Assignment - 1

NLP CS60075
21CS10082
Yasaswani Rongali

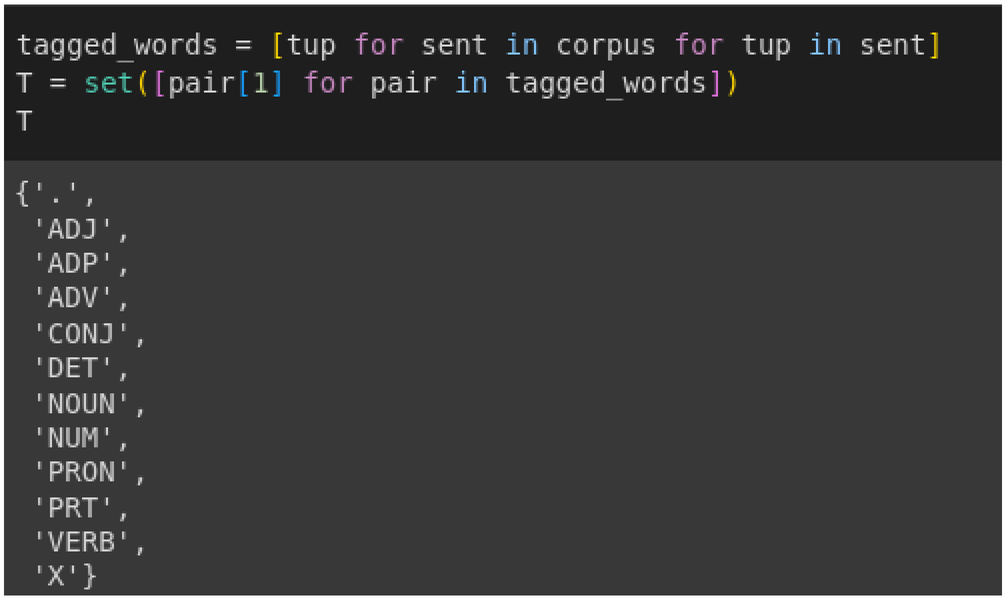
Enhancing Sentiment Analysis Using POS Tagging

REPORT

1) POS Tagger :

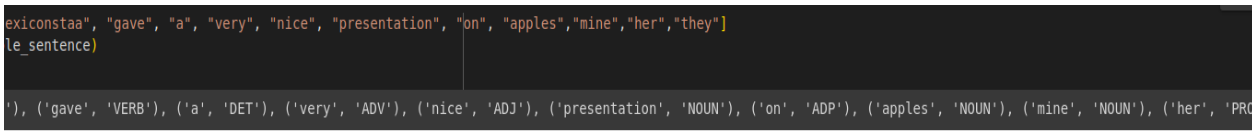
The algorithm identifies tags from the **universal tagset** (consists of 12 tags). For each word it computes max probability among the 12 possible probabilities and then the corresponding tag is assigned as the best tag for the word.

The set of tags



```
tagged_words = [tup for sent in corpus for tup in sent]
T = set([pair[1] for pair in tagged_words])
T
```

```
{'.',
 'ADJ',
 'ADP',
 'ADV',
 'CONJ',
 'DET',
 'NOUN',
 'NUM',
 'PRON',
 'PRT',
 'VERB',
 'X'}
```



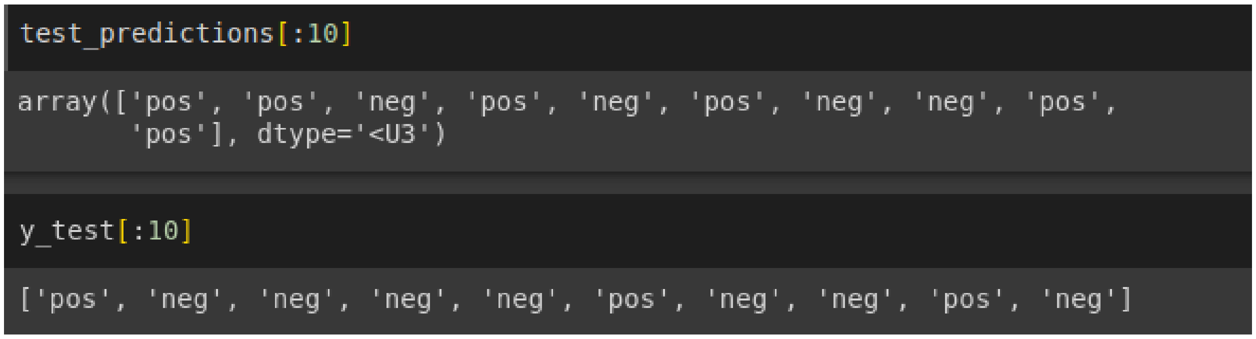
```
exiconstaa", "gave", "a", "very", "nice", "presentation", "on", "apples","mine","her","they"]
le_sentence)

'), ('gave', 'VERB'), ('a', 'DET'), ('very', 'ADV'), ('nice', 'ADJ'), ('presentation', 'NOUN'), ('on', 'ADP'), ('apples', 'NOUN'), ('mine', 'NOUN'), ('her', 'PRO
```

Tags for a sample sentence

2) <u>Vanilla Sentiment Analyzer:</u>

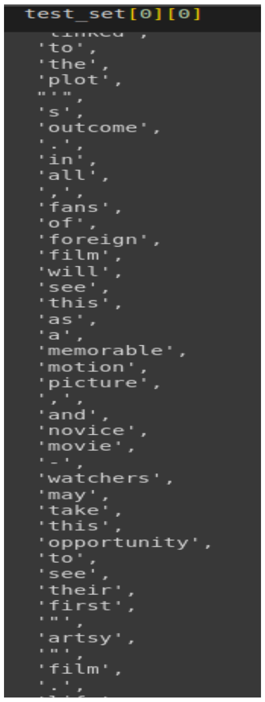
This is done by first creating word embeddings(converting words to vectors) using **Word2Vec** vectorizer. And then the **SVM Classifier** is trained on the training set(movie review corpus).

Actual output vs expected output for 1st 10 reviews of test data



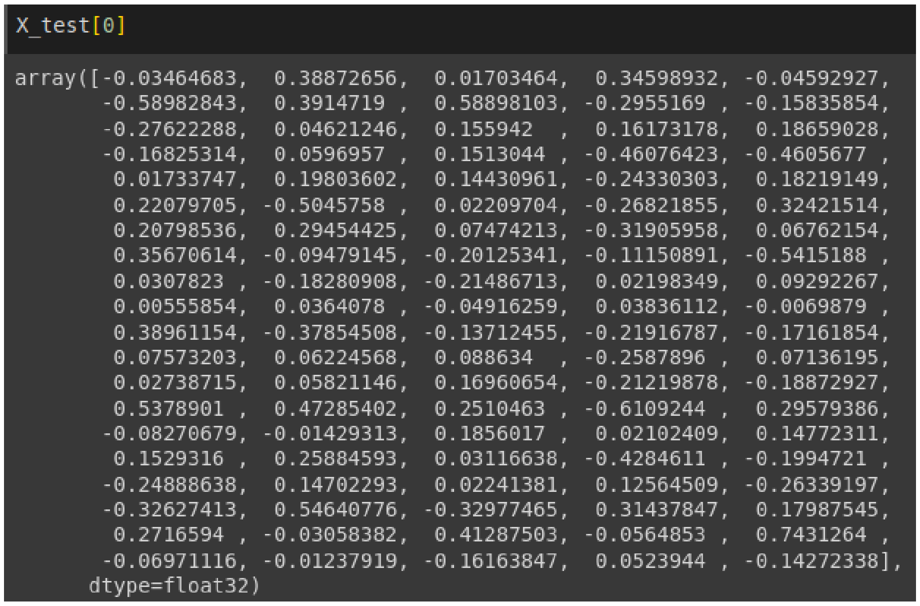
```
test_predictions[:10]

array(['pos', 'pos', 'neg', 'pos', 'neg', 'pos', 'neg', 'neg', 'pos',
       'pos'], dtype='<U3')

y_test[:10]

['pos', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'pos', 'neg']
```

Tokens of a review



```
test_set[0][0]
 'to',
 'the',
 'plot',
 "'",
 's',
 'outcome',
 '.',
 'in',
 'all',
 ',',
 'fans',
 'of',
 'foreign',
 'film',
 'will',
 'see',
 'this',
 'as',
 'a',
 'memorable',
 'motion',
 'picture',
 ',',
 'and',
 'novice',
 'movie',
 '-',
 'watchers',
 'may',
 'take',
 'this',
 'opportunity',
 'to',
 'see',
 'their',
 'first',
 '"',
 'artsy',
 '"',
 'film',
 '.',
```

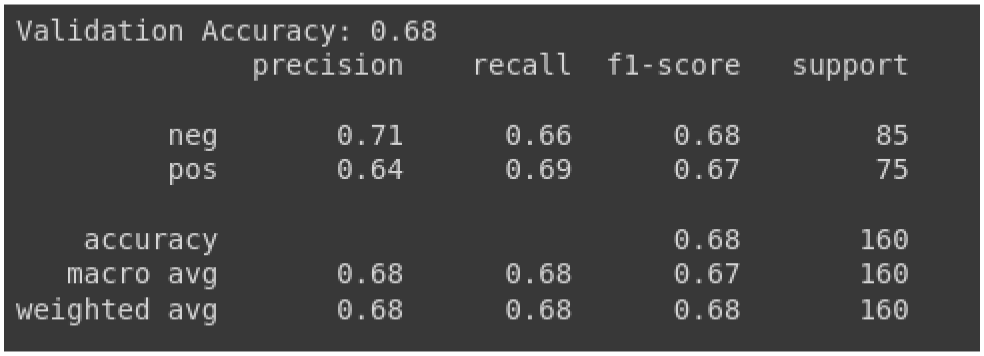
Embedding of the above review



```
X_test[0]
```

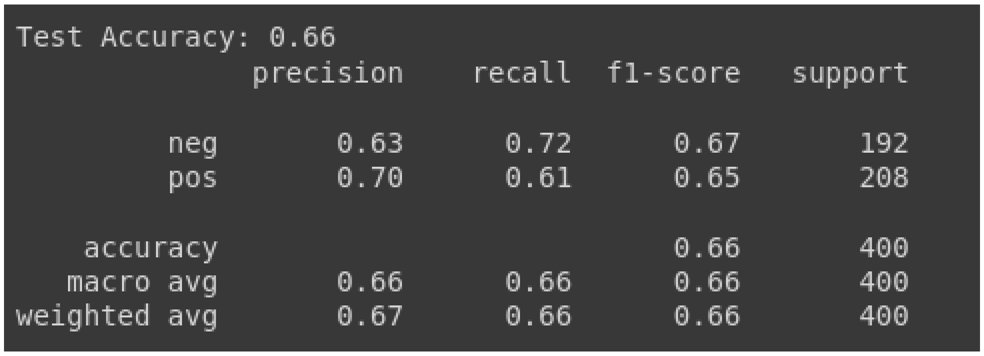
```
array([-0.03464683,  0.38872656,  0.01703464,  0.34598932, -0.04592927,
       -0.58982843,  0.3914719 ,  0.58898103, -0.2955169 , -0.15835854,
       -0.27622288,  0.04621246,  0.155942  ,  0.16173178,  0.18659028,
       -0.16825314,  0.0596957 ,  0.1513044 , -0.46076423, -0.4605677 ,
        0.01733747,  0.19803602,  0.14430961, -0.24330303,  0.18219149,
        0.22079705, -0.5045758 ,  0.02209704, -0.26821855,  0.32421514,
        0.20798536,  0.29454425,  0.07474213, -0.31905958,  0.06762154,
        0.35670614, -0.09479145, -0.20125341, -0.11150891, -0.5415188 ,
        0.0307823 , -0.18280908, -0.21486713,  0.02198349,  0.09292267,
        0.00555854,  0.0364078 , -0.04916259,  0.03836112, -0.0069879 ,
        0.38961154, -0.37854508, -0.13712455, -0.21916787, -0.17161854,
        0.07573203,  0.06224568,  0.088634  , -0.2587896 ,  0.07136195,
        0.02738715,  0.05821146,  0.16960654, -0.21219878, -0.18872927,
        0.5378901 ,  0.47285402,  0.2510463 , -0.6109244 ,  0.29579386,
       -0.08270679, -0.01429313,  0.1856017 ,  0.02102409,  0.14772311,
        0.1529316 ,  0.25884593,  0.03116638, -0.4284611 , -0.1994721 ,
       -0.24888638,  0.14702293,  0.02241381,  0.12564509, -0.26339197,
       -0.32627413,  0.54640776, -0.32977465,  0.31437847,  0.17987545,
        0.2716594 , -0.03058382,  0.41287503, -0.0564853 ,  0.7431264 ,
       -0.06971116, -0.01237919, -0.16163847,  0.0523944 , -0.14272338],
      dtype=float32)
```

Classification report for validation set



```
Validation Accuracy: 0.68
              precision    recall  f1-score   support

         neg       0.71      0.66      0.68        85
         pos       0.64      0.69      0.67        75

    accuracy                           0.68       160
   macro avg       0.68      0.68      0.67       160
weighted avg       0.68      0.68      0.68       160
```

Classification report for test set



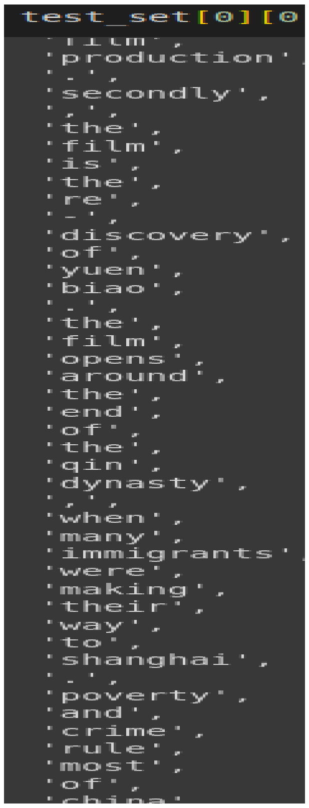
```
Test Accuracy: 0.66
              precision    recall  f1-score   support

         neg       0.63      0.72      0.67       192
         pos       0.70      0.61      0.65       208

    accuracy                           0.66       400
   macro avg       0.66      0.66      0.66       400
weighted avg       0.67      0.66      0.66       400
```

3) Enhanced sentiment analysis:

Using POS tagger gives us an extra edge, because **adjectives** and **adverbs** are the words that actually contribute to the sentiment of a review or a paragraph of text. So, instead of creating word embeddings of all words, if we created embeddings of just adverbs and adjectives and trained a classifier on this dataset, we would get better results of sentiment analysis.

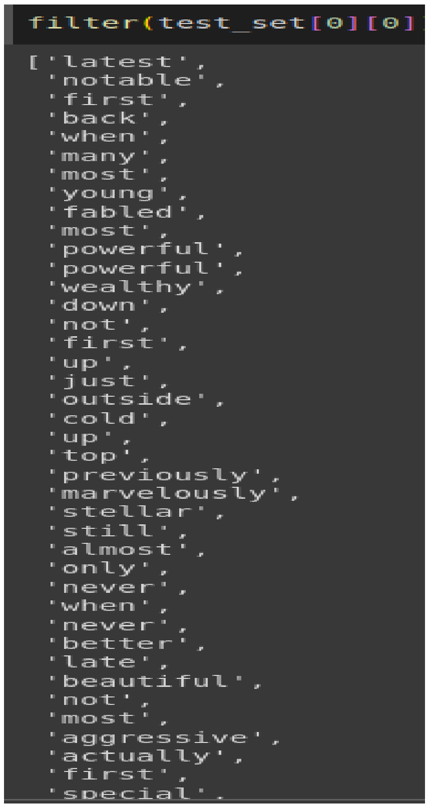
So first, the viterbi algorithm is applied on the document whose embeddings are to be generated. We only retain the words which have **"ADV"(Adverb)** and **"ADJ"(Adjectives)** as their tags. Now this filtered document is used for training the SVM Classifier.

Tokens of test_set[0][0]



```
test_set[0][0
  'film',
  'production',
  '.',
  'secondly',
  ',',
  'the',
  'film',
  'is',
  'the',
  're',
  '-',
  'discovery',
  'of',
  'yuen',
  'biao',
  '.',
  'the',
  'film',
  'opens',
  'around',
  'the',
  'end',
  'of',
  'the',
  'qin',
  'dynasty',
  ',',
  'when',
  'many',
  'immigrants',
  'were',
  'making',
  'their',
  'way',
  'to',
  'shanghai',
  '.',
  'poverty',
  'and',
  'crime',
  'rule',
  'most',
  'of',
  'china',
```

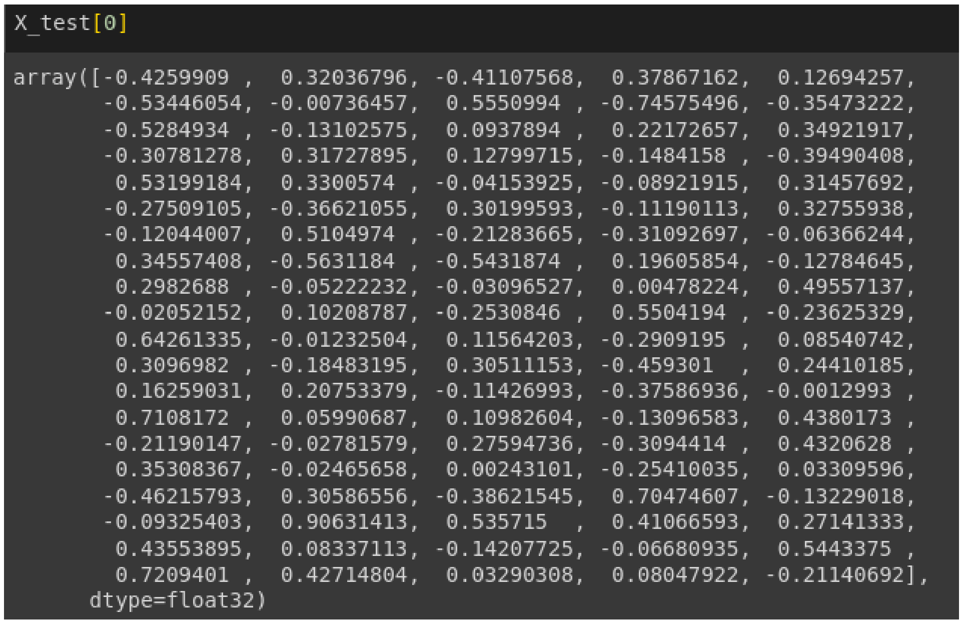
*Filtered tokens of test_set[0][0] containing only **adverbs** and **adjectives***



```
filter(test_set[0][0]
```

```
['latest',
 'notable',
 'first',
 'back',
 'when',
 'many',
 'most',
 'young',
 'fabled',
 'most',
 'powerful',
 'powerful',
 'wealthy',
 'down',
 'not',
 'first',
 'up',
 'just',
 'outside',
 'cold',
 'up',
 'top',
 'previously',
 'marvelously',
 'stellar',
 'still',
 'almost',
 'only',
 'never',
 'when',
 'never',
 'better',
 'late',
 'beautiful',
 'not',
 'most',
 'aggressive',
 'actually',
 'first',
 'special',
```

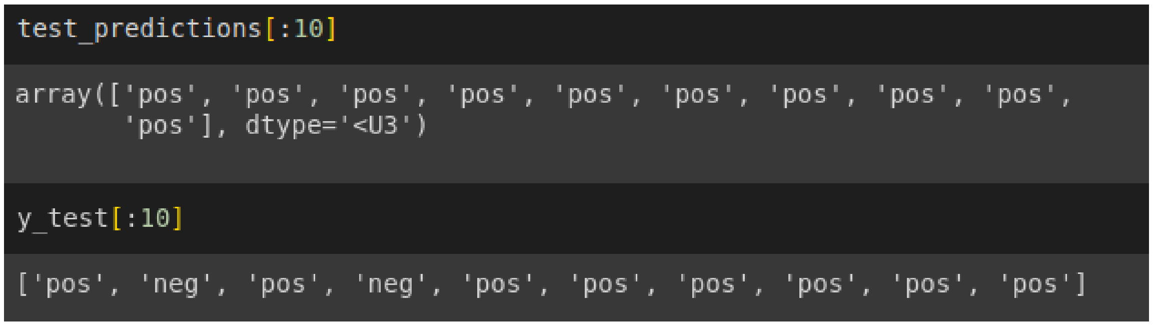
Embeddings of the above words



```
X_test[0]
```

```
array([-0.4259909 ,  0.32036796, -0.41107568,  0.37867162,  0.12694257,
       -0.53446054, -0.00736457,  0.5550994 , -0.74575496, -0.35473222,
       -0.5284934 , -0.13102575,  0.0937894 ,  0.22172657,  0.34921917,
       -0.30781278,  0.31727895,  0.12799715, -0.1484158 , -0.39490408,
        0.53199184,  0.3300574 , -0.04153925, -0.08921915,  0.31457692,
       -0.27509105, -0.36621055,  0.30199593, -0.11190113,  0.32755938,
       -0.12044007,  0.5104974 , -0.21283665, -0.31092697, -0.06366244,
        0.34557408, -0.5631184 , -0.5431874 ,  0.19605854, -0.12784645,
        0.2982688 , -0.05222232, -0.03096527,  0.00478224,  0.49557137,
       -0.02052152,  0.10208787, -0.2530846 ,  0.5504194 , -0.23625329,
        0.64261335, -0.01232504,  0.11564203, -0.2909195 ,  0.08540742,
        0.3096982 , -0.18483195,  0.30511153, -0.459301  ,  0.24410185,
        0.16259031,  0.20753379, -0.11426993, -0.37586936, -0.0012993 ,
        0.7108172 ,  0.05990687,  0.10982604, -0.13096583,  0.4380173 ,
       -0.21190147, -0.02781579,  0.27594736, -0.3094414 ,  0.4320628 ,
        0.35308367, -0.02465658,  0.00243101, -0.25410035,  0.03309596,
       -0.46215793,  0.30586556, -0.38621545,  0.70474607, -0.13229018,
       -0.09325403,  0.90631413,  0.535715  ,  0.41066593,  0.27141333,
        0.43553895,  0.08337113, -0.14207725, -0.06680935,  0.5443375 ,
        0.7209401 ,  0.42714804,  0.03290308,  0.08047922, -0.21140692],
      dtype=float32)
```

Expected output vs actual output for the first 10 reviews of test set



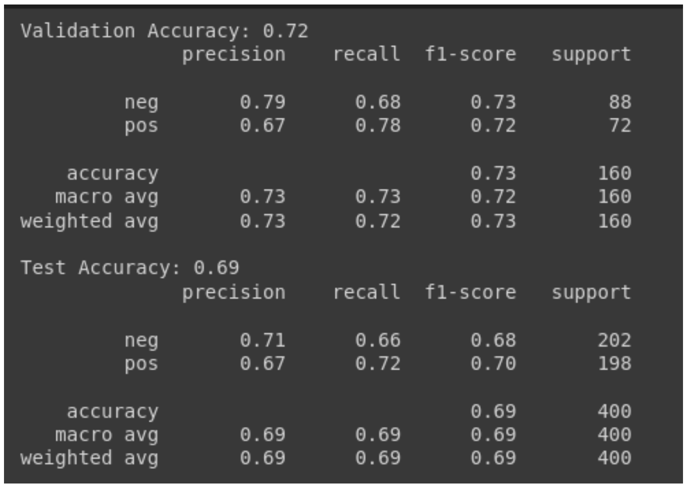
```
test_predictions[:10]
```

```
array(['pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos',
       'pos'], dtype='<U3')
```

```
y_test[:10]
```

```
['pos', 'neg', 'pos', 'neg', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos']
```

Classification reports for validation and test set



```
Validation Accuracy: 0.72
              precision    recall  f1-score   support

         neg       0.79      0.68      0.73        88
         pos       0.67      0.78      0.72        72

    accuracy                           0.73       160
   macro avg       0.73      0.73      0.72       160
weighted avg       0.73      0.72      0.73       160

Test Accuracy: 0.69
              precision    recall  f1-score   support

         neg       0.71      0.66      0.68       202
         pos       0.67      0.72      0.70       198

    accuracy                           0.69       400
   macro avg       0.69      0.69      0.69       400
weighted avg       0.69      0.69      0.69       400
```

It can be seen from the accuracy scores of test datasets in both the methods, that the enhanced sentiment analyzer gives better results than the baseline method.

-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-