Humor Detection

Submitted to Harrisburg University for Science and Technology as a Partial fulfillment of the Requirements for the Master of Science degree in Computer Science

BY:

Manan Jain

manan.jain149@gmail.com

Supervised by

Nushwan Al-Nakash, PhD.

Assistant Professor of Computer Science

nal-nakash@harrisburgu.edu

August 2017

Table of Contents

ABSTRACT

Humor is a very complex characteristic concept that defines us as human beings and social entities. Humor is an essential component in personal communication.
How to create a method or model to discover the structures behind humor, recognize humor and even extraction of humor remains a challenge because of its subjective nature. Humor also provides valuable information related to linguistic, psychological, neurological and sociological phenomena. However, because of its complexity, humor is still an undefined phenomenon. Because the reaction that make people laugh can hardly be generalized or formalized. For instance, cognitive aspects as well as cultural knowledge, are some of the multi-factorial variables that should be analyzed in order to understand humor's properties.

Although it is impossible to understand universal humor characteristics, one can still capture the possible latent structures behind humor. In my work, I will try to uncover several latent semantic structures behind humor, in terms of meaning incongruity, ambiguity, phonetic style and personal affect. In addition to humor recognition, identifying anchors, or which words prompt humor in a sentence, is essential in understanding the phenomenon of humor in language.

Proposed technique is created using the concepts of linguistics and it has significant accuracy of over 70+% compared to 23.06% of Word Index power method.

Keywords: Humor Detection, Natural languages, Computational linguistics, Computational modeling.

Chapter 1
Humor Detection

1.1 Introduction

Humor is a vital component of human well-being. Neuroimaging studies conducted with adults indicate that humor activates specific brain regions, including the temporo-occipito-parietal junction (TOPJ), involved in incongruity resolution, and mesolimbic regions, involved in reward processing. However, no study to date has used neuroimaging to examine humor [1].

Humor also provides valuable information related to linguistic, psychological, neurological and sociological phenomena. However, because of its complexity, humor is still an undefined phenomenon. Because the reaction that make people laugh can hardly be generalized or formalized. For instance, cognitive aspects as well as cultural knowledge, are some of the multi-factorial variables that should be analyzed in order to understand humor's properties. Despite such in-conveniences, different disciplines such as philosophy, linguistics, psychology, or sociology, have attempted to study humor in order to provide formal insights to explain better its basic features. From a psychological point of view, the analysis of the relationship between personality and humor appreciation, providing interesting observations about this perspective, and about the type of necessary stimuli required to produce a response. One the other hand, some linguistic studies have tried to explain humor by means of semantic and pragmatic patterns.

The problem of detecting and differentiating between humor and spam is widely used by social networking websites like Facebook and twitter. So, if the problem of humor detection is solved it might help users on Facebook to easily distinguish between fake spams and humorous contents.

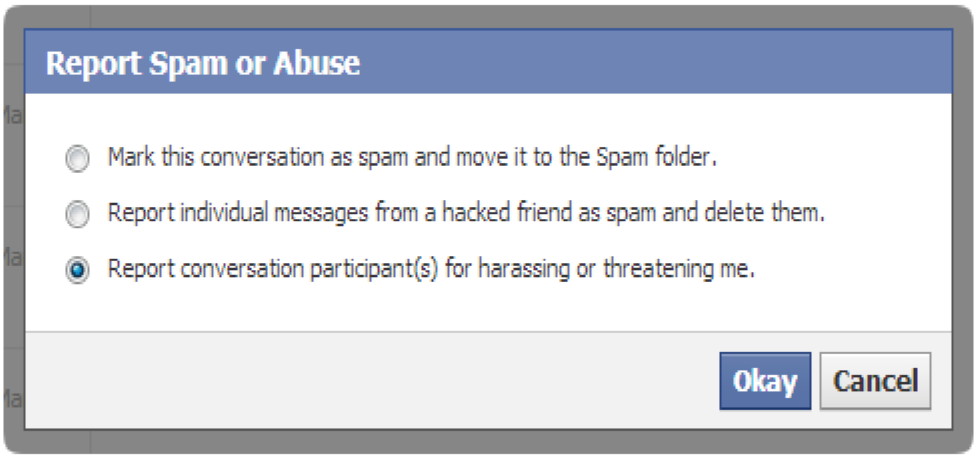

Figure 1.1: Typical Facebook Report Spam box

Approximately 500 million tweets are being shared on twitter out of which nearly 200+ million are humorously structured. Being a global platform as per our definition of humor not every funny stuff is accepted. So, everyday twitter receives around 15+ million spam reports. So, it very difficult to find spam without a proper humor detection algorithm because humor is highly subjective.

Definition of Humor is so complex being the reason that problem till date doesn’t has any solution.

Figure 1.2 A Small funny conversation [2]

After examining Figure 1.2 we really can't say whether it is humorous, offensive or a simple statement. This is the kind of problem which is faced when solving the problem [2].

Most work on humor detection approaches the problem as binary classification: humor or not humor. While this is a reasonable initial step, in practice humor is subjective, so we believe it is interesting to evaluate different degrees of humor, particularly as it relates to a given person's sense of humor.

1.2 Basic Definition of Humor

Humor is the use of cognitive experiences to provoke laughter and provide amusement (Figure 1.3). So, Humor detection is hard in absence of cognitive experience i.e. experiences from your past.

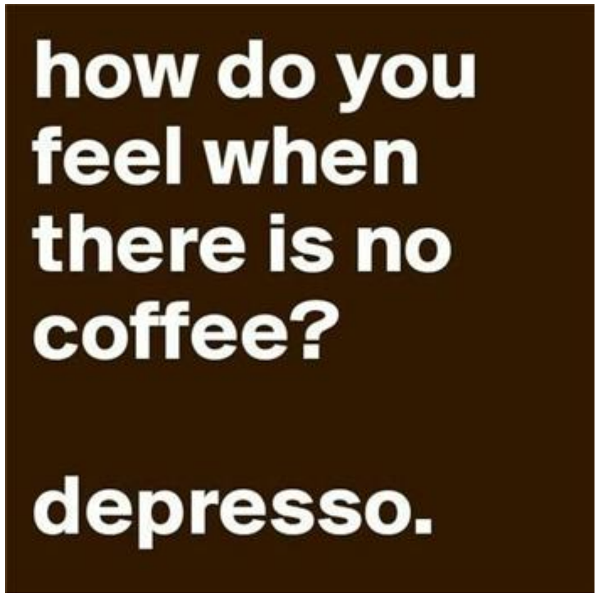

Figure 1.3 A common meme of Starbucks [2]

Second point in humor definition can be stated like this it is highly subjective for example it can be humorous and a piece of laughter to somebody but quite offensive to someone else [3]. For e.g. as shown in Figure 1.4. This image is quite humorous to Germans but at the same time for brazil people it is quite offensive and rude because of their 7-1 world cup semi-final loss.

Figure 1.4 A common meme but offensive for certain set of people [3]

Third and main point is definition of humor differs from people to people. For ex based on our interests likes dislikes etc. Figure 1.5 is quite funny for harry potter fans but those who don't share this line of interest it is a normal meme.

Figure 1.5 Harry Potter meme humorous to specific set of audience [3]

Hence it is clear that humor detection is a challenging natural language problem. First, a universal definition of humor is hard to achieve, because different people hold different understandings of even the same sentence. Second, humor is always situated in a broader context that sometimes requires a lot of external knowledge to fully understand it. For example, consider the sentence, "The one who invented the door knocker got a No Bell prize".

Although it is impossible to understand universal humor characteristics, one can still capture the possible latent structures behind humor. In my work, I will try to uncover several latent semantic structures behind humor, in terms of meaning incongruity, ambiguity, phonetic style and personal affect. In addition to humor recognition, identifying anchors, or which words prompt humor in a sentence, is essential in understanding the phenomenon of humor in language.

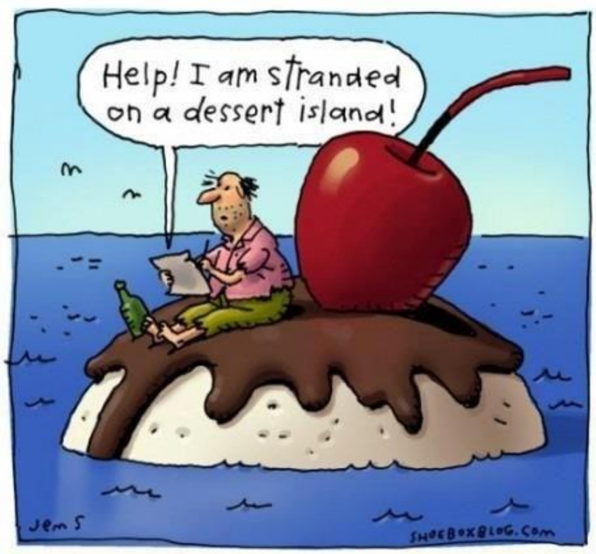

Figure 1.6 A humorous statement with intentional grammatical error[4]

One of the biggest challenge is detection between grammatical error and humor. Figure 1.6 illustrates this by comparing deserted and dessert [4].

So, the research work is aimed towards coming up with a technique that will help in deducing whether the sentence is humorous or not simply based on the structure of the sentence. It is a quite difficult job to achieve since there is no real system to achieve this feat and to compare the results with. So, some of the selfmade datasets are used for running and testing the results and meanwhile using a lot of datasets from twitter is also being used.

1.3 Analysis of Problem Statement

Now, after defining the problem, we should discuss why this problem should be solved and who all are affected with this problem. So, the basic problem is that there is no definition of humor therefore we need to formulate one in order to proceed in the problem this is achieved by following the steps in the proposed algorithm later in Chapter 3.

Now, the question arises why the problem needs to be solved. This problem of humor detection is being constantly faced by twitter(Fig 7) in differentiating between spam and humor plus a lot of other social networking websites. For instance, On approximately 500 million tweets are being shared on twitter out of which nearly 200+ million are humorously (one liner) structured. Being a global platform as per our definition of humor not every funny stuff is accepted. So,

everyday twitter receives around 15+ million spam reports. So, it very difficult to find spam without a proper humor detection algorithm because humor is highly subjective [5].

Also, the problem arises in deception detection i.e. differentiating between funny and fake reviews on yelp (Figure 1.8), amazon etc [5].

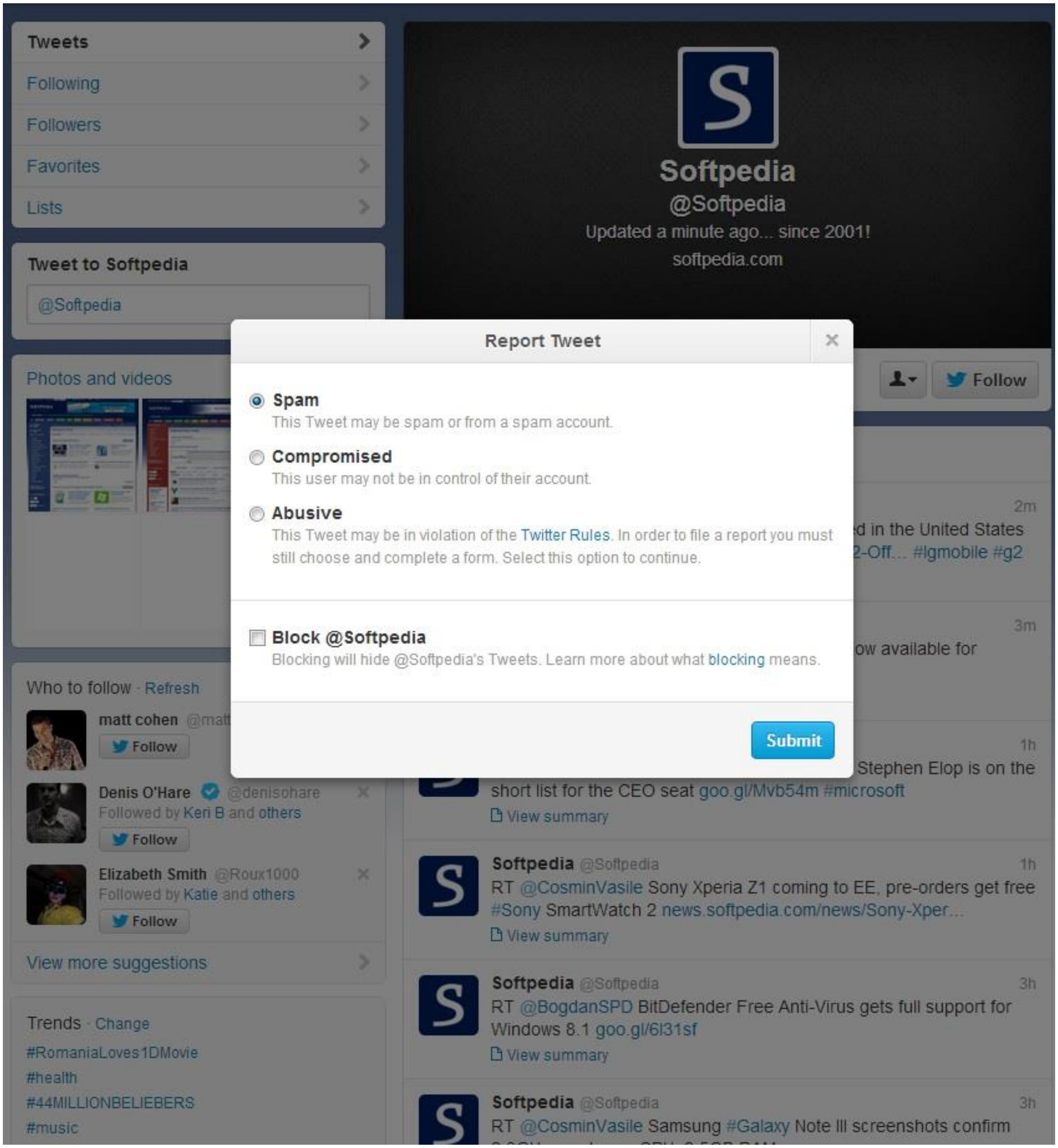

Figure 1.7 Typical Report a spam page on twitter.

Figure 1.8 [5]

Figure 1.9 demonstrates a small and simple model of how the algorithm will work. I will take a sentence as an input and it will determine whether sentence is humorous or not in probabilistic way ie if there is a chance of the sentence being humorous or not or whether there is an absolute chance of it being humorous or no chance at all.

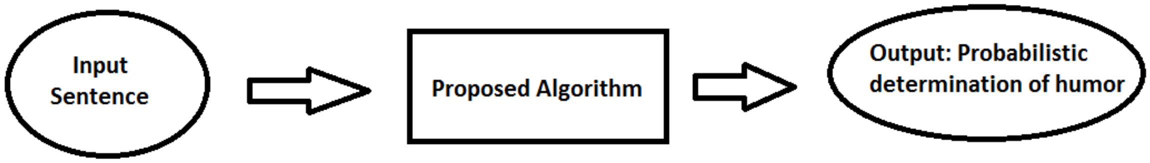

Figure 1.9: Simplified 3 Phase diagram of proposed algorithm

Algorithm used by Twitter[6] is dependent on word powers where if a specific word is there in the sentence than probability of it being a spam increases or it being humorous increases, which is not correct all the time. So the problem with the algorithm(of twitter etc) can be solved by figuratively trying to formulate the meaning of humor on the basis of the concepts of linguistics and structure of sentence and implementing it using the proposed algorithm.

Hence this area of Humor Detection requires extensive research starting from building a small cognitive experience approach to basics of a sentence creation so as to generalize a small humor definition or detecting a pattern or creating a model from which humor detection is possible. And possibly be able to tackle further big problem Sarcasm Detection in the future (Chapter 5).

Chapter 2

Theoretical Background

2.1 ANALYSIS OF RELATED WORK

In Chapter 1, we defined the problem and the necessity of why its needs to be solved and who will be affected by the solution of the problem. A tremendous amount of research is going on not only in the field of computer science but also in linguistics and psychology. Frontrunners are the Word2Vec, Twitter word indexing power, Mihalcea and Strapparava‘s dataset of puns and humorous one-liners intended for supervised learning.

2.1.1 Word2Vec:

Word2vec creates vectors that are distributed numerical representations of word features. Features such as context of individual words which detects similarities mathematically by grouping the vectors of similar word. Based on this approach twitter created word indexing power approach [7].

Word2Vec approach search for clusters of words in the sentence to already humorous sentences if the scanned word cluster is present than sentence can be judged humorous.

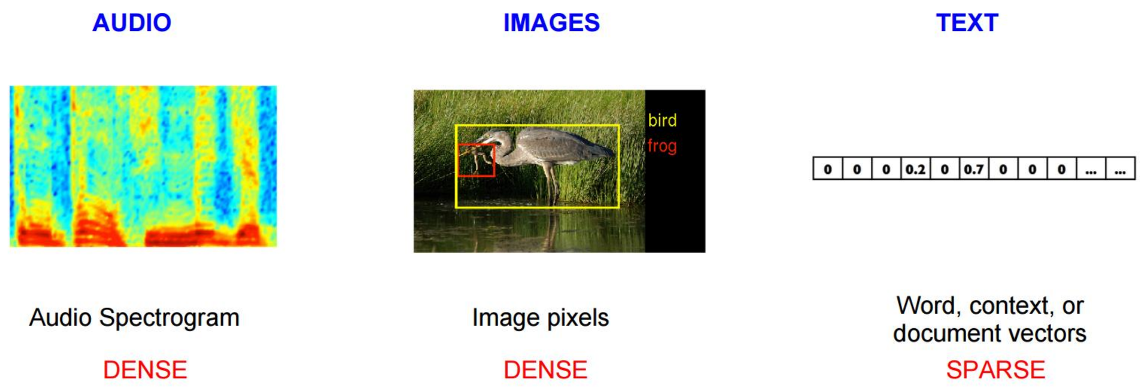

Figure 2.1: Word2Vec Representation [7]

Word2vec is a particularly computationally-efficient predictive model for learning word embeddings from raw text. It comes in two flavors, the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model (Section 3.1 and 3.2 in Mikolov et al.). Algorithmically, these models are similar, except that CBOW predicts target words (e.g. 'mat') from source context words ('the cat sits on the'), while the skip-gram does the inverse and predicts source context-words from the target words. This inversion might seem like an arbitrary choice, but statistically it has the effect that CBOW soothes over a lot of the distributional information (by treating an entire context as one observation). For the most part, this turns out to be a useful thing for smaller datasets. However, skip-gram treats each context-target pair as a new observation, and this tends to do better when we have larger datasets [7].

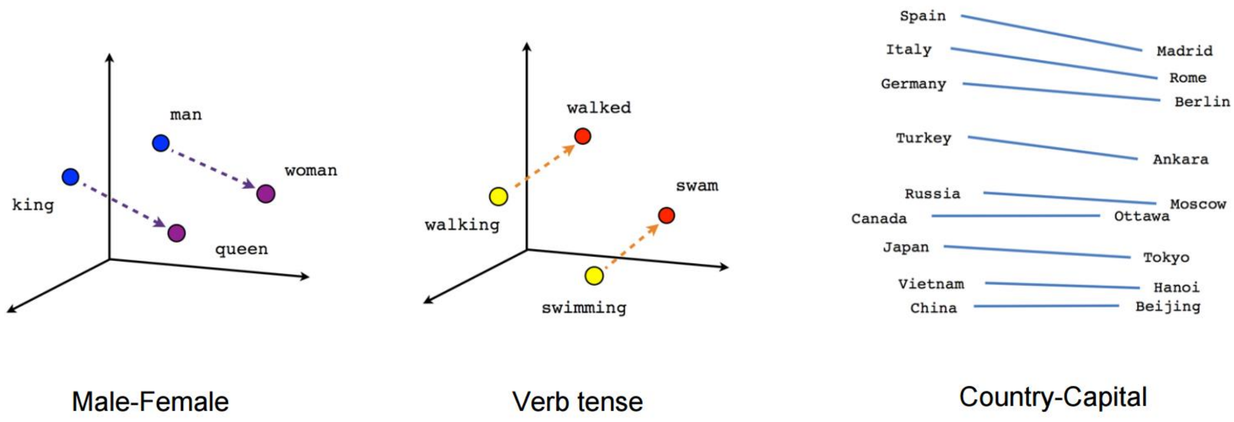

Figure 2.2: Words Representation [7]

Figure 2.2 explains how the words are represented and assigned to each other for ex. King and queen, man and woman. Same way this technique connects the popular funny words in a sentence and tries to come up with a small structure and when the structure is found the sentence is judged humorous.

2.1.2 Twitter Word Indexing Power:

Twitter daily gets around 10 million spam reports which are being handled using the word power.

η = Sum of all word Indexes in the sentence / Total number of Words in the sentence.

η is humor factor. If η is greater than the threshold value than the sentence is judged humorous. This threshold values varies from language to language and area to area.

Following are the most common word powers on twitter [8].

(’kidding’, 0.8124330043792725),
(’heck’, 0.7966572642326355),
(’wtf’, 0.7851443290710449),
(’hell’, 0.7803547382354736),
(’what’, 0.7729526162147522),
(’anyways’, 0.7656563520431519),
(’why’, 0.7625856995582581),
(’mean’, 0.7603179812431335),
(’ya’, 0.7591337561607361),
(’yeah’, 0.7561845779418945)
’awesome’,0.8051108121871948),
(’amazing’, 0.8016425967216492),
(’omg’, 0.7824660539627075),
(’def’, 0.7814326286315918),
(’soooo’, 0.7789645195007324),
(’:)’, 0.770503044128418),
(’sooooo’, 0.7704325318336487),
(’sooo’, 0.7700715065002441),
(’:))’, 0.7666398286819458),
(’soo’, 0.7654560804367065)

This method has its own drawbacks like it does not focus on the structure of sentence at all and it just scans for these specific words and judge a sentence based on the words present in the sentence. This is discussed in Chapter 4 [8].

2.1.3 Mihalcea and Strapparava's dataset of puns and humorous one-liners:

Researchers Zhang and Liu have developed a dataset similar to dataset based on the New Yorker Caption contest (NYCC) (Radev et al., 2015; Shahaf et al., 2015). While for the HW viewers submit a tweet in response to a hashtag, for the NYCC readers submit humorous captions in response to a cartoon. It is important to note this key distinction between the two datasets, because of the presence of the hashtag allows for further innovative NLP methodologies aside from solely analyzing the tweets themselves. In Radev et al. (2015), the authors developed more than 15 unsupervised methods for ranking submissions for the NYCC. The methods can be categorized into broader categories such as originality and content-based [9].

Zhang and Liu (2014) constructed a dataset for recognizing humor in Twitter in two parts. First, the authors use the Twitter API with target user mentions and hashtags to produce a set of 1,500 humorous tweets. After manual inspections, 1,267 of the original 1,500 tweets were found to be humorous, of which 1,000 were randomly sampled as positive examples in the final dataset. Second, the authors collect negative examples by extracting 1,500 tweets from Twitter Streaming API, manually checking for the presence of humor. Next, the authors combine these tweets with tweets from part one that were found to actually not contain humor. The authors argue this last step will partly assuage the selection bias of the negative examples [9].

Alternatively, the approach with a supervised model, evaluating on a pairwise comparison task, upon which we base our evaluation methodology. The features to represent a given caption fall in the general areas of Unusual Language, Sentiment, and Taking Expert Advice. For a single data point, the authors concatenate the features of each individual caption, as well as encoding the difference between each caption's vector. The authors' best-performing system records a 69% accuracy on the pairwise evaluation task. Note that for this evaluation task, random baseline is 50%. Therefore, the incremental improvement above random guessing dictates the difficulty of predicting degrees of humor [10].

2.2 Theoretical Idea of proposed work

Although it might be impossible to understand universal humor characteristics, one can still capture the possible latent structures behind humor. In the proposed work, several latent semantic structures behind humor are being uncovered, in terms of meaning incongruity, ambiguity, phonetic style and personal affect. In addition to humor recognition, identifying anchors, or which words prompt humor in a sentence, is essential in understanding the phenomenon of humor in language.

The proposed technique is based on understanding the semantics of a sentence and is targeted towards one liner sentences. So, the proposed methodology will try to understand the semantics by searching for funny emojis and funny words in the sentence if these are found than there is good chance of a sentence being humorous.

After this step, we will try to find weird comparison between subject and object of a sentence. If there is a presence of weird comparison of a subject e.g. a human to something inappropriate than the chance of sentence being humorous increases.

We can also look for weird combinations when a subject can be compared to something which is out of bound and thus will lead to humor content because of a funny comparison between the subject and the object for example you have ten eyes. So theoretically if something has a bound and it is being changed than it can cause amusement which is being taken care of in the proposed technique.

The difficult part is to detect a shift in tone in a sentence which according to the linguistics of a sentence will lead to amusement. Shift in tone means first half of the sentence is positive and second half of the sentence is negative or vice versa. In this case the sentence has a high probability of being humorous. Plus, the sentence can be sarcastic too which is discussed in chapter 5.

Antonym pairs in a sentence also leads to humorous work. The semantics of a sentence in this case will be organized to support one word and the remaining part will support the other antonym pair of the same word present in the sentence. For e.g. Old and young etc. in a single sentence [11].

But apart from these tweets are also targeted to specific person so the proposed technique will also look for weird position uppercase words in the sentence or unnecessary symbols.

2.3 Issues in proposed technique

Detection of humor is a very difficult task because of its subjective nature. The proposed method is based on the dataset it is using so if a funny word is scanned which is not there in the dataset than even if sentence was to judge humorous the result would be different. Also, the above theoretical ideas generally work for humorous sentences but they do not work all the time for example for a particular sentence even if there is a shift of tone it can still might not sound humorous. So. the problem with humor is that it is very tricky to judge but with a correct dataset and techniques the result can be close to ideal solution just like predictive models of weather forecasting which is not always correct. But is quite close to the accuracy.

Chapter 3
Proposed Solution Approach

3.1 Introduction

The suggested approach is designed to work on one-liner sentences which generally comprises of 80% of the input patterns of tweets on the twitter. Output yield the probability of a sentence being humorous or not for e.g. the sentence is humorous or not or it might be humorous or it can be humorous etc.

The approach is procedure based so all the steps will be visited at least once based on the result calculated at each step, the sentence will be judged humorous or not accordingly. Figure 3.1shows a simplified 3-Phase diagram of solution approach.

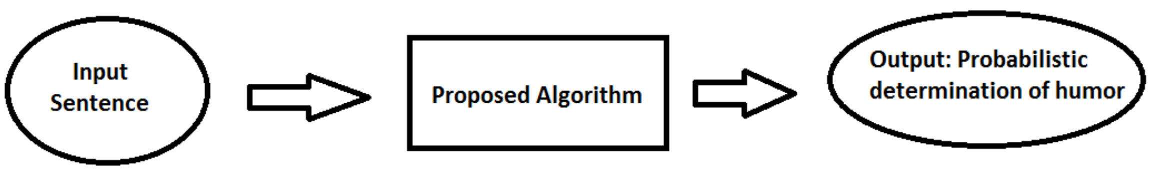

Figure 3.1: Simplified 3-Phase diagram of solution approach

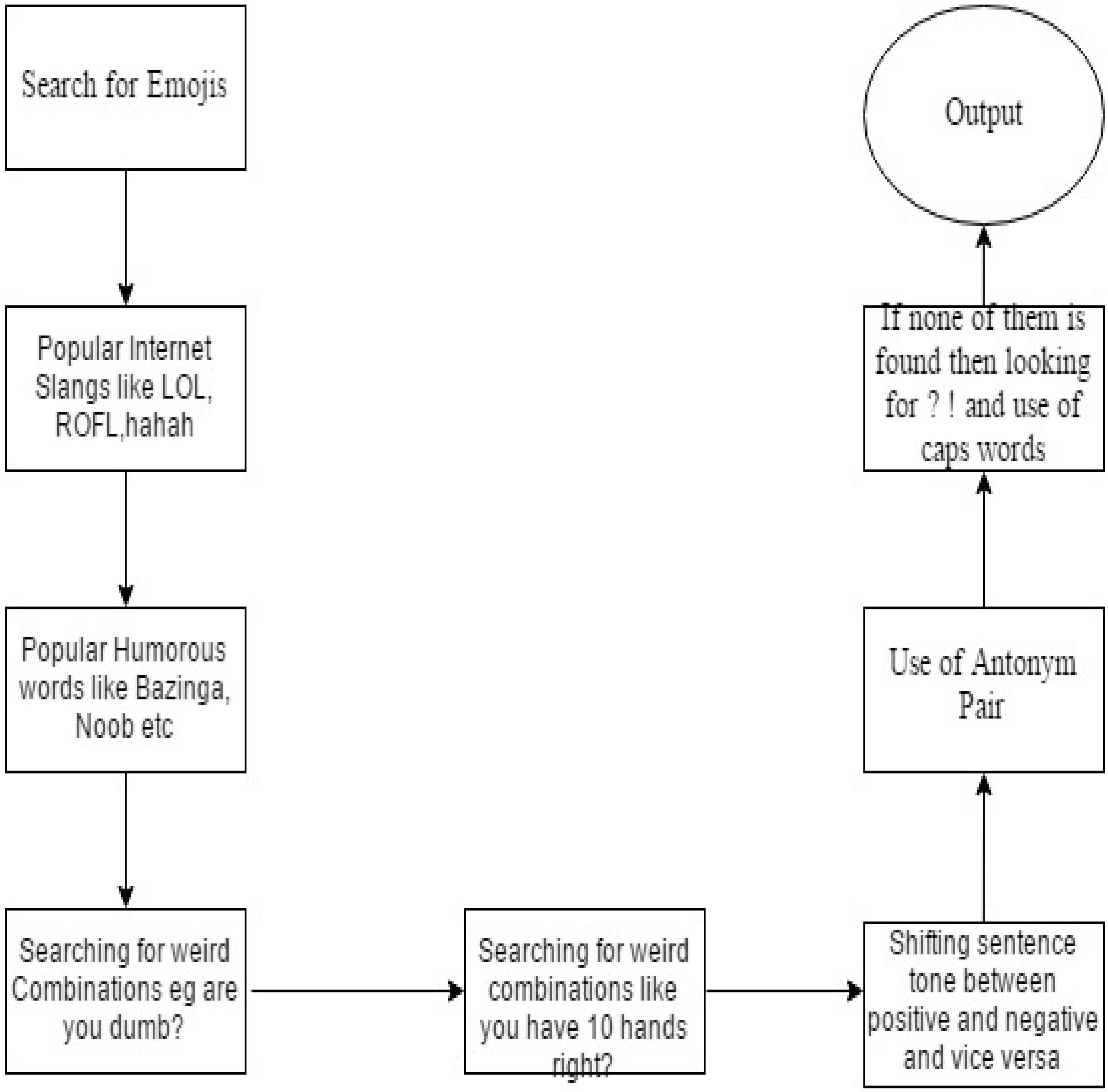

Figure 3.2 shows a block diagram of the suggested method working mechanism.

Referring to Figure 3.2:

Step 1. Fig 3.2 describes the solution approach the algorithm takes input one liner sentence and it will first search for the funny emoji's like :D :) :P o.O ;) >:O ^_^ 8-) 8| :v :3. So if it has any of these emoji's there is high probability of being humor content in it.

Step 2. Next step is to search for popular Internet funny slangs which contributes a lot of humorous stuff. For e.g. bamboozled, bazinga, bevy, buccaneer, bulgur, bumfuzzle, canoodle, coot etc.

Step 3. Searching for weird comparisons in the sentences like comparison of a subject with idiot, jerk, blind, deaf, dumb. Which again results in a humorous sentence.

Step 4. Searching for weird combinations in the sentence like comparing a subject with inappropriate amount of stuff like age to more than 200, hands more than 2 etc.

Step 5. Searching for Shift in tone i.e. positive to negative and vice versa which also helps in detection of sudden pinch of sarcasm also.

Step 6. One of the most classic example of humorous sentences contains pair of antonyms like young old, lively dying etc. So, sentences with presence of antonym pair again contributes to huge percentage of it being humorous.

Step 7. If none of them are found then look for user specific stuff which generally are referred in bold letter or caps. For e.g. funny nicknames.

3.2 Algorithm

Input: A simple one liner Sentence one at a time.

Output: Judging whether the sentence is humorous or not. Plotting all the results as a graph and displaying the results for comparison.

1. **We create a list of funny-emoji's ie femojis using the dataset containing list of funny emojis from twitter.**

```
femoji=open("Emojis.txt","r")
```

where Emojis.txt contains the list of funny emojis

```
l=[]


femojis=""
l=femoji.readlines()

for i in range(len(l)):
    femojis=femojis+l[i]
```

```
femojis=femojis.split()
```

Using these steps, we can create femojis which contain a list of funnyemojis.

Using the **regular expression** e, we select only words no spaces or any other symbols and replace them with a **space**.

```
e=re.compile(r'\b\w+?\W*\w+\b')
eemojis=e.sub(" ",file)
```

Now eemojis contains everything except the annotations.

```
if eemojis:
    eemojis=eemojis.split()

for i in eemojis:
    for w in femojis:
        if i==w
                //Funny emojis are there in the sentence
```

So, if there is any similarity between the list eemojis and femojis than there is a funny emoji present in the sentence. And we are done.

2. **If step 1 returns false than we check for *typical Internet slangs*.**

```
fabbrev_lol=open("Abbrev_lol.txt","r")
```

Similar to step 1 fabbrev_lol contains the dataset text which contains funny internet slangs like lol etc

```
l=[]
fabbrev_lols=""
l=fabbrev_lol.readlines()
```

Now creating a dictionary of l which will contain every word there is in the dataset

```
for i in range(len(l)):
    fabbrev_lols=fabbrev_lols+l[i]
fabbrev_lols=fabbrev_lols.split()
```

Now using regular expressions, we remove any sort of annotations or punctuations from the sentence we input.

```
e=re.compile(r'[><^@$%*_+=?;!&*)(":\'-.]')

eabbrev_lol=e.sub("",file)
eabbrev_lols=eabbrev_lol.split()
```

We split the sentence on spaces after clearing the annotations.

Now we compare whether any word in the sentence is similar to the word present on the funny list on internet.

```
for i in eabbrev_lols:
    for w in fabbrev_lols:
        if i==w:
```

//Internet slangs are there in the sentence

So, there is a good probability of sentence being humorous.

3. **If step 2 returns false than Considering the Funniest words in English like *bazinga, noob, dumb* etc.**

```
ffunny_word=open("funny_words.txt","r")
   l=[]
   ffunny_words=""
   l=ffunny_word.readlines()
   for i in range(len(l)):
       ffunny_words=ffunny_words+l[i]
   ffunny_words=ffunny_words.split()
```

Creating a list of funny words in English and storing in **ffunny_words**

```
   efunny_words=file.lower()
   efunny_words=efunny_words.split()
```

Now comparing the words in the sentence with the list created above.

```
   for i in ffunny_words:
       for w in efunny_words:
           if i==w:
                   //Funny Words are there in the sentence
```

So, there is a good probability of sentence being humorous.

4. **Finding the *antonym pairs* in the sentence.**

   Using the dataset Antonym.txt and comparing the sentence with various rules defined. If sentence has an antonym pair than there is a hint of sarcasm in it.

```
fantonym=open("Antonym.txt","r")
l=[]
fantonym1=""
l=fantonym.readlines()
for i in range(len(l)):
    fantonym1=fantonym1+l[i]
fantonym1=fantonym1.lower()



e=re.compile(r'—')
fantonyms=e.sub(" ",fantonym1)
```

Now **fantonyms** contains pair of antonyms as a list.

```
fantonym=re.findall(r'\b(\w+)\s(\w+)\b',fantonyms)

eantonym=file.lower()
eantonym=eantonym.split()
```

Now comparing if a pair of antonyms exist in a sentence or not.

```
for i in fantonym:
    for w in eantonym:
        if w==i[0]:
            if i[1] in eantonym:
```

//Antonym pair is there in the sentence. High probability of the sentence being sarcastic.

5. **Finding the bounds i.e. comparison of the subject with something impossible like I have 10 hands.**

Using the dataset Bound.txt and comparing the sentence where the subject is there with some number of bounds possible like hands and 10 etc.

```
fbound=open("Bound.txt","r")
l=[]
fbounds=""
l=fbound.readlines()
```

Implementing the subject and checking the bound of the subject in the variable **fbounds.**

```
for i in range(len(l)):
    fbounds=fbounds+l[i]
```

```
fbounds=fbounds.lower()
fbound1=re.findall(r'(\w+)/(\w+)',fbounds)

ebound=file.lower()
ebound=ebound.split()

for i in fbound1:
    for w in ebound:
        if i[0]==w:
            if i[1] not in ebound:
                //If the bound specified in the sentence is not normal than the
sentence can be humorous.
```

6. **Finding the change of tone in the sentence ie *positive to negative or happiness to sadness and vice versa.***

Using the dataset Positive_words.txt, Negative_words.txt and comparing the sentence with various rules defined.

```
fpositive=open("Positive_words.txt","r")
l=[]
fpositives=""
l=fpositive.readlines()
for i in range(len(l)):
    fpositives=fpositives+l[i]
fpositives=fpositives.lower()
fpositives=fpositives.split()
```

```
fnegative=open("Negative_words.txt","r")
l=[]
fnegatives=""
l=fnegative.readlines()
for i in range(len(l)):
    fnegatives=fnegatives+l[i]
fnegatives=fnegatives.lower()
fnegatives=fnegatives.split()
```

Creating a big list of positive and negative words in variable **fpositives** and **fnegatives** respectively.

```
eposneg=file.lower()
eposneg=eposneg.split()
```

```
i1=0
i2=0
```

i1 and i2 are used as flags

```
for i in eposneg:
    for w1 in fpositives:
        if i==w1:
            i1+=1
    for w2 in fnegatives:
        if i==w2:
            i2+=1
```

```
if i1 and i2:
    if i1-i2==0:
        //Number of positive and negative words are equal in the sentence which hints at big tone of sarcasm.

    if abs(i1-i2 and i1!=0 and i2!=0)==1:
        // If they are within one short of each other than sentence might be sarcastic.
```

7. **Finding the weird comparisons in the sentence for example comparison of the subject with some funny word like are you a nerd?**

Using the file Comparison.txt and comparing the sentence with various rules defined.

```
fcomparison=open("Comparison.txt","r")
l=[]
fComparisons=""
l=fcomparison.readlines()
for i in range(len(l)):
    fComparisons=fComparisons+l[i]
fComparisons=fComparisons.lower()
fComparisons=fComparisons.split()
```

Creating a list of words with subject is being compared.

```
ecompare=file.lower()
```

Using regular expression finding which words subject is being compared to in the sentence.

```
e=re.search(r'[?!]',file)
if e:
    e=re.compile(r'[?!]')
    ecompare=e.sub("",ecompare)

    e=re.search(r'\bare you\b',ecompare)
    if e:
        ecompare=ecompare.split()
        for i in ecompare:
            for j in fComparisons:
                if i==j:
                    //If the word subject is being compared to in the list than the
sentence can be judged humorous.
```

8. **If none of the steps yield a concrete result than searching for caps words which refers to a specific person like it can be a nickname to a person and might be funny for a local set of people.**

Using regular expression finding the caps word in the sentence

```
e=re.search(r'[?!]',file)
if e:
    file=file[0].lower()+file[1:]
    e=re.search(r'[A-Z]',file)
```

if e:

//If there are certain caps word than sentence might be humorous with very a low probability.

9. Algorithm asks for another sentence and go to Step 1.

10. Plot the result of all the sentences and comparing them. (Algorithm Ends)

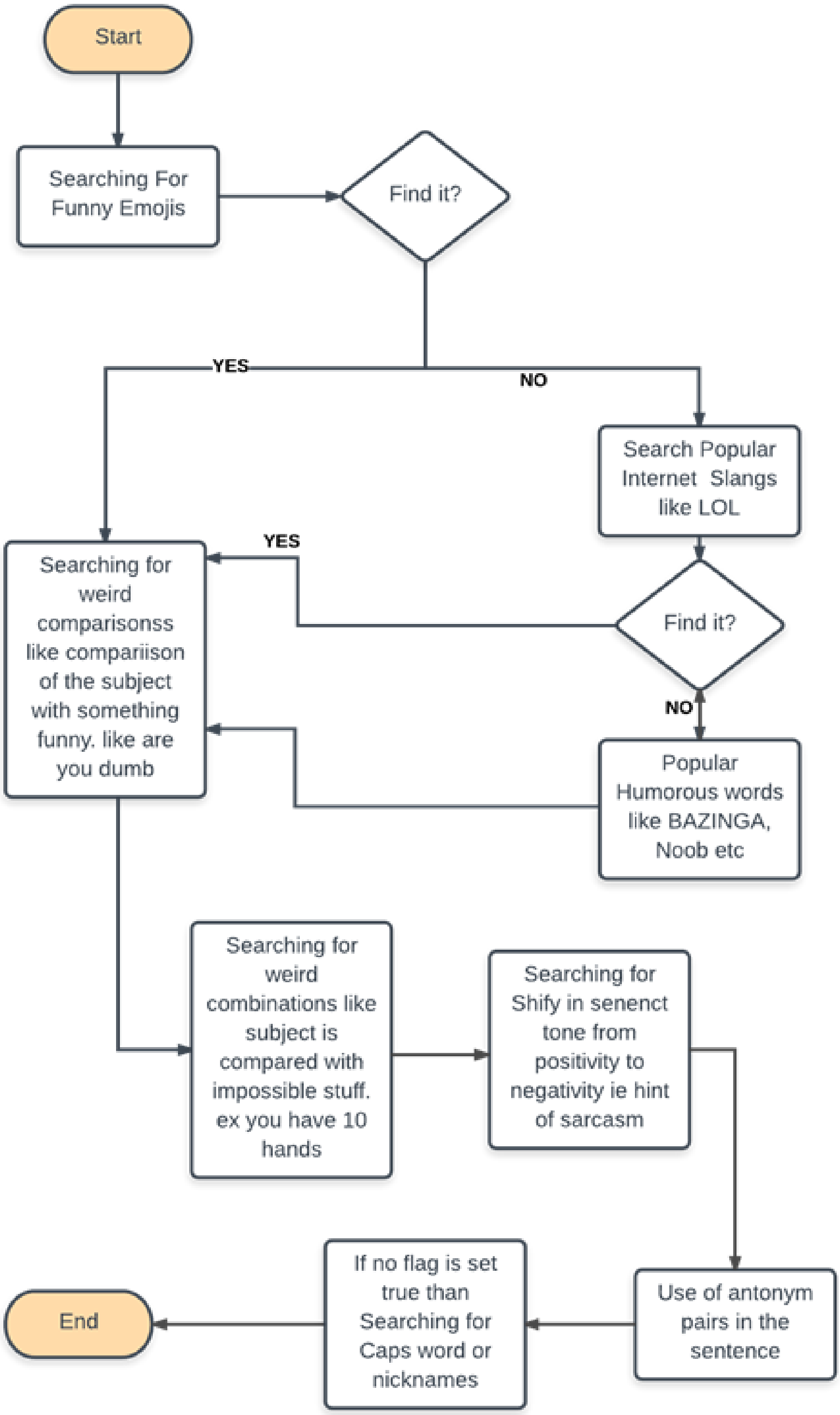

Figure 3.3: Flow Chart of Solution Approach for just one iteration

Chapter 4
Experimental Results

4.1 Introduction

Proposed Algorithm uses the concepts of semantics, linguistics and syntax of the sentence to judge whether a sentence should be judged humorous or not. The algorithm is designed to work specifically for one liners on twitter to check whether they are humorous or not.

The Proposed Algorithm uses a lot of datasets like antonym pairs, funny emojis, internet slangs etc. to work so, the results are highly dependent on the datasets and therefore high quality of the datasets will lead to more precise results.

The proposed algorithm uses the input datasets as shown in section 3.2. For example dataset containing a list of emojis like :D :) :P o.O ;) >:O ^_^ 8-) 8| :v :3 etc. a list of funny English words like dumb, noob, doohickey, eschew, fiddledeedee, finagle, flanker, floozy, fungible, Girdle, gobsmacked, a big list of positive words, negative words and antonym pairs etc.

Figure 4.1 contains typical funny words in English like panache, nitwick, noob etc. So, if the dataset of funny words covers most of the funny words in English then the probability of results being accurate increases also with it. Same goes for other datasets also if their accuracy increases hence the probability of getting a better result will be there.

This is true because the algorithm will search for key words extracted by the regular expression in the datasets available. For example, let's say if the keyword extracted by RE is not found in the datasets than chances of the sentence being humorous decreases. So, if the dataset is small and inaccurate it will reduce the accuracy of algorithm.

Proposed algorithm can be compared to the word indexing power (discussed in Chapter 2). This method just scans for all the words in the sentence and calculate the average of the humor factor of each word.

$$\boldsymbol{\eta = (\text{Sum of all word Indexes / Total number of Words})}$$

If η is greater than a particular humor factor than the sentence is judged humorous. η is variable and can be changed according to context.

bamboozled
bazinga
bevy
bifurcate
bilirubin
bobolink
buccaneer
bulgur
bumfuzzle
canoodle
cantankerous
carbuncle
caterwaul
cattywampus
cheeky
conniption
coot
didgeridoo
dingy
doodle
gay
asshole
dickhead
dumb
noob
doohickey
eschew
fiddledeedee
finagle
flanker
floozy
fungible
girdle
gobsmacked
grog
gumption
gunky
dodecahedron
doodad
doohickey
doppelganger
dumbfounded
ebullient
effervescence
egads
eggcorn
ephemeral
extraterrestrial
finagle
fandango
festooned
fez
filigree
firebrand
fisticuffs
flabbergasted
flapdoodle
flibbertigibbet
flimflam
flotsam
flummoxed
flyaway
flyspeck
foofaraw
foolhardy
foolscap
fopdoodle
fortuitous
fracas
frangipani
freewheeling
fricassee
frippery
froufrou
miasma
milquetoast
misanthrope
mishmash
moocher
mojo
mollycoddle
monsoon
muckety-muck
mulligatawny
murmuration
nincompoop
nitwit
nomenclature
noodge
nudnik
numbskull
onomatopoeia
orotund
oxymoron
pachyderm
pagoda
palindrome
palomino
panache
pandemonium
pantaloons
parabola
parallelogram
pedagogue
pell-mell
persimmon
persnickety
pettifogger
phalanx
phantasmagorical
phylactery

Figure 4.1 Dataset for funny words in English

4.2 Results and Comparisons

The following are some examples of test sentences to be interrogated for being humorous:

Sentence 1 You think I was studying lol

Sentence 2 I used to love my mind now I hate it the most

Sentence 3 are you nerd?

Sentence 4 Your friend has 10 hands he is always studying

Sentence 5 go to hell man

```
Enter the Sentence for detecting the humor :You think I was studying lol.
Sentence is Humorous
Press 1 to continue :1
Enter the Sentence for detecting the humor :I used to love my mind now I hate it the most
Sentence is Humorous
Press 1 to continue :1
Enter the Sentence for detecting the humor :are you nerd?
Sentence is Humorous
Press 1 to continue :1
Enter the Sentence for detecting the humor :Your friend has 10 hands he is always studying
Sentence is Humorous
Press 1 to continue :1
Enter the Sentence for detecting the humor :go to hell man
Sentence Doesn't seem to have a humorous content
Press 1 to continue :2
```

Figure 4.2 Results of Sentences 1-5

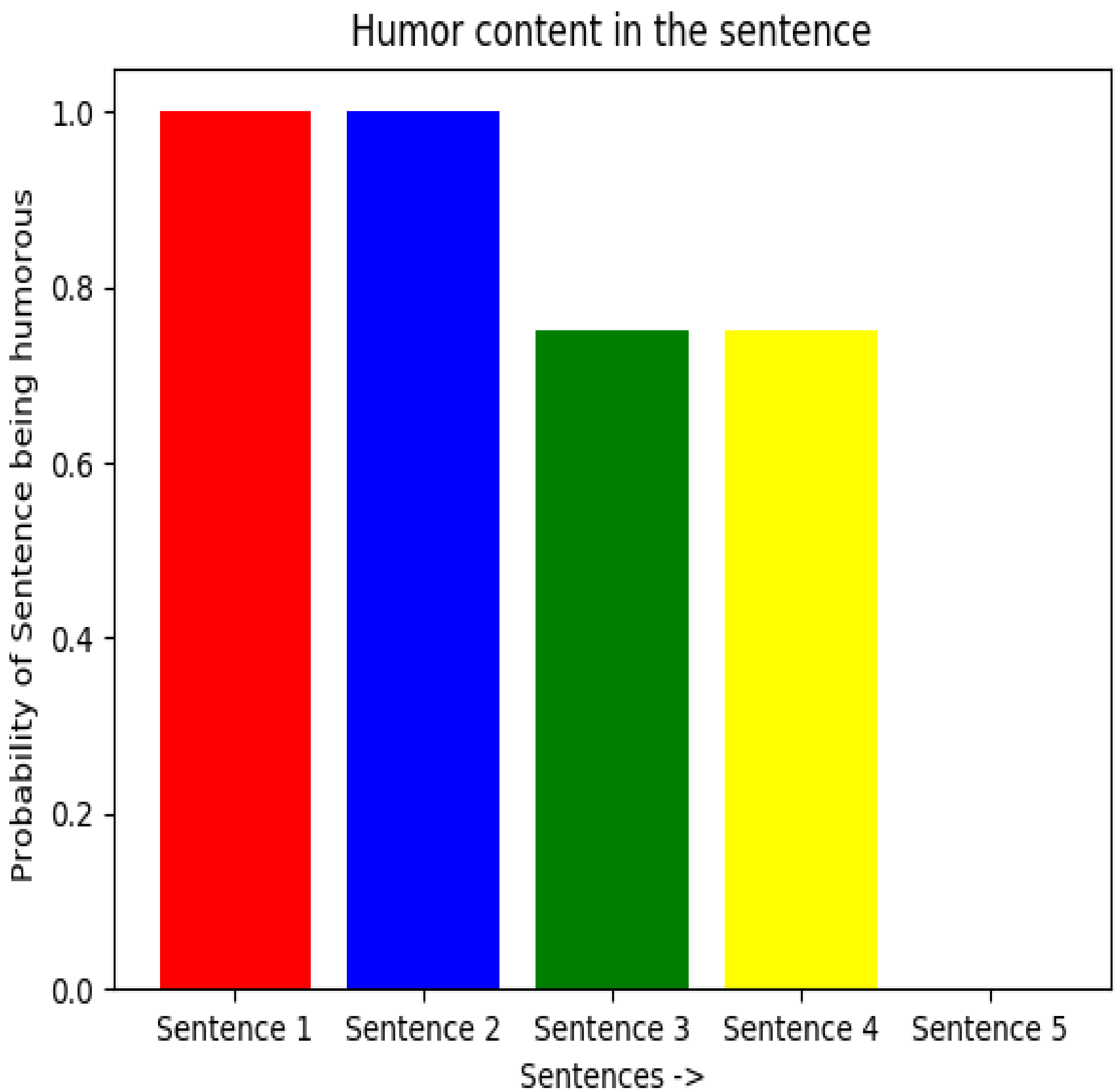

Figure 4.3 Bar Chart describing probability of sentence being humorous

As shown in Figure 4.3, it describes the probability of a sentence being humorous. **Sentence 1** is clearly humorous with an internet slang like lol. So, there is a high probability i.e. 1 of it being humorous.

Sentence 2 is also humorous with a strong description of variations of semantics.

Sentence 3 has a good probability of being humorous but again it can be offensive too. So, probability 0.75.

Sentence 4 has a sarcastic tone in it which increases the chance of it being humorous.
So, probability is 0.75.

Sentence 5 doesn't seem to have humor content. It is offensive only.

4.2.1 Comparison with Word Indexing power.

Word indexing (η), *as discussed in chapter2*, is the average of humor content, related to all the words in a given sentence.

η = (Sum of all word Indexes / Total number of Words)
If η is greater than a particular humor factor than the sentence is judged humorous.

Word Indexing will work for all the sentence except sentence 4 which has a sarcastic tone while the remaining sentences has some sort of funny words in them. But proposed technique will be able to work on Sentence 4 because the technique is able to relate the subject and object to a particular number i.e. 10 in the sentence and in turn able to understand a weirdness in sentence thus judging it humorous.

4.2.2 Different Set of Sentences;

Sentence 1 MOOOOOOOOONNNNNNNNEEEEEEYYYY!!!!!!

Sentence 2 why was six scared of seven because seven ate nine :)

Sentence 3 random sample

Sentence 4 Bazinga

Running the algorithm on these sentences again.

```
Enter the Sentence for detecting the humor :MOOOOOOOOONNNNNNNNEEEEEEYYYY!!!!!!
Sentence might be Humorous
Press 1 to continue :1
Enter the Sentence for detecting the humor :why was six scared of seven because seven ate nine :)
Sentence is Humorous
Press 1 to continue :1
Enter the Sentence for detecting the humor :random sample
Sentence Doesn't seem to have a humorous content
Press 1 to continue :1
Enter the Sentence for detecting the humor :Bazinga
Sentence is Humorous
Press 1 to continue :2
```

Figure 4.5 Results of Sentences 1-4

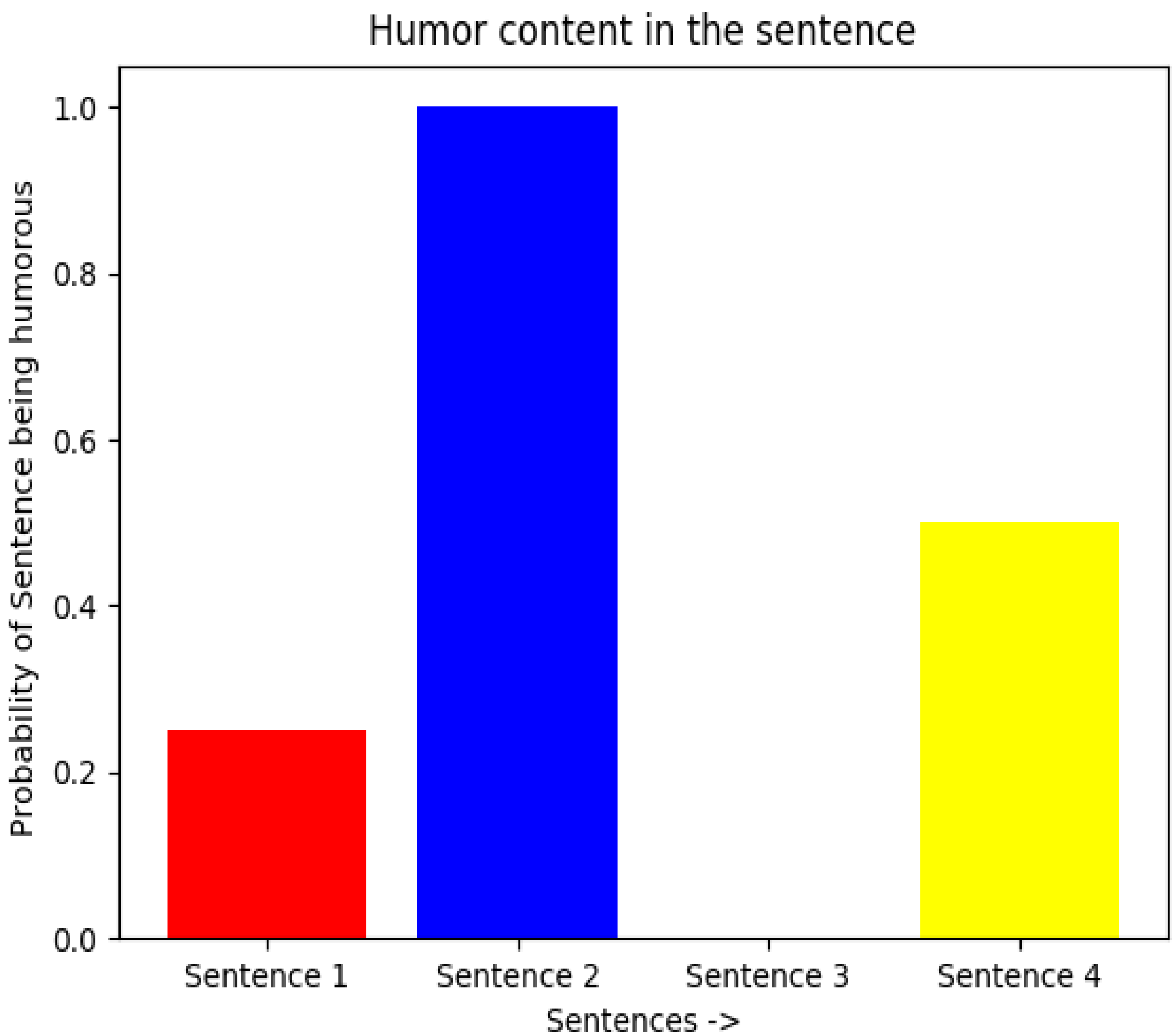

Figure 4.4 Bar Chart describing probability of sentence being humorous

Sentence 1 has a probability (0.25) of being humorous as it depends on the reference in which it is used.

Sentence 2 is humorous with a strong description of variations words and semantics. The probability of the sentence being humorous is 1.

Sentence 3 doesn't seem to have any sort of humor i.e. 0 probability. Since random sample cannot be humorous.

Sentence 4 is a common internet slang from famous TV series Big Bang Theory. So, it has a probability (0.5) of being humorous.

4.2.3 Comparison with Word Indexing power.

Word Indexing power will handle all the cases except Sentence 2 where the sentence sounds funny but word indexing power would not be able to judge because no funny words will be encountered in the sentence 2. But the proposed approach will work for sentence 2 as again it is able to compare subject and object with a number.

Word index power only look for words in the sentence it would not account for the sentence structure that is why it will lag behind the proposed technique which checks for semantics, sentence structure, funny words etc. to decide whether a sentence is humorous or not.

4.2.4 Different Set of Sentences

Sentence 1: Yesterday, I fell down from a 10 meter ladder. Thank God I was on the third step.

Sentence 2: Are you a man or a horse?

Sentence 3: Our conscience is clear- we don’t use it.

Sentence 4: It is the last example.

```
Enter the Sentence for detecting the humor :Yesterday, I fell down from a 10 meter ladder. Thank God I was on the third step.
Sentence is Humorous
Press 1 to continue :1
Enter the Sentence for detecting the humor :Are you a man or a horse?
Sentence is Humorous
Press 1 to continue :1
Enter the Sentence for detecting the humor :Our conscience is clear- we don't use it.
Sentence Doesn't seem to have a humorous content
Press 1 to continue :1
Enter the Sentence for detecting the humor :It is the last example.
Sentence Doesn't seem to have a humorous content
Press 1 to continue :2

Process finished with exit code 0
```

Figure 4.6 Sentence 1 to 4

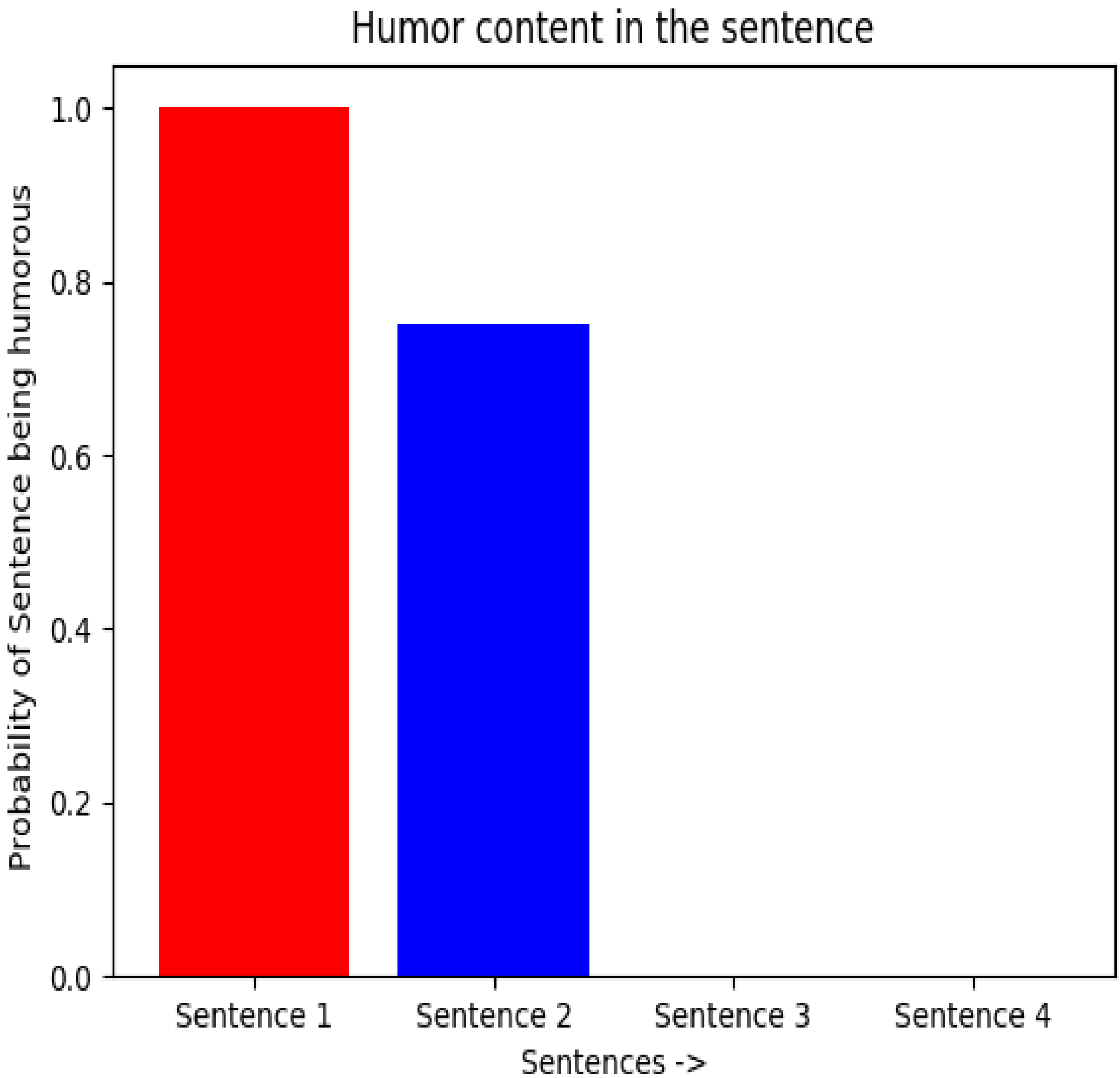

Figure 4.7 Bar Chart descripting probability of sentence being humorous

Sentence 1 has a very high probability of being humorous. It is also very funny to hear in linguistics terms. The technique is able to find this funny because of the organization of words at a particular place.

Sentence 2 is humorous but can be a bit offensive too. Because of the comparison with a horse. So, probability is 0.75 of it being humorous.

Sentence 3 doesn't seem to have any sort of humor. But can be humorous the technique is not able to find any sort of semantic connections so as to judge it humorous. So, the sentence having a clear conscious is transparent and cannot be humorous. So, technique failed here.

Sentence 4 doesn't seem to have any sort of humor.

4.2.5 Comparison with Word Indexing power.

The proposed technique has better results compared to word indexing power method.

Word Indexing power will not be able to handle Sentence 1 and Sentence 2 where the sentence sounds funny. For example, in sentence 2 only man and horse are there as subjects both are not funny words. So, word indexing power will yield incorrect results for these sentences. But proposed technique will be able to relate the subject and object of the sentence and see they both are compared to each other.

Word index power will work better in case of Sentence 3 compared to the proposed algorithm. Because proposed algorithm is not able to handle the organization of this sentence and fails to understand the semantics of the sentence.

4.2.6 Analysis of Proposed Technique and Word Indexing Power

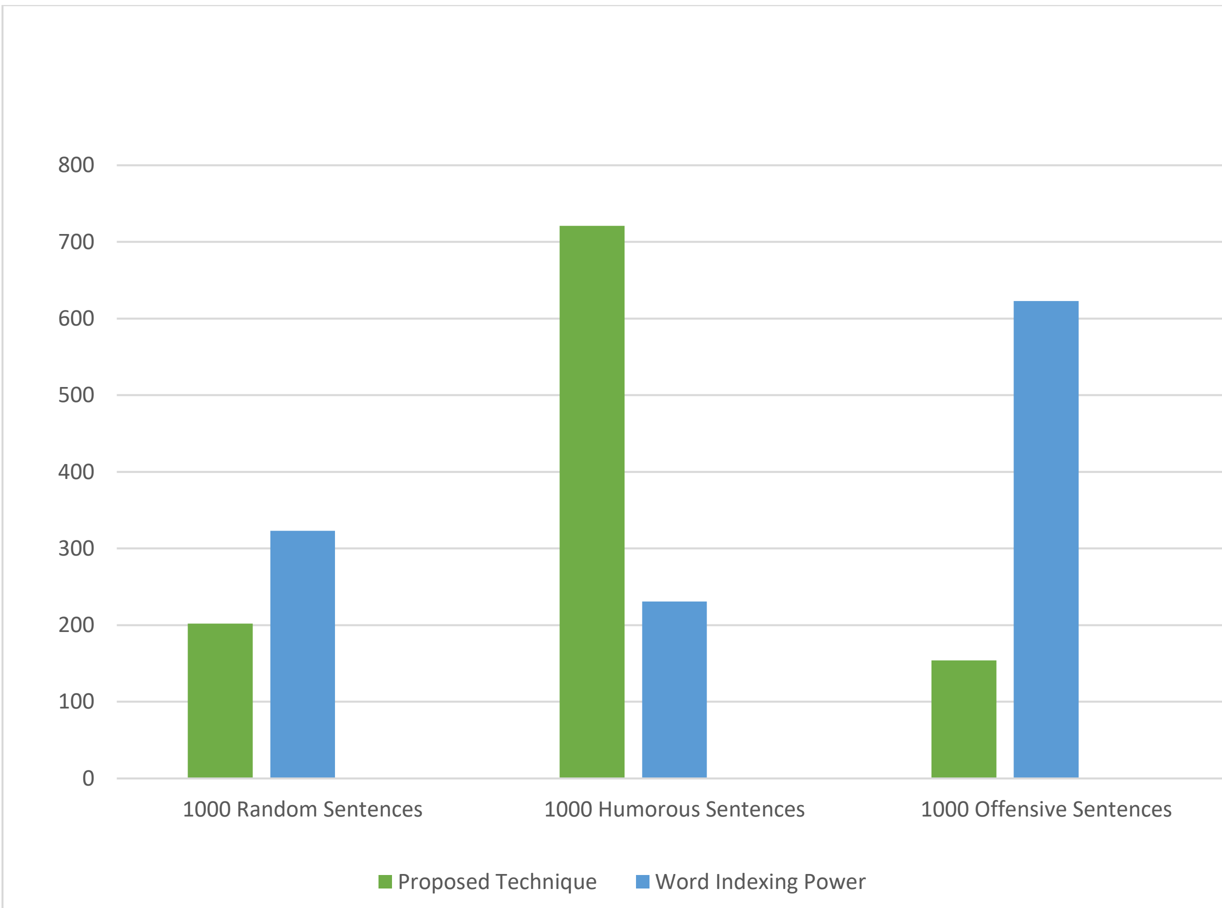

Figure 4.8 Bar Chart Displaying Results running on 1000 sentences

Figure 4.8 illustrates the comparison of proposed algorithm and word indexing power method.

1. First set contains 1000 random samples, out of which 128 are humorous sentences,

   The proposed technique judged 202 sentences as humorous while word indexing power judged 323 sentences as humorous.

2. In second set, 1000 humorous sentences were subjected to both methods.

   The proposed technique judged 700+ sentences humorous while word indexing power only judged 231 sentences as humorous (because of lack of semantic approach). A much better result is gained over word indexing power.

3. Finally, offensive sentences were subjected to both methods. word indexing power judged much more humorous sentences because it just simply looks for the funny side of a particular word in the sentence. Because a lot of offensive words have high humor factor.

As it is clear in the second point, for humorous sentences, the proposed method generally has better results compared to word indexing power because the latter only scans for the critical humorous words and compares it with a factor

But the proposed technique incorporates the method of word indexing by scanning critical humorous words and also trying to understand the semantics and meaning of the sentence and based on the concepts of linguistics.

For example, a humorous sentence - are you a man or a horse?

are -> 0.21 humor factor
you -> 0.18 humor factor
a -> 0.09 humor factor
man -> 0.32 humor factor
or-> 0.12 humor factor
horse> 0.41 humor factor

$$\boldsymbol{\eta = (0.21+0.18+0.09+0.32+0.12+0.09+0.41)/7}$$

$$\boldsymbol{\eta = 0.202}$$

η is significantly less than the factor which is above 0.4. So, the sentence is judged not humorous.

So that’s why a lot humorous sentences without any humorous words in them will fail to qualify as humorous sentences based on word index power method but proposed technique will be able to incorporate them.

4.2.7 Time Complexity Analysis of Proposed Technique and Word Indexing Power

Time Complexity of Word Indexing power is O(nm).

Where **n** is the number of words in the sentence and **m** is the size of the dataset of humorous words.

Time Complexity of Proposed algorithm is O(n^2m).

Where **n** is the number of words in the sentence and **m** is the size of the largest dataset which is antonym pair. The complexity can be reduced using segmentation if the dataset is sorted by 0(m*nlog n) but since we aqre considering the worst case we assume that the dataset is not alphabetically sorted.

Proposed algorithm will take more time by a factor of n which is the length of the sentence. And generally. in twitter one liners the size of sentence is small (less than 12 word a sentence). So, the time factor would not be that significant.

On the other hand, the accuracy of the proposed algorithm is much more than the word indexing power method as shown in section 4.2.6 part 2 by a factor of 70% on 1000 humorous sentences.

Chapter 5
Conclusion and Future works

5.1 Conclusion

Proposed algorithm tries to understand the meaning of a sentence using concepts of semantics, linguistics and syntax of the sentence. Also, the Proposed Algorithm uses a lot of datasets like antonym pairs, funny emojis, internet slangs etc. to work so, the results are highly dependent on the datasets used and therefore high quality of the datasets will lead to more accurate results.

Proposed algorithm can be compared to the word indexing power as shown in Chapter 4. The results generated by proposed algorithm for 1000 random sentences generate an accuracy of 47.2 % while that for word indexing power accuracy is 12.6%. So, the accuracy of proposed technique is way better than the word indexing power for random sentences.

Now when 1000 humorous sentences were subjected to both the techniques, proposed technique yielded an accuracy of 70.6% while word index power yielded an accuracy of 23.2%. Again, proposed technique yielded better results compared to word index power.

Now when 1000 offensive sentences were subjected to both of the techniques accuracy of Word index power is 42% and that of proposed technique is 37%. Slightly better than the proposed technique. But again by the basic definition of our humor, we neglected offensive language so as to restrain ourselves from ambiguous results. That is why proposed technique lagged marginally behind the word indexing power method.

From the conclusion drawn above it is quite clear that the proposed technique is a better alternative compared to word indexing power though it is slightly more complex in terms of run time complexity compared to word indexing power. But it yields far better results.

5.2 Future Works

Proposed technique still requires a lot of improvement. Because detection of humor is not a subjective thing. So, a lot of approaches can be used and inclusion and detection of Sarcastic tone can also be used because Sarcasm can lead to humor.

The time complexity of the algorithm can also be improved by segmenting the datasets first, using the sorted dataset or by sorting the datasets alphabetically. It can be reduced to O (n*log nm).

REFERENCES

[1] Attardo S. (1994). Linguistic theories of humor. Berlin: Walter de Gruyter.

[2] Attardo S., Hempelmann C. F., & Di Maio S. (2002). Script oppositions and logical mechanisms: Modeling incongruities and their resolutions. Humor: International Journal of Humor Research, 15, 3–46.

[3] Beale S., Lavoie B., McShane M., Nirenburg S., & Korelsky T. (2004). Question answering using ontological semantics Proceedings of the 2nd Workshop on Text Meaning and Interpretation (pp. 41–48). East Stroudsberg, PA: Association for Computational Linguistics.

[4] Bergson H. (1914). Laughter: An essay on the meaning of the comic. London: Macmillan.

[5] Brône G., Feyaerts K., & Veale T. (2006). Introduction: Cognitive linguistic approaches to humor. Humor-International Journal of Humor Research, 19(3), 203–228.

[6] Hempelmann C. F. (2004). Script opposition and logical mechanism in punning. Humor-International Journal of Humor Research, 17(4), 381–392

[7] Kao J. T., Levy R., & Goodman N. D. (2013). The funny thing about incongruity: A computational model of humor in puns Proceedings of the 35th Annual Conference of the Cognitive Science Society (pp. 728-733). Wheat Ridge, CO: Cognitive Science Society.

[8] Yoshua Bengio, Holger Schwenk, Jean-Sebastien ´ Senecal, Fr ´ ederic Morin, and Jean-Luc Gauvain. ´ 2006. Neural probabilistic language models. In Innovations

[9] Pawel Dybala, Michal Ptaszynski, Rafal Rzepka, Kenji Araki, and Kohichi Sayama. 2013. Metaphor, humor and emotion processing in human-computer interaction. International Journal of Computational Linguistics Research, 4(1):1–13.

[10] Chloe Kiddon and Yuriy Brun. 2011. That's what she said: double entendre identification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, pages 89–94. Association for Computational Linguistics.

[11] Humor recognition and humor anchor extraction. pages 2367–2376. Bianca Zadrozny. 2004.