

Ayesha Ashraf

Task 31

Word Embedding and One Hot Encoding

One Hot Encoding and Word Embedding are two of the most popular concept for vector representation in Natural Language Processing. Even though both have their own pros and cons, they tend to work better on different types of problems.

One Hot encoding is a representation of categorical variables as binary vectors. Each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1. Here is a One Hot vector representation:

Rome = $[1, 0, 0, 0, 0, 0, \dots, 0]$

Paris = $[0, 1, 0, 0, 0, 0, \dots, 0]$

Italy = $[0, 0, 1, 0, 0, 0, \dots, 0]$

France = $[0, 0, 0, 1, 0, 0, \dots, 0]$

word V = $[0, 0, 0, 0, 0, 0, \dots, 1]$

From the above, we can see every word has its own value in a vector. It is easy to implement and can work really fast, but in this process, it loses the inner meaning of the word in a sentence. Thus it loses the context of the sentence. Because of this one hot encoding is not widely used in many natural language processing applications.

On the other hand, word embedding takes context into account and gives word with similar meaning or influence in a sentence similar value for a specific feature. Here is a representation:

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	<u>0.93</u>	<u>0.95</u>	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97

One hot encoding

From the above image, we can see different words have almost similar value on different features(here Gender, Royal, Age, and Food are features). Because of this, word embedding can hold the context of the sentence from the words and after training on large data set it can even recognize words that are not available in vector representation from sentences. In that way, we can use a model trained on a large number of datasets on a relatively small unlabeled dataset. Thus word embedding can be used for transfer learning. Because of this property, word embedding has been very useful in a wide range of applications like named entity recognition, text summarization, co-reference resolution, and parsing. On the other hand, word embedding does not work exceptionally well in applications that require a lot of data dedicated to that task like language modeling, machine translation.

Overall word embedding can be more useful when we have a relatively small unlabeled dataset then we can use a model trained on a larger dataset and use transfer learning. But it is more useful to use One Hot vector in tasks where we have dedicated large datasets because word embedding does not work exceptionally well on those tasks but is more complicated to implement and it is also computationally more expensive than One Hot vector. Then again, the most effective way is to test both methods(if you have the time, patience and resources) for any specific natural language processing task.