28.9 Word2Vec

We have seen the BOW and TF-IDF techniques for converting a text into a vector. But these techniques do not take the semantic meanings into consideration, whereas Word2Vec is a state of the art technique used to convert the text into a vector, and also takes the semantic meaning into consideration.

So far in BOW/TF-IDF, we have seen a text is given as an input and the output is a sparse vector. But Word2Vec takes a word as an input and gives a d-dimensional vector as an output which is dense.

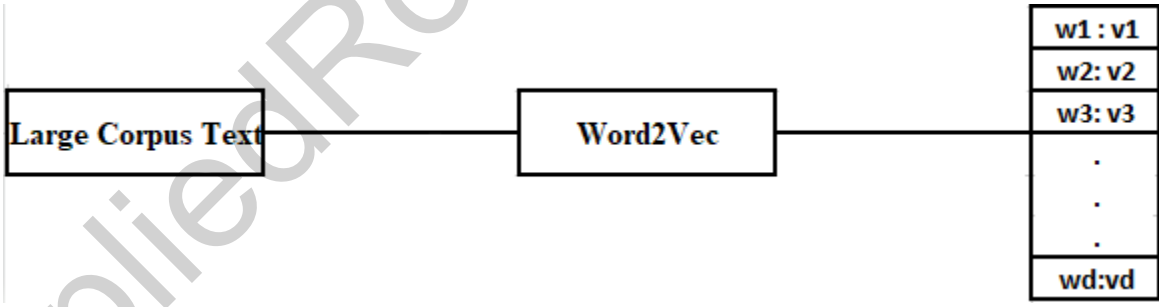
If the words 'w_1' and 'w_2' are semantically similar, then their vectors 'v_1' and 'v_2' are closer. This is the principle, Word2Vec tries to achieve. Word2Vec also tries to satisfy the relationships between the words.

For example, if we have the words 'w_1', 'w_2', 'w_3' and 'w_4' as

w_1 = "man"
w_2 = "woman"
w_3 = "king"
w_4 = "queen"

Here the vector difference (V_{man} - V_{woman}) is parallel to (V_{king}-V_{queen}). This is the relationship it holds. Here it is a male-female relationship. Similarly, it holds country-capital, country-currency relationships, etc. It also holds verb-tense relationships. All these relationships are learnt by Word2Vec automatically from the raw text, without being explicitly programmed.

Top Level View of how Word2Vec works



Here a large corpus text is given as an input to Word2Vec and then Word2Vec creates a vector for each word. These vectors are high-dimensional. The more the dimensions we have in our vectors, the more rich the information is going to be. In order to get as many dimensions in the vectors, we need to give as much large corpus text as input.

In the core nutshell, for every word, the Word2Vec checks for the neighborhood of that word and if the neighborhood of this word is similar to the neighborhood of other words, then the vector of this word and other words is similar.

If we have two words 'w_i' and 'w_j', then if the neighborhood of the word 'w_i' is the same as the neighborhood of 'w_j', then the vectors of 'wi' and 'wj' are similar.

So far in BOW and TF-IDF we have converted documents into vectors. In Word2Vec, we are converting each word of the corpus into a vector.

Note:

Word2Vec could not give the correct results for stemmed words. For example, for words like 'tasti', we do not get appropriate results as the stemmed words are not present in the text directly. So we should not perform stemming on the data if we want to go for Word2Vec.