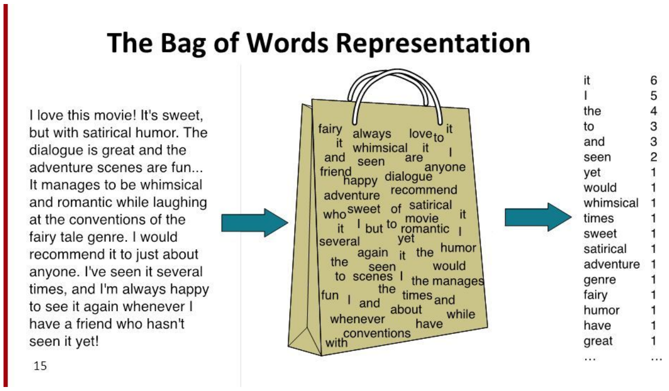
1 What is Bag-of-Words?

We need a way to represent text data for machine learning algorithm and the bag-of-words model helps us to achieve that task.

The bag-of-words model is simple to understand and implement. In this model a text (such as a sentence or a document) is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity.

The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier.

A very common feature extraction procedures for sentences and documents is the bag-of-words model (BOW). In this approach, we look at the histogram of the words within the text, i.e. considering each word count as a feature.



The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

15

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

2 Example of the Bag-of-Words Model

Let's take an example to understand the Bag-of-Words.

- "It was the best of times"
- "It was the worst of times"
- "It was the age of wisdom"
- "It was the age of foolishness"

We treat each sentence as a separate document and we make a list of all words from all the four documents excluding the punctuation. The unique words we get are :

'It', 'was', 'the', 'best', 'of', 'times', 'worst', 'age', 'wisdom', 'foolishness'

The next step is to create vectors. Vectors convert text that can be used by the machine learning algorithm. The simplest method is to mark the presence of words as a boolean value, 0 for absent, 1 for present. We take the first document-"It was the best of times" and we check the frequency of words from the 10 unique words.

“it” = 1
“was” = 1
“the” = 1
“best” = 1
“of” = 1
“times” = 1
“worst” = 0
“age” = 0
“wisdom” = 0
“foolishness” = 0

Rest of the documents will be:
“It was the best of times” = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
“It was the worst of times” = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
“It was the age of wisdom” = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
“It was the age of foolishness” = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

In this approach, each word or token is called a “gram”. Creating a vocabulary of two-word pairs is called a bigram model. For example, the bigrams in the first document : “It was the best of times” are as follows:
“it was”
“was the”
“the best”
“best of”
“of times”

The process of converting text into numbers is called vectorization in ML. Different ways to convert text into vectors are:

- Counting the number of times each word appears in a document.
- Calculating the frequency that each word appears in a document out of all the words in the document.

3 TF-IDF Vectorizer

In practice the Bag-of-Words is mainly used as a tool of feature generation. After transforming the text into a ”bag of words”, we can calculate various measures to characterize the text.

TF-IDF stands for term frequency-inverse document frequency. TF-IDF weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

- **Term Frequency (TF)**: is a scoring of the frequency of the word in the current document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. The term frequency is often divided by the document length to normalize.

$$\text{TF(t)} = \frac{Number\,of\,times\,term\,t\,appears\,in\,a\,document}{Total\,number\,of\,terms\,in\,the\,document}$$

However, term frequencies are not necessarily the best representation for the text. Common words like ”the”, ”a”, ”to” are almost always the terms with highest frequency in the text. Thus, having a high raw count does not necessarily mean that the corresponding word is more important. To address this problem, one of the most popular ways to ”normalize” the term frequencies is to weight a term by the inverse of document frequency

- **Inverse Document Frequency (IDF)**: is a scoring of how rare the word is across documents. IDF is a measure of how rare a term is. Rarer the term, more is the IDF score.

$$\text{IDF(t)} = \log_e(\frac{Total\,number\,of\,documents}{Number\,of\,documents\,with\,term\,t\,in\,it})$$

Thus,

$$\text{TF - } IDF_Score = \text{TF * IDF}$$