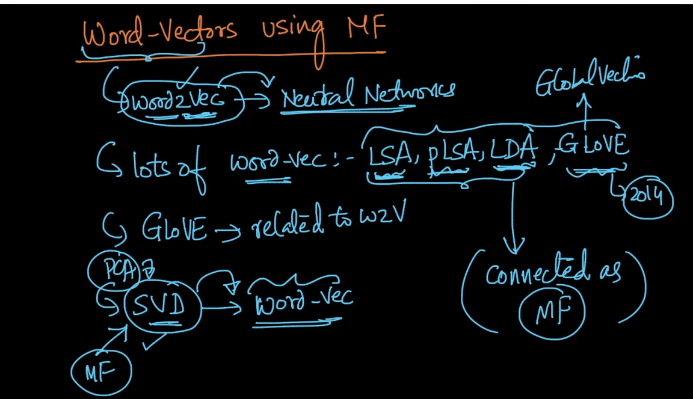
So Word Vectors which we generated using word2vec can also be generated using MF.

So there are multiple word vectors shown in below image but for our understanding we will use SVD which we know is a form of MF to obtain Word Vectors and Word Vectors generated using SVD are very powerful, of course not as powerful as word2vec but are powerful..

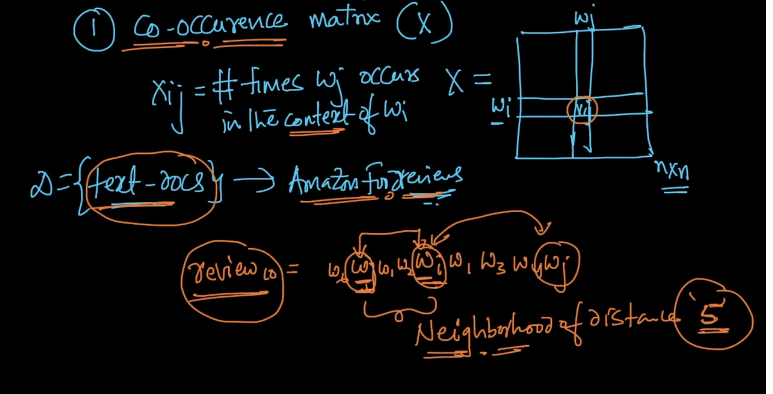


Co-Occurrence Matrix :

It is nothing but the matrix that store the occurrence count between any two entity in matrix as shown in below image.

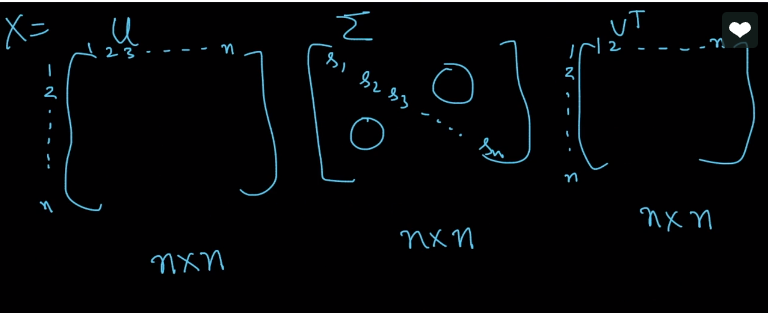
Two words can be said in context to each other only if they occur within specified distance from each other.

For any review in Amazon Food review dataset we can say W3 is neighbour of W5 if it occurs within distance of 5 words (either side) which is also shown in below image.



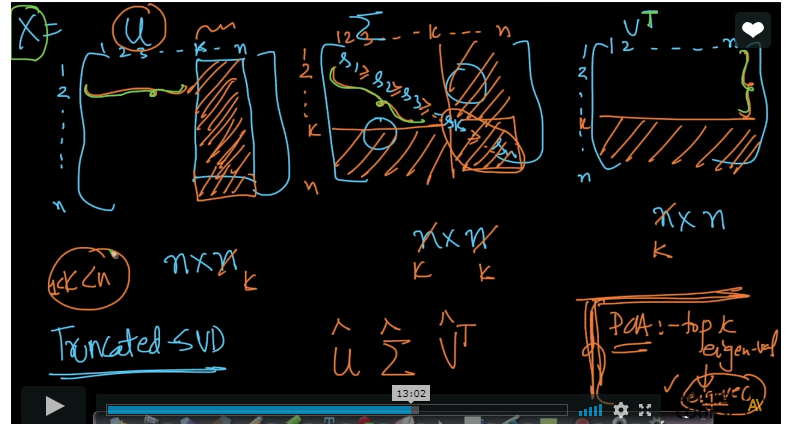
Co-Occurrence matrix is n\*n matrix where n is number of words.

Now our second step is what we saw in SVD.



So this is what we get when we perform SVD.

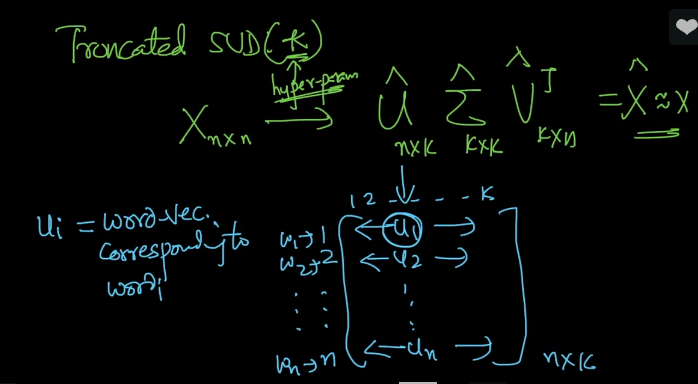
Now there is another concept like truncated SVD in which we just truncate number of words as shown in below image.



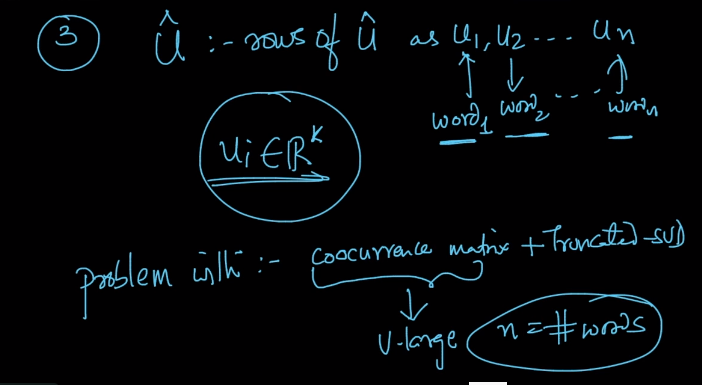
So this is much similar to PCA as we do in PCA i.e. we take top k eigen values, so similarly we are taking top k singular vectors and top k left eigen vectors and top k right eigen vectors.

So when we do truncated SVD we get X(hat) which is very nice approximation to X.

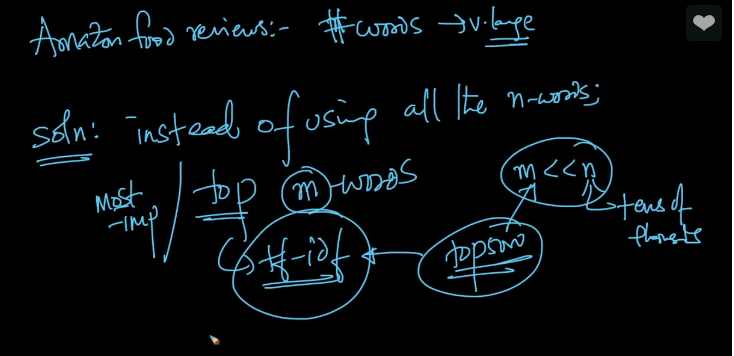
And it looks like this



And this we can say Ui is the word corresponding to word(i).

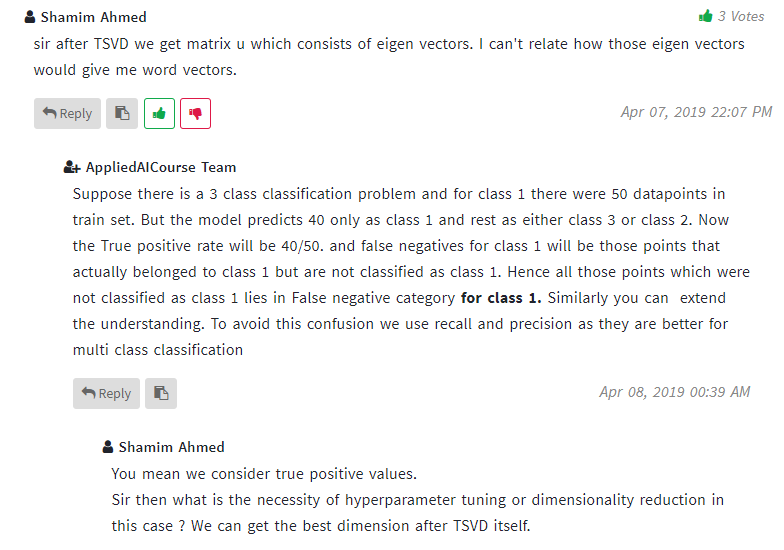


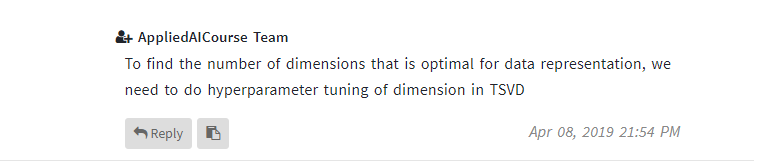
In step 3 we get a problem with co-occurrence matrix + truncated SVD because n is very large and so what people do is they take to m words instead of using n words and they do it by using tfidf.

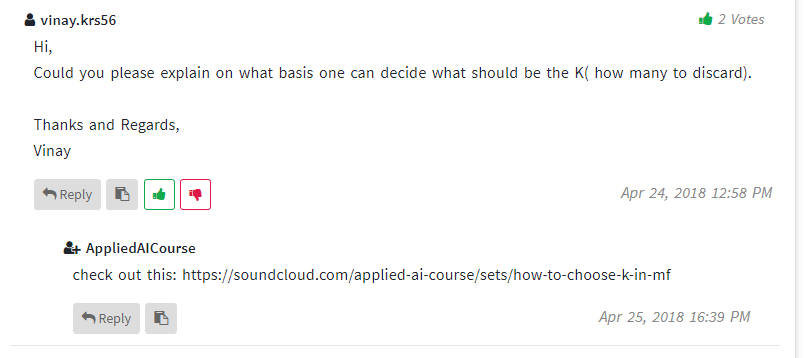


In this way we get our word vectors.

**Comments:**







Link to above questions answer:  <https://soundcloud.com/applied-ai-course/sets/how-to-choose-k-in-mf>