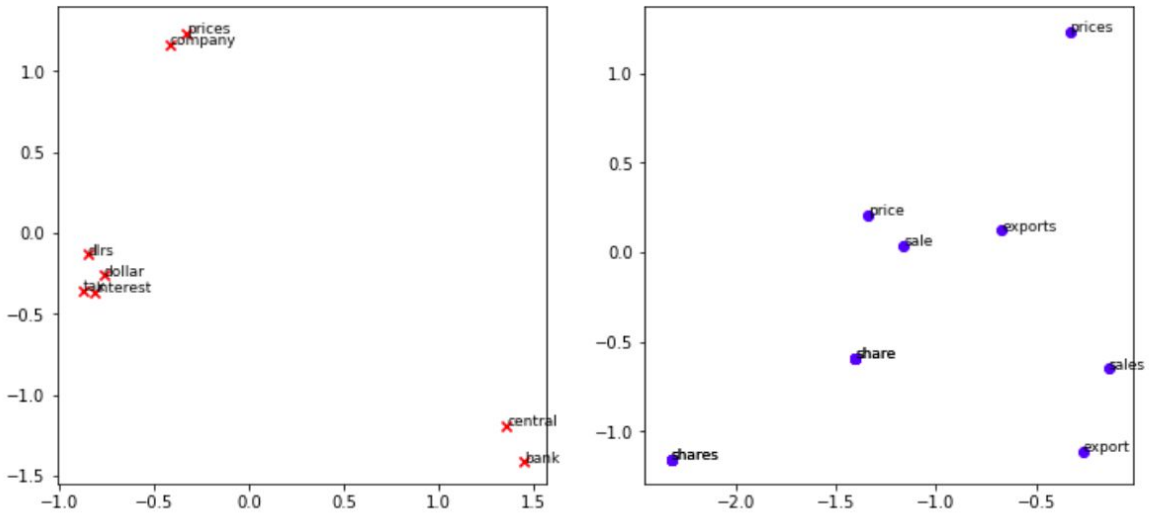
Word2Vec Assignment

2018H1120276P (Nimmi Ghetia)
2018H1120272P (Arunima Ghosh)

Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing where words or phrases from the vocabulary are mapped to vectors of real numbers. Determining suitable vector representations for words is a very useful feature extraction step to effectively represent a document in semantic space. It can help in solving problems related to information retrieval, document classification, question answering, named entity recognition, parsing, etc.
In this assignment, we have worked on developing the Word2Vec model and developed various syntactic and semantic relations

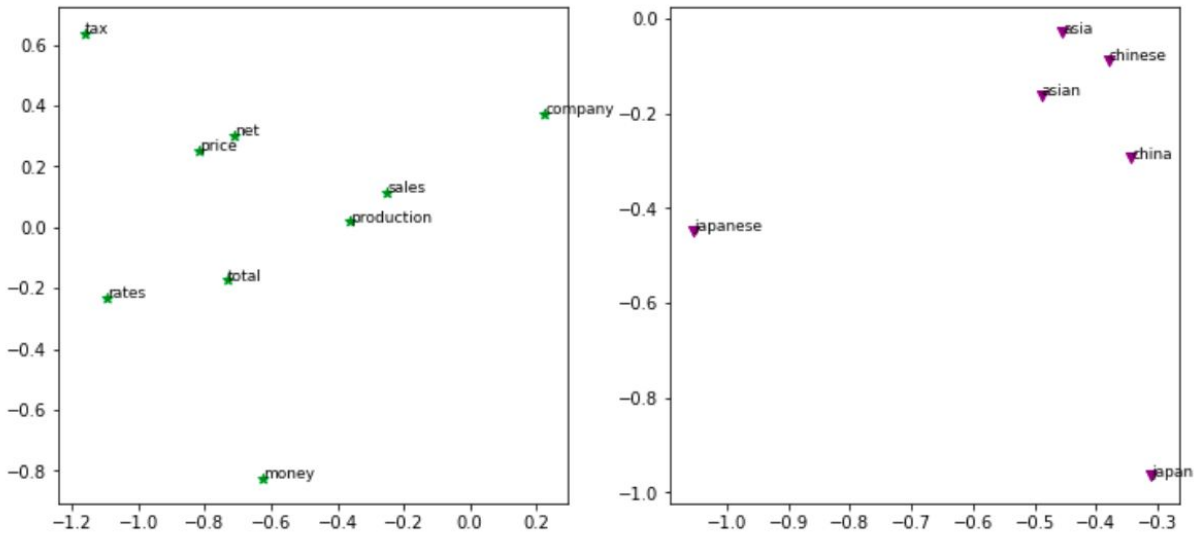
1. Selection: We have selected Reuters Corpus to work on. It is a multi-class, multi-label corpus. It consists of 90 classes, 7769 training documents, and 3019 testing documents.
2. Implementation:
    a. Data Preprocessing:
    In this step, we have converted the entire data in lower cases, removed all the stop words and new lists or words are generated.
    a. Co-occurrence:
    We have implemented a co-occurrence matrix on the processed data where the size of window 'k' is a user-defined variable. This will basically store the relationship between words. So this matrix basically stores the number of times a word and its related word has occurred in the corpus
    b. SVD:
    Next, we have implemented SVD which basically reduces dimensionality. Here dimensions' is a user-defined vector.
    c. Word2Vec representation:
    We implemented both the models of CBOW as well as the skip-gram.

3. Investigations:
    The results of the investigations are plotted with the help of SVD since the feature-vector dimensions were very large.
    a. Feature Vector Size = 30
    CBOW model



We can see that the relation between price-company and dollar-rupees- interest and central bank are clearly visible in the first figure. The similar similarity is found between price-prices, share-shares, sale-sales, export exprts

skip gram model:



Here we can see the relation between asia-asian and china-chinese in the second figure. In the first figure proper relationship was not established

b. Feature Vector Size = 150

CBOW model

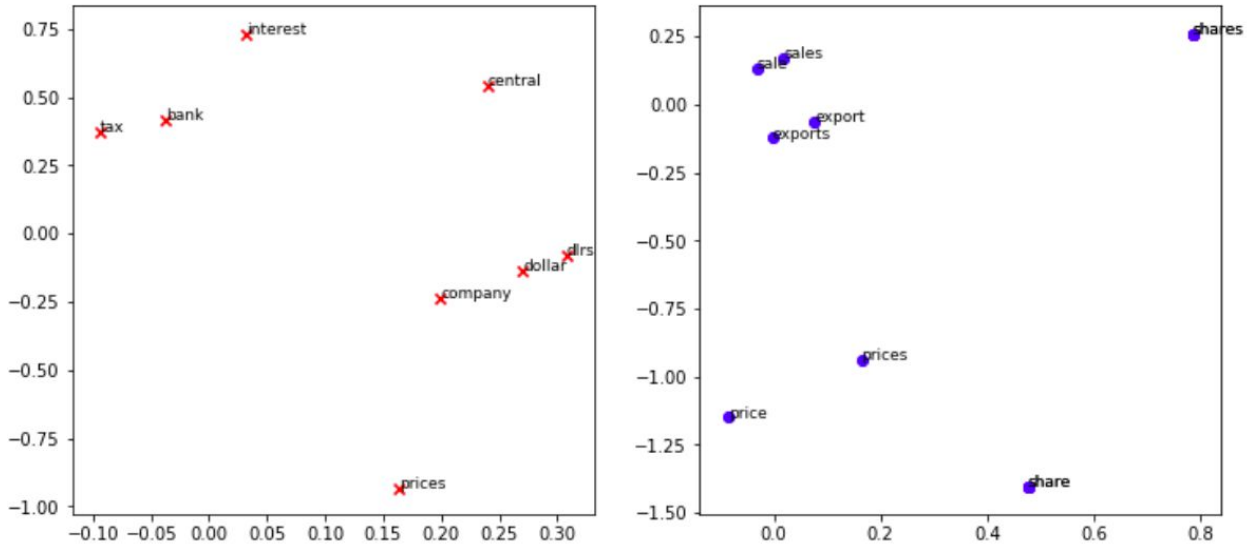


Figure 1 does not establish any proper relation. In figure 2 although some relationships like sale-sales and export-exports is visible the rest of the relations are distorted

Skip-Gram

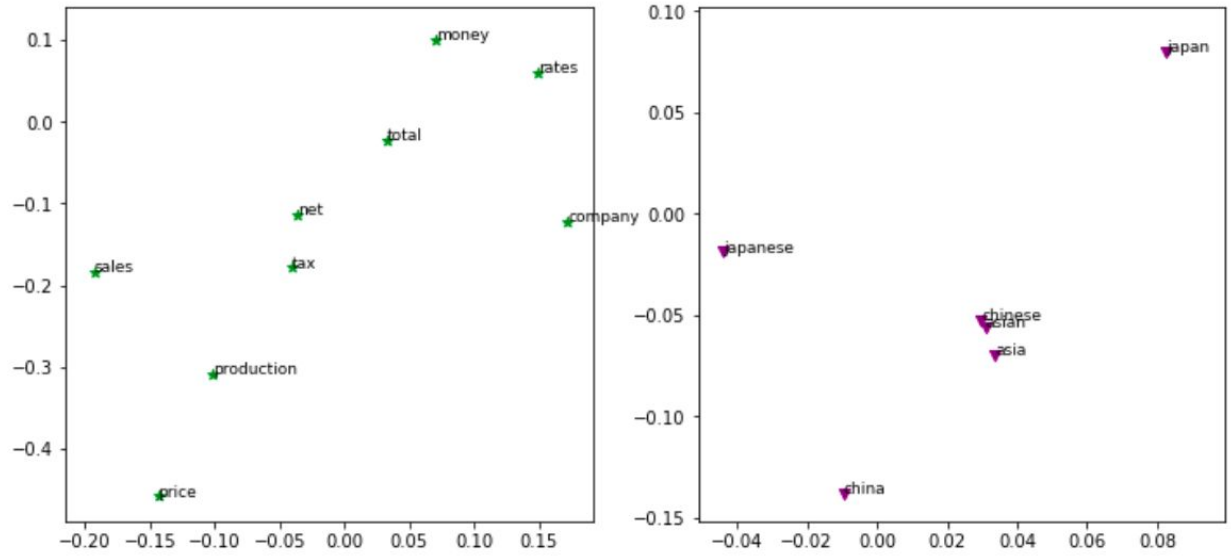


Figure 1 determine some relation like sales-production, money-company rest of the relations are distorted. Figure 2 all the relations are distorted.

Observation:

Similar experiment was performed for feature vector size = 30, 40, 50, 60, 90, 120, 150. The rest of the outputs are attached in the zip file. And it was observed that since in 150 feature-set the data is becoming sparse appropriate result is not obtained.

FindNearest()

-> takes three words and calculate the relation word1: word2:: word3: predword
Here is the example from Reuters dataset:

findNearest('gas','natural','oil')
---
burmah : 0.8098706940384643
crude : 0.7954616135301802
heating : 0.7663713232760455
columbia : 0.7659936486596814
palm : 0.7504905111056672

findNearest('oil','crude','gas')
---
natural : 0.8883573031151772
columbia : 0.8192052962280348
burmah : 0.7839666305918241
singapore : 0.7537032806143429
pte : 0.7421060100686429

It correctly predicts the word natural for relation oil: crude:: gas: natural
and almost 2nd correct in case of gas: natural:: oil: crude

MostSimilar()
-> takes a word and gives top k words with highest cosine similarity to the given word

mostSimilar('tax',5)
tax : 1.0000000000000002
gains : 0.8379168385115271
income : 0.8186382107373872
credit. : 0.7813233529641934
includes : 0.768354544936447

mostSimilar('bank',5)
bank : 1.0
banks : 0.754396326610673
bankers : 0.6651193532764178
federal : 0.5757686604560467
city : 0.5528639733398484

mostSimilar('oil',5)
oil : 0.9999999999999998
natural : 0.809120944572269
crude : 0.777939703235165
gas : 0.7655390381723018
palm : 0.7545248820516619

4. Innovation:
   On changing the value of alpha to 0.9 for reuters dataset we observe much better relation result for the relation

   findNearest('gas','natural','oil')

----	alpha = 0.75	alpha = 0.9
crude :	0.8382340594439821	0.7954616135301802
palm :	0.7607873125645798	0.7504905111056672
heating :	0.7607706907967298	
singapore :	0.7506262159686083	
refined :	0.7229045106788065	

   It also improves the accuracy of the prediction with better words as compared to natural , columbia , burmah , singapore , pte

   findNearest('oil','crude','gas')

```
----
natural  :  0.8704015458986947
exploration  :  0.7881949711979475
land  :  0.7611645552392589
petroleum  :  0.7583670079487373
singapore  :  0.7528134029867511
```

   Few architectural design change ideas are discussed for word2vec which can improve the accuracy of the word2vec model.

   a. Inserting an activation function at a hidden layer
   b. For out of vocabulary we can take prefix and postfix of the word for prediction in both models creating two extra hidden layers for capturing prefix word embedding and postfix word embeddings.