1. Explain One-Hot Encoding

One Hot Encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. One hot encoding is the most widespread approach, and it works very well unless your categorical variable takes on a large number of values (i.e. you generally won't it for variables taking more than 15 different values. It'd be a poor choice in some cases with fewer values, though that varies.) One hot encoding creates new (binary) columns, indicating the presence of each possible value from the original data.

1. Explain Bag of Words

The bag-of-words (BOW) model is a representation that turns arbitrary text into fixed-length vectors by counting how many times each word appears. This process is often referred to as vectorization.

Suppose we wanted to vectorize the following:

the cat sat

the cat sat in the hat

the cat with the hat

We’ll refer to each of these as a text document.

We first define our vocabulary, which is the set of all words found in our document set. The only words that are found in the 3 documents above are: `the, cat, sat, in, the, hat, and with`.

To vectorize our documents, all we have to do is count how many times each word appears:

Now we have length-6 vectors for each document

`the cat sat: [1, 1, 1, 0, 0, 0]`

`the cat sat in the hat: [2, 1, 1, 1, 1, 0]`

`the cat with the hat: [2, 1, 0, 0, 1, 1]`

The Problem with BOW is, it does not preserve contextual information among the words. We lose contextual information, e.g. where in the document the word appeared, when we use BOW. It’s like a literal bag-of-words: it only tells us what words occur in the document, not where they occurred.

1. Explain Bag of N-Grams

A Bag-of-N-Grams model is a way to represent a document, similar to a bag-of-words terms bag-of-words model.

A bag-of-n-grams model represents a text document as an unordered collection of its n-grams.

For example, let’s use the following phrase and divide it into bi-grams (n=2).

James is the best person ever.

becomes

James, James is, is the, the best, best person, person ever., ever.

In a typical bag-of-n-grams model, these 6 bigrams would be a sample from a large number of bigrams observed in a corpus. And then James is the best person ever. would be encoded in a representation showing which of the corpus’s bigrams were observed in the sentence.

A bag-of-n-grams model has the simplicity of the bag-of-words model, but allows the preservation of more word locality information.

1. Explain TF-IDF

TF-IDF stands for “Term Frequency – Inverse Document Frequency.” It reflects how important a word is to a document in a collection or corpus. This technique is often used in information retrieval and text mining as a weighing factor.

TF-IDF is composed of two terms:

Term Frequency (TF): The number of times a word appears in a document divided by the total number of words in that document.

Inverse Document Frequency (IDF): The logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

The TF-IDF value increases as the word’s frequency in a document (TF) increases. However, this is offset by the number of times the word appears in the entire collection of documents or corpus (IDF).

1. What is OOV problem?

Out-Of-Vocabulary (OOV) words is an important problem in NLP, we will introduce how to process words that are out of vocabulary in this tutorial.

We often use word2vec or glove to process documents to create word vector or word embedding.

However, we may ignore some words that appear rarely in documents, which may cause OOV problem.

Meanwhile, we may use some pre-trained word representation file, which may do not contain some words in our data set. It also can cause OOV problem.

1. What are word embeddings?

Word embedding is one of the most popular representation of document vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. They are vector representations of a particular word.

1. Explain Continuous bag of words (CBOW)

The word2vec model has two different architectures to create the word embeddings. They are:

Continuous bag of words(CBOW)

Skip-gram model

Continuous bag of words(CBOW):

The CBOW model tries to understand the context of the words and takes this as input. It then tries to predict words that are contextually accurate. Let us consider an example for understanding this. Consider the sentence: `It is a pleasant day` and the word `pleasant` goes as input to the neural network. We are trying to predict the word `day` here. We will use the one-hot encoding for the input words and measure the error rates with the one-hot encoded target word. Doing this will help us predict the output based on the word with least error.

1. Explain SkipGram

The Skip-gram Model:

The Skip-gram model architecture usually tries to achieve the reverse of what the CBOW model does. It tries to predict the source context words (surrounding words) given a target word (the center word). Considering our simple sentence from earlier, “the quick brown fox jumps over the lazy dog”. If we used the CBOW model, we get pairs of (context\_window, target\_word)where if we consider a context window of size 2, we have examples like ([quick, fox], brown), ([the, brown], quick), ([the, dog], lazy) and so on. Now considering that the skip-gram model’s aim is to predict the context from the target word, the model typically inverts the contexts and targets, and tries to predict each context word from its target word. Hence the task becomes to predict the context [quick, fox] given target word ‘brown’ or [the, brown] given target word ‘quick’ and so on. Thus the model tries to predict the context\_window words based on the target\_word.

1. Explain Glove Embeddings.

GloVe word is a combination of two words- Global and Vectors. In-depth, the GloVe is a model used for the representation of the distributed words. This model represents words in the form of vectors using an unsupervised learning algorithm. This unsupervised learning algorithm maps the words into space where the semantic similarity between the words is observed by the distance between the words. These algorithms perform the Training of a corpus consisting of the aggregated global word-word co-occurrence statistics, and the result of the training usually represents the subspace of the words in which our interest lies.