GRAPH THEORETIC TEXT ANALYSIS

Mini Project - II

Submitted in Partial Fulfilment for the degree of
BACHELOR OF TECHNOLOGY
(Computer Science and Engineering)

By
ARIJIT SAHA (510517004)
ROHIT KUMAR (510517007)
AKASH SINGH (510517012)
SUBHAJYOTI SAHA (510517024)

Under the Guidance of
PROF. SUSANTA CHAKRABORTY

INDIAN INSTITUTE OF ENGINEERING SCIENCE AND
TECHNOLOGY, SHIBPUR
HOWRAH – 711103, WEST BENGAL
MAY, 2019

Signature of Students:

1.

2. Signature of Supervisor

3.

4.

Signature of Head of Department

OUTLINE

1. Introduction
2. Motivation
3. Objective
4. Preliminaries
5. Our Approach
6. Example
7. Application
8. Conclusion
9. Future Scope

1 INTRODUCTION

In recent years, the web has become an essential resource for obtaining information associated with any topic or domain.

The amount of text produced by interactions on social media, blogs, URLs, etc., has made essential to use advanced techniques to be able to understand and obtain valuable patterns from these large volumes of data, which is mainly text.

It is estimated that around 80% of all information is unstructured, with text being one of the most common types of unstructured data [1].

Because of the messy nature of text, analyzing, understanding, organizing, and sorting through text data is hard and time-consuming so most companies fail to extract value from that.

By using graph theoretic text analysis, we can structure business information such as email, legal documents, web pages, chat conversations, and social media messages in a fast and cost-effective way.

This allows us to save time when analyzing text data, help inform business decisions, and automate business processes.

1.1 MOTIVATION

The motivation of this analysis is to show how co-occurrence graphs can be used to represent text documents in a practical manner independently of the text analysis task [2].

To show this kind of representation can be a valuable asset to extract features/patterns, due to its structural simplicity.

Text Analysis can be used in real life for the following:

1. Finding representative keywords in text.
2. Text Summarisation
3. Finding keywords for tagging News Articles.
4. DNA Analysis
5. Text Classification

1.2 OBJECTIVE

Determination of Important nodes of a text co-occurrence graph generated using n-gram model.

Applying algorithm to literary text to detect words of significance to check credibility.

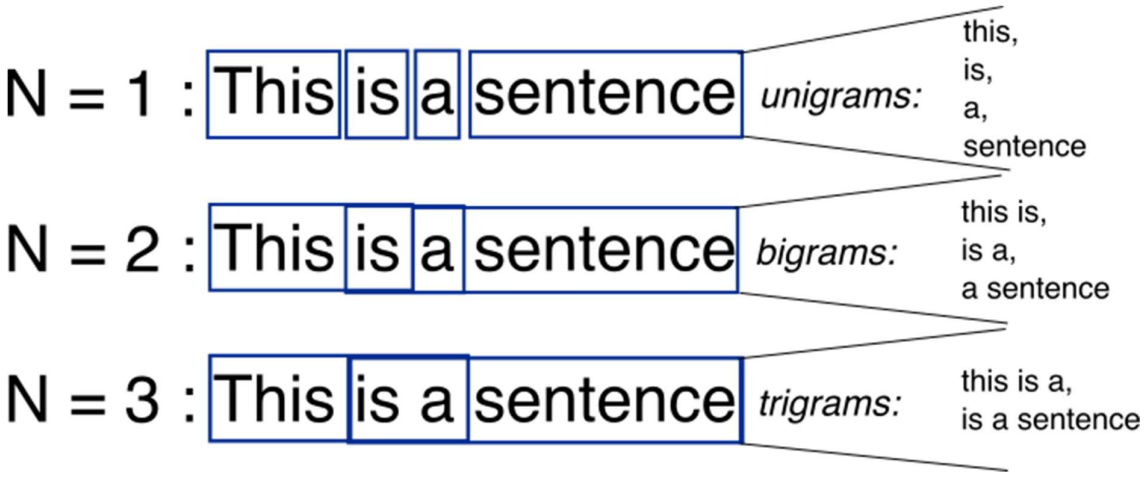
Applying algorithm to news articles to find keywords for tagging and sorting to check credibility.

2 PRELIMINARIES

2.1 N-GRAM

In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech [7]. The items can be syllables, letters, words or base pairs according to the application.

The n-grams typically are collected from a text or speech corpus. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram"; size 3 is a "trigram".



2.2 NLTK TEXT PROCESSING

We have used the Natural Language Tool Kit (NLTK) Text Processing Library available in python for pre-processing the text beforehand.

We have used the following features:

- *Wordnet*
- *Tokenization*
- *Stemming*
- *Lemmatization*

2.2.1 WORDNET

It is a Lexical Database for the English language, which was created by Princeton, and is part of the NLTK corpus [3][4]. It is used alongside the NLTK module to find the meanings of words, synonyms, antonyms, and more.

2.2.2 TOKENIZATION

Tokenization is the process by which big quantity of text is divided into smaller parts called tokens [3]. These tokens are very useful for finding patterns in the text so that text analysis can be performed on them.

Tokenization is also considered as is considered as a base step for stemming and lemmatization.

2.2.3 STEMMING

Stemming is the process of reducing inflected words to their word stem, base or root form [4]. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. We have used the Porter Stemmer Algorithm.

For Example, the words argue, *argued*, *argues*, *arguing*, and *argus* are reduced to the stem *argu.*

2.2.4 LEMMATIZATION

Lemmatization is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form [5].

Unlike stemming, lemmatisation attempts to select the correct lemma depending on the context.

The word *"better"* has *"good"* as its lemma. This link is missed by stemming.

We have primarily used this for Synonym Replacement to compress the vocabulary of the text.

3 OUR APPROACH

First, the text was tokenised in forms of sentences which in turn were tokenised further into words. Words were converted into lowercase letters. The word tokens were lemmatised, according to the Parts of Speech Tag. Stop-words (like *the*, *of*, *and*, *or*, etc) and punctuations were removed.

Words were replaced by their synonyms according to word frequency in that particular text by employing the synsets (Set of Synonyms) from NLTK Wordnet Library.

Tri-gram was generated from the text. An undirected graph was created on the basis of the Tri-gram where each Tri-gram was made into a fully connected component. Any edge added twice was merged into original edge with increased weight.

The centrality factors – Betweeness Centrality, Degree Centrality and clustering Coefficient were calculated for each node in the resultant graph.

A mean of the three factors was calculated for each node and the nodes were sorted according to this mean. The nodes with highest resultant mean value were taken as most influential and top 10% word-nodes were taken out of these to represent the text.

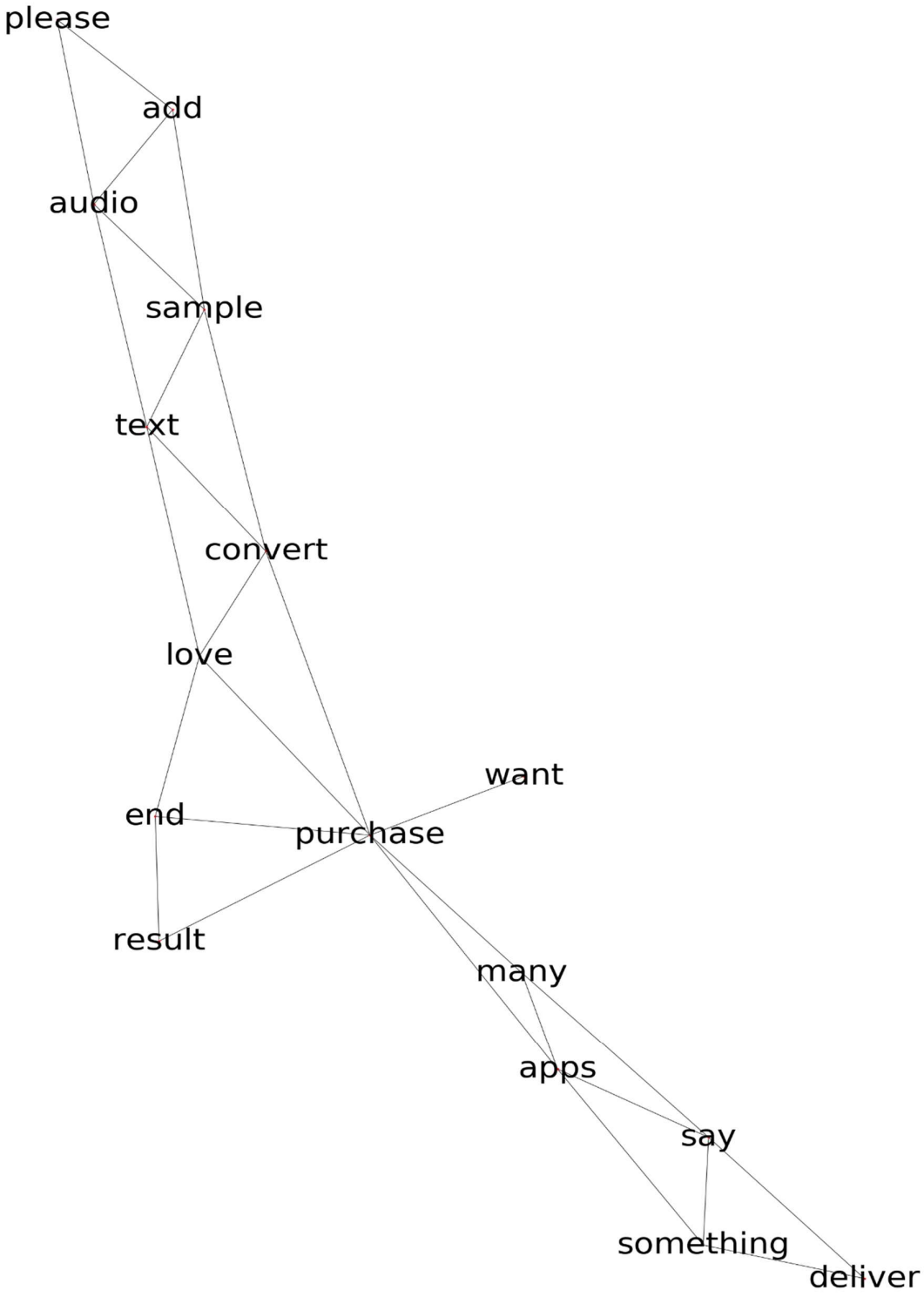
4 AN EXAMPLE

On Processing the text:

> *I would love to try or hear the sample audio your app can produce. I do not want to purchase, because I've purchased so many apps that say they do something and do not deliver. Can you please add audio samples with text you've converted? I'd love to purchase the end results.*

We get tokenised filtered text:

> *would love sample hear sample audio app produce*
> *want purchase purchase many apps say something deliver*
> *please add audio sample text convert*
> *love purchase end result*



Tri-Gram Text Graph Generated for above filtered Text

5 APPLICATION

5.1 IN LITERARY TEXTS

We took the Story "*Sonar Kella*" by *Satyajit Ray* and applied our algorithm.

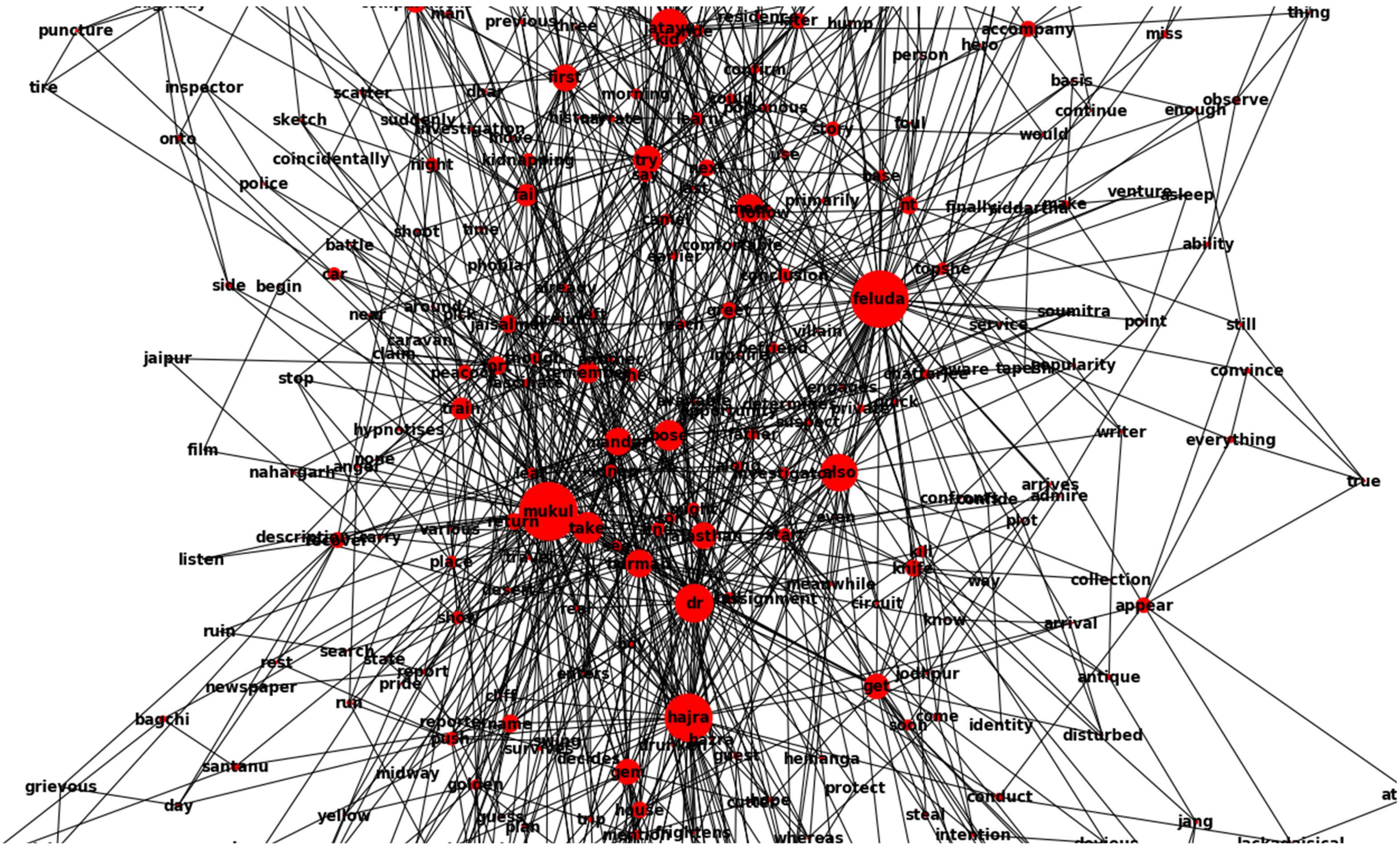
Top 10 word Nodes after voting:

Mukul	*feluda*
Hajra	*jatayu*
Dr	*bose*
Mandar	*rajasthan*
Also	*fort*

The words that are highlighted, are the keywords of the text. They are one of the following:

- *Protagonist/Antagonist (eg. Mukul, Feluda, Hajra, etc.)*
- *Geographical Location of Story Setting (eg. Rajasthan)*
- *Other keywords (eg. Fort)*

There are some discrepancies also as the words *"dr"* and *"also"* are also highlighted.


feluda
mukul
hajra
dr
also
bose
take
try
first
get
return
puncture
tire
inspector
onto
police
sketch
coincidentally
scatter
suddenly
investigation
night
kidnapping
morning
poisonous
story
foul
would
continue
basis
enough
observe
miss
thing
accompany
hero
person
hump
previous
three
man
residence
primarily
finally
venture
asleep
make
ability
point
still
convince
everything
true
writer
soumitra
service
popularity
villain
conclusion
comfortable
earlier
battle
car
side
begin
shoot
time
phobia
around
near
caravan
claim
jaipur
stop
film
hypnotises
none
nahargarh
anger
train
listen
various
place
ruin
rest
search
state
report
pride
newspaper
bagchi
santanu
grievous
day
yellow
midway
golden
guess
plan
trip
house
decides
drunken
hazra
guest
heltanga
protect
steal
intention
conduct
jang
identity
disturbed
antique
arrival
appear
collection
know
circuit
meanwhile
assignment
knife
way
plot
admire
arrives
engages
suspect
whereas
reporter
push
show
errors
even
at

5.2 IN NEWS ARTICLES

We applied our algorithm to news articles and got the results as shown in the next slide.

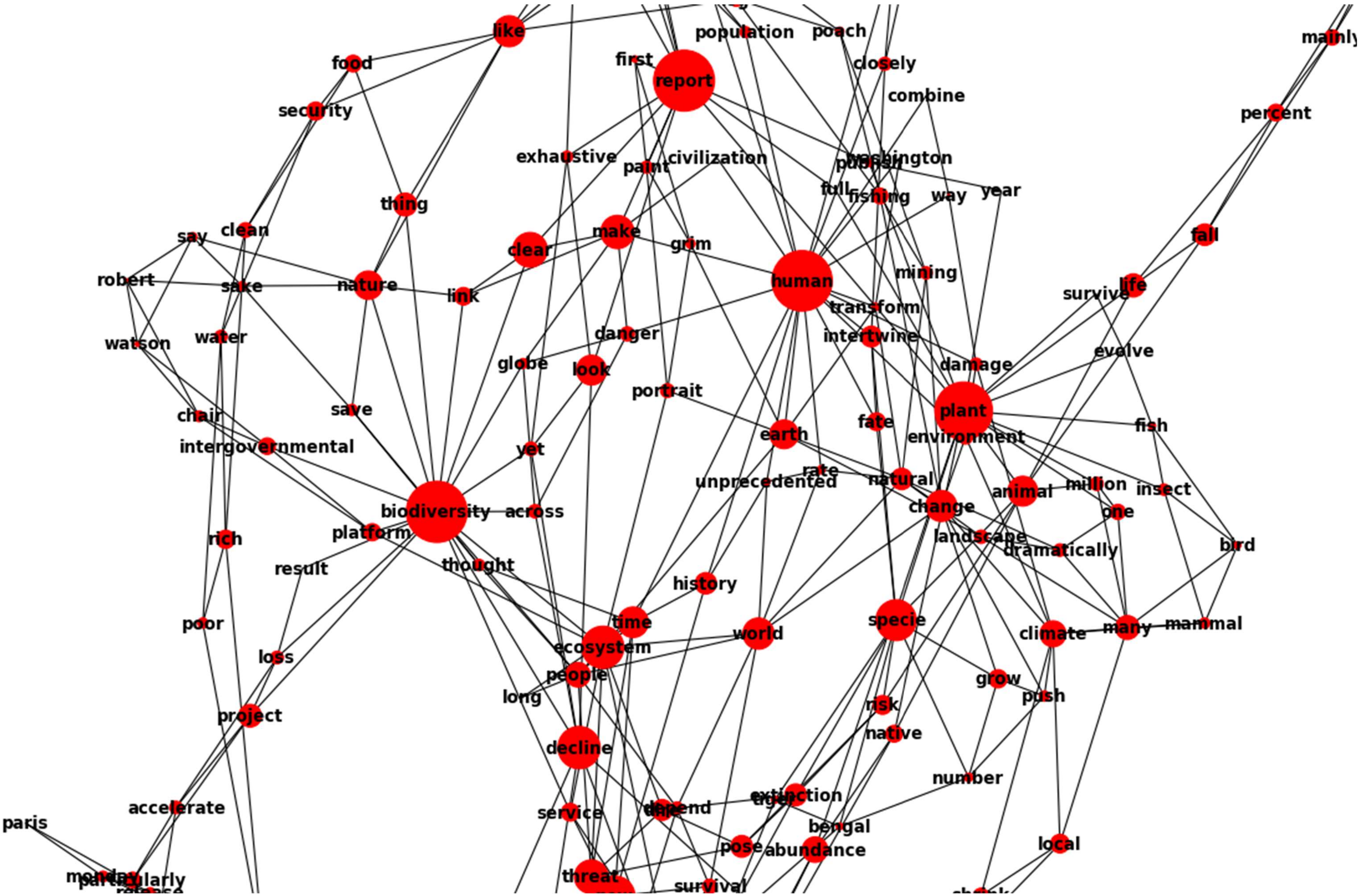
The example news article here is - *Humans Are Speeding Extinction and Altering the Natural World at an 'Unprecedented' Pace.*

The keywords generated are:

- *Human*
- *Biodiversity*
- *Plant*
- *Report*
- *Threat*
- *Decline*

The keywords this produces can be used to summarise the news articles into small keywords so that the news articles may be tagged for sorting and management.

The tags can be associated to news articles, so that searching softwares might use these tags to pull up news articles easily. Since, the news articles are smaller in size, the graphs tend to produce skewed results sometimes too.


like
population
poach
mainly
food
first
report
closely
security
combine
percent
exhaustive
paint
civilization
washington
thing
full
fishing
way
year
say
clean
make
grim
fall
clear
robert
sake
nature
link
human
mining
survive
life
transform
watson
water
danger
intertwine
evolve
globe
look
damage
portrait
save
plant
chair
fate
fish
earth
environment
intergovernmental
yet
rate
natural
million
unprecedented
animal
insect
biodiversity
across
change
one
rich
platform
landscape
dramatically
bird
result
thought
history
poor
time
specie
many
mammal
world
climate
ecosystem
loss
people
grow
long
push
project
risk
native
decline
number
extinction
accelerate
depend
bengal
paris
service
pose
abundance
local
particularly
threat
survival

6 RESULTS AND CONCLUSIONS

The results obtained show the relevance of graphs compared with traditional text analysis approaches that use the same dataset.

Co-occurrence graphs with a window of 3 words (Tri-Gram) works the best in comparison to lesser grams, which gives back very general keywords or higher grams, that tend to give more irrelevant keywords.

Betweenness Centrality gives the best results as a node with high betweenness acts central to the flow of meaning in a text. Degree Centrality gives the next best result, as a node with high degree represents more connectivity with other words in the graph.

Clustering Coefficient gives worst result as it is a measure of the surroundings rather than the node itself. If the text size is very less, then it may result in skewed results. The Synonym replacement process is not perfect yet, and it produces undesired replacements in some cases. Very rich literature or archaic literature poses challenges as it becomes very tedious to handle words that are not present in the NLTK library.

7 FUTURE SCOPE

Considering the growing amount of information in social media, the creation of proposed graphs will require tools capable of handling this kind of large datasets (Big Data analytics) [1].

Incorporation of Machine learning models to predict genre of text, and build more powerful classifiers of text.

8 REFERENCES

1. **Tanwar, Mona & Duggal, Reena & Khatri, Sunil Kumar. (2015)**. Unravelling unstructured data: A wealth of information in big data. 10.1109/ICRITO.2015.7359270.
2. **Esteban Castillo, Ofelia Cervantes, Darnes Vilarino (2017).** Computación y Sistemas, Vol. 21, No. 4, 2017, pp. 581–599 doi: 10.13053/CyS-21-4-2551
3. **George A. Miller (1995).** WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.
4. **Christiane Fellbaum (1998, ed.)** WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.
5. **Jursic, Matjaz & Mozetic, Igor & Erjavec, Tomaž & Lavrac, Nada. (2010).** LemmaGen:

Multilingual Lemmatisation with Induced Ripple-Down Rules. J. UCS. 16. 1190-1214. 10.3217/jucs-016-09-1190.

6. **Plumer, B. [2019]** Humans are speeding Extinction. The New York Times, available from: nytimes.com/2019/05/06/climate/biodiversity-extinction-united-nations
7. **Johannes, Fillemon. (1999).** A Study Using n-gram Features for Text Categorization.