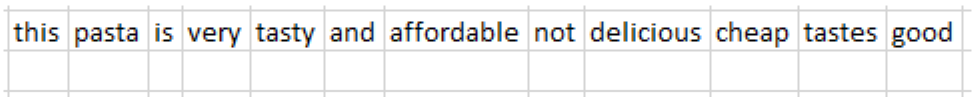
Consider the below 4 reviews as an example.

r_1: this pasta is very tasty and affordable
r_2: this pasta is not tasty and is affordable
r_3: this pasta is delicious and cheap
r_4: pasta is tasty and pasta tastes good

Steps in Bag of Words

1) Construct a dictionary (not Python data structure dictionary) which is going to be a set of all the words that are present in all the reviews. For this example, all the unique words present in all the reviews are to be added to a new set. (All these words should be taken only once)
2) Construct a vector for every review. In this problem, we are using the word "review", but the official terminology in NLP is a "**document**".
   For example, we have to construct the vector 'V_1' from the review 'r_1'. Let us assume we have 'n' documents/reviews and the total number of unique words in all the 'n' documents is 'd'.
   We then have to construct a d-dimensional vector for every document. The vector has to be initialized with '0' in all dimensions and each cell is associated with each dimension and has to be filled with the number of times the corresponding word occurs in the given document/review. Each word is considered as a different dimension here.

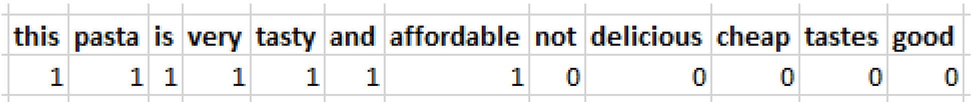
Let us look at converting the above given 4 reviews into vector form. We'll have the dimensions as shown below.



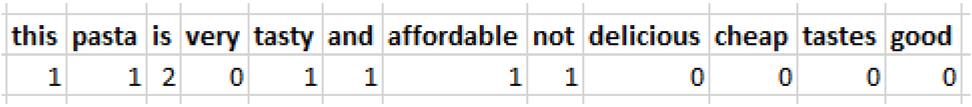
this	pasta	is	very	tasty	and	affordable	not	delicious	cheap	tastes	good

All these features/words are present across the corpus. Now we shall vectorize the given 4 reviews according to these dimensions.

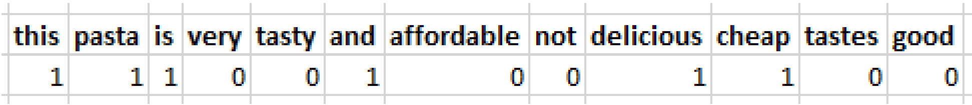
AppliedRoots (Draft Copy)

r1:


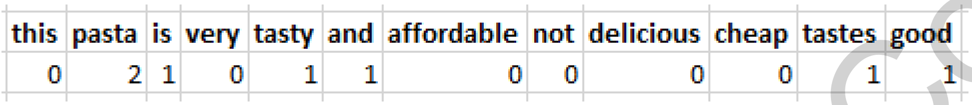
this	pasta	is	very	tasty	and	affordable	not	delicious	cheap	tastes	good
1	1	1	1	1	1	1	0	0	0	0	0

r2:


this	pasta	is	very	tasty	and	affordable	not	delicious	cheap	tastes	good
1	1	2	0	1	1	1	1	0	0	0	0

r3:


this	pasta	is	very	tasty	and	affordable	not	delicious	cheap	tastes	good
1	1	1	0	0	1	0	0	1	1	0	0

r4:


this	pasta	is	very	tasty	and	affordable	not	delicious	cheap	tastes	good
0	2	1	0	1	1	0	0	0	0	1	1

In the above vectors of the reviews, the numbers denote the number of occurrences of that particular word in that review.
Note: The collection of all the documents(here reviews) is called **Corpus**.
The main objective of bag of words is that if documents are more similar semantically, then their corresponding vectors will be closer.
The result of Bag of Words is always a **Sparse Vector**.

Sparse Vector and Dense Vector

A vector is said to be a sparse vector, if most of the dimensions in it have 0 as value.

A vector is said to be a dense vector, if most of the dimensions in it have non zero values.

Note: Let us calculate the length of the difference between two vectors.
length(V_1-V_2) = $||V_1-V_2||$ = sqrt($(2-1)^2 + (0-1)^2 + (1-0)^2$) = sqrt(1 + 1 + 1) = sqrt(3)

Note: One of the disadvantages with the Bag of Words approach is that, even after vectorization, if we observe a difference of a few bits between the two given reviews, but still the semantic meanings could be quite opposite. For example, if we clearly observe the reviews 'r_1' and 'r_2', we see a difference of only two words among them, but their semantic meanings are quite opposite.

Variations of Bag of Words

The so far discussed approach of Bag of Words is also known as **Count Bag of Words**. There is another variation of Bag of Words, which is called **Binary Bag of Words** (or) **Boolean Bag of Words**.

In the Binary/Boolean Bag of Words, unlike the Count Bag of Words, we do not have the count of occurrences of each word in the corpus as a dimension magnitude in

AppliedRoots (Draft Copy)

their vectors. Here it just indicates whether the words in the corpus occur in the documents or not in the form of ‘1’ and ‘0’.

In the case of the Binary Bag of Words, the distance between the vectors is the square root of the total number of different values in the magnitude of each dimension in both the vectors.

length(V_1-V_2) = ||V_1-V_2|| = sqrt(number of different values in the magnitudes of each dimension)

Note: If there is no duplicate word occurring in a vector form among all the reviews/documents in the corpus, then both Count Bag of Words and Binary Bag of Words will give the same performance and the same result.

When to choose Count Bag of Words and when to choose Binary Bag of Words?

Choosing one among the two is problem specific. In some contexts, the information of presence or absence of a word is sufficient for the model to work well. In such a case, Binary Bag of Words is a good choice.

In other contexts, the count of the number of times a word occurs matters a lot. Here the Count based Bag of Words carries more information than the Binary Bag of Words. So the count Bag of Words typically tends to outperform the Binary Bag of Words.

If we want to determine if a text review is “positive” or “negative”, the presence of words like “bad” or “terrific” is good enough to predict the class label. This context is one of the best examples where Binary BOW based encoding can be used.

On the other hand, if we have more classes like “Very Positive”, “Positive”, “Negative”, “Very Negative”, then the number of times words like “terrific”, “bad”, “great”, “terrible” occur could help us determine the extent of positivity and negativity helping us classify better. This context is one of the best examples where Count BOW based encoding can be used.

AppliedRoots (Draft Copy)