anger\_dict = ['acrimony', 'animosity', 'annoyance', 'antagonism', 'displeasure', 'enmity', 'exasperation', 'fury',  
 'hatred', 'impatience', 'indignation', 'ire', 'irritation', 'outrage', 'passion', 'rage', 'resentment',  
 'temper', 'violence', 'chagrin', 'choler', 'conniption', 'dander', 'disapprobation', 'distemper', 'gall',  
 'huff', 'infuriation', 'irascibility', 'irritability', 'miff', 'peevishness', 'petulance', 'pique',  
 'rankling', 'soreness', 'stew', 'storm', 'tantrum', 'tiff', 'umbrage', 'vexation', 'blow up', 'cat fit',  
 'hissy', 'ill humor', 'ill temper', 'mad', 'slow burn'  
 ]  
  
sad\_dict = ['unhappy', 'sorrowful', 'dejected', 'depressed', 'downcast', 'miserable', 'down', 'despondent',  
 'despairing', 'disconsolate', 'desolate', 'wretched', 'glum', 'gloomy', 'doleful', 'dismal', 'melancholy',  
 'mournful', 'woebegone', 'forlorn', 'crestfallen', 'heartbroken', 'inconsolable',  
 'informalblue', 'down in the mouth', 'down in the dumps']  
  
fear\_dict = ['terror', 'fright', 'fearfulness', 'horror', 'alarm', 'panic', 'agitation', 'trepidation', 'dread',  
 'consternation',  
 'dismay', 'distress', 'anxiety', 'worry', 'angst', 'unease', 'uneasiness', 'apprehension',  
 'apprehensiveness', 'nervousness',  
 'nerves', 'perturbation', 'foreboding', 'informalthe creeps', 'the shivers', 'the willies',  
 'the heebie-jeebies', 'jitteriness',  
 'twitchiness', 'phobia', 'aversion', 'antipathy', 'dread', 'bugbear', 'nightmare', 'horror',  
 'terror', 'anxiety', 'neurosis']  
  
  
happy\_dict = ['cheerful', 'cheery', 'merry', 'joyful', 'jovial', 'jolly', 'jocular', 'gleeful', 'carefree', 'delight',  
 'smile',  
 'grin',  
 'in good spirits', 'in a good mood', 'lighthearted', 'pleased', 'contented', 'satisfied', 'gratified',  
 'buoyant',  
 'radiant', 'sunny',  
 'blithe', 'joyous', 'thrilled', 'elated', 'exhilarated', 'ecstatic', 'blissful', 'euphoric', 'overjoyed',  
 'exultant', 'rapturous',  
 'in seventh heaven', 'on cloud nine', 'walking on air', 'jubilant', 'over the moon',  
 'on top of the world', 'tickled pink',  
 'on a high', 'bland', 'blissful', 'calm', 'capricious', 'cheerful', 'confident', 'content', 'convinced',  
 'dazed', 'delighted',  
 'delightful', 'ecstatic', 'elated', 'enchanted', 'epicurean', 'excessive', 'fain', 'fanciful',  
 'formidable', 'funny', 'glad',  
 'glorious', 'gratified', 'hilarious', 'hopeful', 'humorous', 'joyful', 'jubilant', 'overwhelmed',  
 'sanguine', 'sensuous', 'solemn',  
 'splendid', 'sprightly', 'spruce', 'sybaritic', 'thrilled', 'voluptuous', 'wry', 'zestful', 'impish',  
 'playful', 'prankish', 'roguish',  
 'whimsical', 'fastidious' 'heedful']

Lexical:

Example:

Tweet:

lol, will not check twitter till the preview and fancam died down on TL. i'm in withdrawal mode. \U0001f602\U0001f602\U0001f602

feature vector:

['laughing', 'out', 'loud', 'check', 'twitter', 'till', 'preview', 'fancam', 'died', 'tl', 'withdrawal', 'mode', 'u001f602', 'u001f602', 'u001f602']

Tweet:

I FIND IT FUNNY that i met alot of bts stans who are tokyo ghoul fans on twitter www

Feature Vector:

['find', 'funny', 'met', 'alot', 'bts', 'stans', 'tokyo', 'ghoul', 'fans', 'twitter', 'ww']

Tweets:

RT @CathyebCathye: As a @FoxNews watcher I am glad to see Obama crumble- Smooth talker has ruined America as we used to know it.\u2026

Feature Vector:

['rt', 'cathyebcathye', 'foxnews', 'watcher', 'glad', 'see', 'obama', 'crumble', 'smooth', 'talker', 'ruined', 'america', 'used', 'know', 'u2026']

Tweets:

@TheRedRag Daily Wail did the digging and found out Nutall is a bullshit merchant, par for the course with Ukip... <https://t.co/8r2p5dHO6P>

Feature Vector:

['theredrag', 'daily', 'wail', 'digging', 'found', 'nutall', 'bullshit', 'merchant', 'par', 'course', 'ukip']

regex\_str = [  
 emoticons\_str,  
 r'<[^>]+>', # HTML tags  
 r'(?:@[\w\_]+)', # @-mentions  
 r"(?:\#+[\w\_]+[\w\'\_\-]\*[\w\_]+)", # hash-tags  
 r'http[s]?://(?:[a-z]|[0-9]|[$-\_@.&amp;+]|[!\*\(\),]|(?:%[0-9a-f][0-9a-f]))+', # URLs  
  
 r'(?:(?:\d+,?)+(?:\.?\d+)?)', # numbers  
 r"(?:[a-z][a-z'\-\_]+[a-z])", # words with - and '  
 r'(?:[\w\_]+)', # other words  
 r'(?:\S)' # anything else  
]

Applied Syntactic Improvement:

We have applied POS tagging and Chunking feature using NLTK /regular expression.

First we have applied POS tagging using NLTK library and then we have added words to our feature vector which are tagged with Noun, Verb, Adjective and Adverb. We have also added count of Noun, Verb, Adjective and Adverb in our feature vector which we are using during sentiment.

For Example:

Tweet:

RT @itslifethought: Whatever is bringing you down, let it GO! You don't need that negativity in your life. Keep calm and be positive. Good\u2026

POS tagging:

[('rt', 'NN'), ('itslifethought', 'NN'), ('whatever', 'WDT'), ('bringing', 'VBG'), ('let', 'NN'), ('go', 'VB'), ('need', 'VB'), ('negativity', 'JJ'), ('life', 'NN'), ('keep', 'VB'), ('calm', 'JJ'), ('positive', 'JJ'), ('good', 'JJ'), ('u2026', 'NN')]

Tweet:

RT @Tyche\_Boys: Dont u dare drag exo down w/o knowing actual facts.And u have no right to say exos floppin bc ur group debuted only a year\u2026

POS tagging:

[('rt', 'NN'), ('tyche', 'NN'), ('boys', 'VBZ'), ('dont', 'JJ'), ('u', 'JJ'), ('dare', 'NN'), ('drag', 'NN'), ('exo', 'NN'), ('w', 'NN'), ('knowing', 'VBG'), ('actual', 'JJ'), ('facts', 'NNS'), ('u', 'JJ'), ('right', 'JJ'), ('say', 'VBP'), ('exos', 'JJ'), ('floppin', 'NN'), ('bc', 'NN'), ('ur', 'JJ'), ('group', 'NN'), ('debuted', 'VBD'), ('year', 'NN'), ('u2026', 'NN')]

Tweet:

I want a dude that still kiss me when he mad

POS tagging:

[('want', 'JJ'), ('dude', 'NN'), ('still', 'RB'), ('kiss', 'VBZ'), ('mad', 'NN')]

Second syntactic Feature:

Applied Chunking using NLTK library we used regular expression to chunk the required tagged words:

Below regular expression being used for chunking:

grammar = """ NP:  
 {<NNS><VBP>} # combine nns and vbp  
 {<V.\*> <TO> <V.\*>} #combine V to V.\*  
 {<N.\*>(4,)} # combine nouns  
 VB: {<VBD>\*<JJ>\*<NNS>} #combine verb adverb and noun   
 """

Example:

**Tweet:**

RT @jacksonwang\_je: @kkangmiina hello there! Welcome to FnP do enjoy your stay here! GOT7's Jackson here, pleased to meet you &amp; happy b\u2026

**After Chunking:**

**(S**

**rt/NN**

**jacksonwang/NN**

**je/NN**

**kkangmiina/NNP**

**hello/VBZ**

**welcome/JJ**

**fnp/NN**

**enjoy/NN**

**stay/VBP**

**got7/JJ**

**jackson/NN**

**pleased/VBD**

**meet/NN**

**amp/NN**

**happy/JJ**

**b/NN**

**u2026/NN)**

**Tweet:**

RT @beckyxcruz: I'm so mad and irritated fml I need to smoke

**After Chunking:**

**(S rt/NN beckyxcruz/NN mad/NN irritated/VBD fml/JJ need/NN smoke/NN)**

Tweet:

I liked a @YouTube video https://t.co/K6Unn4VwlZ Jess Glynne - Love Come Down - Jools' Annual Hootenanny - BBC Two

**After Chunking:**

**(S**

**liked/VBN**

**youtube/JJ**

**video/NN**

**jess/NN**

**glynne/NN**

**love/VBP**

**come/NN**

**(VB jools/NNS)**

**annual/JJ**

**hootenanny/JJ**

**bbc/NN**

**two/CD)**

**Tweet:**

#NEW #99c "Hilarious and perfectly 'played' sports romance" Destiny on Ice by @AuthorSRGrey https://t.co/TPOFDChruH <https://t.co/k9OsVnJMFT>

**After Chunking:**

(S

new/JJ

hilarious/JJ

perfectly/RB

played/VBN

(VB sports/NNS)

romance/NN

destiny/JJ

ice/NN

authorsrgrey/NN)

On top of both this method we applied semantic feature and got different accuracy.

**Our analysis after various experiments:**

In case of chunking as we are considering only words which matches with our grammar so sometimes we are removing some important noun and adverbs. Hence we are getting less accuracy in case of chunking compared with POS tagging.

Experiments Result:

Accuracy after applying chunking:

positive accuracy

85.0

negative accuracy

15.0

Accuracy after applying POS tagging:

positive accuracy

95.0

negative accuracy

5.0

**Semantic Features:**

We have applied Lesk algorithm and hypernym feature using NLTK tool.

**Lesk Algorithm:**

Once we get list result after pos-tagging/chunking we have applied Lesk algorithm to below conditions:

**if** words[1] == 'NN' **or** 'NNS' **or** 'NNP':

**elif** words[1] == 'VB' **or** 'VBD':

**elif** words[1] == 'JJ':

**elif** words[1] == 'RB':

**Examples:**

['questions']

**After Lesk algorithm:**

['interrogate']

**2nd Semantic Feature:**

**Hypernym: Used NLTK library**

On top of Lesk algorithm and syntactic feature applied Hypernym to all the word token which we get. Also I have applied hypernym to new word if we get after Lesk algorithm(example above case) . We are adding only unique words in hypernym result.

Example:

Below feature vector we will get for example tweets:

If we use POS-tagging and after applying sematic feature:

**Tweet:**

**@JaclynHStrauss Is it me or do all Assad shills in the west have the ugliness of their hearts written all over thei\u2026** [**https://t.co/4UdwQPKYhd**](https://t.co/4UdwQPKYhd)

**Feature Vector build after applying all the features:**

{'mode': 1, 'deed': 1, 'telephone': 1, 'assistant': 1, 'write': 1, 'written': 1, 'acquiring': 1, 'getting': 1, 'noesis': 1, 'find': 1, 'telephone\_call': 1, 'want': 1, 'vow': 1, 'manner': 1, 'psychological\_feature': 1, 'style': 1, 'game': 1, 'act': 1, 'terror': 1, 'property': 1, 'nowadays': 1, 'regain': 1, 'create\_verbally': 1, 'aid': 1, 'phone\_call': 1, 'representation': 1, 'verb': 0, 'news': 1, 'appropriation': 1, 'conversation': 1, 'compose': 1, 'physical\_object': 1, 'jaclynhstrauss': 1, 'voice\_communication': 1, 'attention': 1, 'abstraction': 1, 'someone': 1, 'satisfy': 1, 'telephony': 1, 'oral\_communication': 1, 'photograph': 1, 'artifact': 1, 'fact': 1, 'textile': 1, 'supporter': 1, 'instrumentation': 1, 'person': 1, 'decoy': 1, 'affright': 1, 'group\_action': 1, 'entity': 1, 'help': 1, 'causal\_agency': 1, 'invariably': 1, 'emotion': 1, 'photo': 1, 'skilled\_workman': 1, 'commitment': 1, 'fearfulness': 1, 'indite': 1, 'appearance': 1, 'direction': 1, 'accomplice': 1, 'visual\_aspect': 1, 'talking': 1, 'telecom': 1, 'confederate': 1, 'panic': 1, 'relation': 1, 'activity': 1, 'fright': 1, 'medium': 1, 'cognition': 1, 'cloth': 1, 'cards': 1, 'abstract\_entity': 1, 'mortal': 1, 'thei': 1, 'detail': 1, 'recapture': 1, 'human\_action': 1, 'fabric': 1, 'intelligence': 1, 'object': 1, 'gratify': 1, 'wish': 1, 'please': 1, 'location': 1, 'creation': 1, 'state': 1, 'subject\_matter': 1, 'knowledge': 1, 'u2026': 1, 'today': 1, 'instrumentality': 1, 'auditory\_communication': 1, 'huffy': 1, 'speech': 1, 'pen': 1, 'present': 1, 'West': 1, 'individual': 1, 'whist': 1, 'hearts': 1, 'long\_whist': 1, 'worker': 1, 'causal\_agent': 1, 'cause': 1, 'retrieve': 1, 'message': 1, 'recover': 1, 'item': 1, 'group': 1, 'adjective': 0, 'pic': 1, 'ugliness': 1, 'post': 1, 'picture': 1, 'artefact': 1, 'substance': 1, 'functionary': 1, 'speech\_communication': 1, 'management': 1, 'telecommunication': 1, 'respect': 1, 'short\_whist': 1, 'acquire': 1, 'soul': 1, 'west': 1, 'place': 1, 'regard': 1, 'adverb': 0, 'steerer': 1, 'get': 1, 'talk': 1, 'helper': 1, 'social\_relation': 1, 'grouping': 1, 'work': 1, 'assad': 1, 'time': 1, 'way': 1, 'delight': 1, 'noun': 9, 'spoken\_communication': 1, 'seizure': 1, 'communication': 1, 'shills': 1, 'dedication': 1, 'region': 1, 'social\_control': 1, 'constantly': 1, 'official': 1, 'material': 1, 'promise': 1, 'pledge': 1, 'tidings': 1, 'care': 1, 'human\_activity': 1, 'whole': 1, 'feeling': 1, 'update': 1, 'information': 1, 'language': 1, 'Occident': 1, 'exposure': 1, 'content': 1, 'desire': 1, 'call': 1, 'attribute': 1, 'unit': 1, 'feisty': 1, 'politics': 1, 'political\_relation': 1, 'card\_game': 1, 'station': 1, 'shill': 1, 'word': 1, 'event': 1, 'quality': 1, 'skilled\_worker': 1, 'position': 1, 'fear': 1, 'pique': 1, 'tending': 1, 'Black\_Maria': 1, 'touchy': 1, 'trained\_worker': 1, 'wishing': 1, 'spoken\_language': 1, 'fashion': 1, 'info': 1, 'make': 1, 'thin-skinned': 1, 'somebody': 1, 'always': 1, 'physical\_entity': 1, 'assurance': 1, 'point': 1, 'create': 1}

if we use chunking and after applying sematic feature:

**Tweet:**

RT @hiphop\_babylion: im so happy to see him in a good state. he looks very much fine to me and im beyond satisfied to see him casually w\u2026

**Feature Vector:**

{'aspect': 1, 'facial\_expression': 1, 'face': 1, 'abstract\_entity': 1, 'visage': 1, 'noun': 0, 'entity': 1, 'cause': 1, 'adjective': 0, 'vital\_principle': 1, 'looks': 1, 'attribute': 1, 'look': 1, 'appearance': 1, 'expression': 1, 'abstraction': 1, 'life\_principle': 1, 'visual\_aspect': 1, 'countenance': 1, 'verb': 1, 'quality': 1, 'adverb': 0, 'causal\_agency': 1, 'spirit': 1, 'physical\_entity': 1, 'causal\_agent': 1}

**Tweet:**

Put down the Bible and go take a science Class. Seriously your ignorance is at an astounding level and all you sound like uneducated fool.

**Feature Vector:**

{'period\_of\_time': 1, 'amount': 1, 'sound': 1, 'noun': 0, 'period': 1, 'tour': 1, 'adjective': 0, 'proceeds': 1, 'issue': 1, 'measure': 1, 'hours': 1, 'relation': 1, 'takings': 1, 'payoff': 1, 'verb': 2, 'gain': 1, 'time\_period': 1, 'fundamental\_quantity': 1, 'attribute': 1, 'turn': 1, 'quantity': 1, 'abstract\_entity': 1, 'duty\_period': 1, 'take': 1, 'entity': 1, 'shift': 1, 'work\_shift': 1, 'yield': 1, 'assets': 1, 'go': 1, 'sum\_of\_money': 1, 'abstraction': 1, 'spell': 1, 'possession': 1, 'fundamental\_measure': 1, 'return': 1, 'work\_time': 1, 'financial\_gain': 1, 'adverb': 0, 'sound\_property': 1, 'property': 1, 'sum': 1, 'income': 1, 'amount\_of\_money': 1}

**Improved System (How to predict sentiment):**

We have used bag of words strategy to predict sentiment of tweet:

We have used Natural language concept while creating Bag of words such as synonyms, hypernyms, tokenization, POS-tagging(In case when we are getting data from Training data)

How we created Bag of words for Anger, Sad, Fear, Happy:

**First step:**

In this step we get the synonyms using NLTK Library for all words anger, sad, fear, happy.

Below are the words in each dictionary it will be use in BaseLine system. As words are less so we did not get good accuracy in our baseline system major factor apart from not applying NLP features.

anger\_dict = ['acrimony', 'animosity', 'annoyance', 'antagonism', 'displeasure', 'enmity', 'exasperation', 'fury',  
 'hatred', 'impatience', 'indignation', 'ire', 'irritation', 'outrage', 'passion', 'rage', 'resentment',  
 'temper', 'violence', 'chagrin', 'choler', 'conniption', 'dander', 'disapprobation', 'distemper', 'gall',  
 'huff', 'infuriation', 'irascibility', 'irritability', 'miff', 'peevishness', 'petulance', 'pique',  
 'rankling', 'soreness', 'stew', 'storm', 'tantrum', 'tiff', 'umbrage', 'vexation', 'blow up', 'cat fit',  
 'hissy', 'ill humor', 'ill temper', 'mad', 'slow burn'  
 ]  
  
sad\_dict = ['unhappy', 'sorrowful', 'dejected', 'depressed', 'downcast', 'miserable', 'down', 'despondent',  
 'despairing', 'disconsolate', 'desolate', 'wretched', 'glum', 'gloomy', 'doleful', 'dismal', 'melancholy',  
 'mournful', 'woebegone', 'forlorn', 'crestfallen', 'heartbroken', 'inconsolable',  
 'informalblue', 'down in the mouth', 'down in the dumps']  
  
fear\_dict = ['terror', 'fright', 'fearfulness', 'horror', 'alarm', 'panic', 'agitation', 'trepidation', 'dread',  
 'consternation',  
 'dismay', 'distress', 'anxiety', 'worry', 'angst', 'unease', 'uneasiness', 'apprehension',  
 'apprehensiveness', 'nervousness',  
 'nerves', 'perturbation', 'foreboding', 'informalthe creeps', 'the shivers', 'the willies',  
 'the heebie-jeebies', 'jitteriness',  
 'twitchiness', 'phobia', 'aversion', 'antipathy', 'dread', 'bugbear', 'nightmare', 'horror',  
 'terror', 'anxiety', 'neurosis']  
  
  
happy\_dict = ['cheerful', 'cheery', 'merry', 'joyful', 'jovial', 'jolly', 'jocular', 'gleeful', 'carefree', 'delight',  
 'smile',  
 'grin',  
 'in good spirits', 'in a good mood', 'lighthearted', 'pleased', 'contented', 'satisfied', 'gratified',  
 'buoyant',  
 'radiant', 'sunny',  
 'blithe', 'joyous', 'thrilled', 'elated', 'exhilarated', 'ecstatic', 'blissful', 'euphoric', 'overjoyed',  
 'exultant', 'rapturous',  
 'in seventh heaven', 'on cloud nine', 'walking on air', 'jubilant', 'over the moon',  
 'on top of the world', 'tickled pink',  
 'on a high', 'bland', 'blissful', 'calm', 'capricious', 'cheerful', 'confident', 'content', 'convinced',  
 'dazed', 'delighted',  
 'delightful', 'ecstatic', 'elated', 'enchanted', 'epicurean', 'excessive', 'fain', 'fanciful',  
 'formidable', 'funny', 'glad',  
 'glorious', 'gratified', 'hilarious', 'hopeful', 'humorous', 'joyful', 'jubilant', 'overwhelmed',  
 'sanguine', 'sensuous', 'solemn',  
 'splendid', 'sprightly', 'spruce', 'sybaritic', 'thrilled', 'voluptuous', 'wry', 'zestful', 'impish',  
 'playful', 'prankish', 'roguish',  
 'whimsical', 'fastidious' 'heedful']

**Second Step:**

We have applied hypernym feature using NLTK library for each words, and after applying these feature I have applied set function to get the unique words in the bag-of-words. In all the dictionary.

**Final Step:**

We use training data and applied all the same lexical feature which we are applying for tweets then we applied POS tagging.

Now we use only those words which are tagged with Noun, verb, adverbs, VBD.

After that I have applied Hypernym to these words

Finally use set function to make it unique dictionary.

After getting Bag-of-words of each sentiment, and feature vector for each tweet we calculated probability of each tweet and finally assign that sentiment which has maximum probability.