**Nearest Document Search**

*Dissertation submitted to praxis business school*

*In Partial Fulfilment for the award of*

*Post Graduate Program in Data Science*

*By*

Abhisikta Biswas (D18001)

Madan P (D18015)

Ruhi Ghosh (D18034)

SatishVavilapalli (D18036)

Vinisha Maurya (D18042)

*Under the Guidance of*

Prof. Gourab Nath

Assistant Professor, Data Science

Praxis Business School, Bangalore



Month: March Year: 2019

**CERTIFICATE**

This is to certify that the Dissertation entitled “Nearest Document Search” is a bonafide record of independent research work done byAbhisikta Biswas (D18001), Madan P (D18015), Ruhi Ghosh (D18034), Satish Vavilapalli (D18036) and Vinisha Maurya (D18042) under my supervision and submitted to praxis business school, Bangalore in partial fulfilment for the award of the Degree of **POST GRADUATE PROGRAM** IN **DATA SCIENCE**.

Signature of the supervisor 1

Signature of the supervisor 2

**DECLARATION**

We Abhisikta Biswas, Madan, Ruhi Ghosh, Satish, and Vinisha are bonfire students of Post Graduate Program in Data Science in Praxis business School, Bangalore would like to declare that the dissertation entitled “Nearest Neighbour Document” submitted by us in partial fulfilment of the requirements for the award of the Degree of Post-Graduation Program in Data Science is my original work.

Place: Bangalore

Date: Signature of the candidates.

Abstract

As we, all know analysing text data is one of the major concerns for almost all. The main purpose of this project is to come up with the news recommendation engine. This application helps us to find the most nearest neighbour of one particular news article. With the use of natural language processor approach, we are able to provide with a summary of news along with its sentiment analysis plot. This project helps the user to save time from searching for the similar news here and there and saves time in reading the summary as well understanding the sentiment of the particular news article even before reading the entire news article. This application also helps to find the most frequently occurring words for any particular article. Thus applied KNN model for fetching three similar news of any particular category of news. In addition, various techniques of NLP (natural language processor) have been used to pre-process the text data before applying model and to find the sentiment of a news and summary of the news we have applied various NLP packages.

Acknowledgements

We have taken huge efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them. We are highly indebted to Prof. Gourab Nath for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project. We would like to express my gratitude towards member of Praxis business school for their kind co-operation and encouragement which helped us in completion of this project and PGP course. Our thanks and appreciations also go to all our colleagues in developing the project and people who have willingly helped us out with their abilities. Last but never the least we would like to thank our parents who have helped us to reach here.

Contents

[Chapter 1 8](#_Toc4187751)

[Introduction 8](#_Toc4187752)

[1.1. Objective 8](#_Toc4187753)

[1.2. Analytics problem 9](#_Toc4187754)

[1.3. Why is it important to solve the problem? 9](#_Toc4187755)

[Chapter 2 10](#_Toc4187756)

[2.1. Natural Language Processor 10](#_Toc4187757)

[2.1.1 Challenges in NLP 10](#_Toc4187758)

[2.2 Basics of NLP 11](#_Toc4187759)

[2.1.1. Bag of Words 11](#_Toc4187760)

[2.2.2 TF-IDF Vectorizer 12](#_Toc4187761)

[2.2.3 Tokenization 12](#_Toc4187762)

[2.2.4 Removal of Stop Words 13](#_Toc4187763)

[2.2.5 Stemming 13](#_Toc4187764)

[2.2.6 Lemmatization 14](#_Toc4187765)

[Chapter 3 16](#_Toc4187766)

[3.1 .Nearest Neighbour Search 16](#_Toc4187767)

[3.2 KNN (K nearest neighbour) 16](#_Toc4187768)

[3.2.1 Strength and Limitation of KNN 16](#_Toc4187769)

[3.3 Distance Metric 17](#_Toc4187770)

[3.4 KNN Algorithm: 19](#_Toc4187771)

[3.5 Pseudo Code for KNN 20](#_Toc4187772)

[3.6 Problem with KNN 20](#_Toc4187773)

[Chapter 4 23](#_Toc4187774)

[4.1 Word Cloud 23](#_Toc4187775)

[4.2 Text Summarization 24](#_Toc4187776)

[4.2.1 Objective of Text Summarization 24](#_Toc4187777)

[4.2.2 Overview 24](#_Toc4187778)

[4.2.3Stages 26](#_Toc4187779)

[4.2.4 EXTRACTIVE SUMMARIZATION METHODS 27](#_Toc4187780)

[4.2.4.1. Graph theoretic approach: LexRank 27](#_Toc4187781)

[4.3 Sentiment Analysis 29](#_Toc4187782)

[4.4 Code for Sentiment Analysis 32](#_Toc4187783)

[Chapter 5 34](#_Toc4187784)

[Conclusion 34](#_Toc4187785)

[Chapter 6 35](#_Toc4187786)

[Future Work 35](#_Toc4187787)

[Chapter 7 36](#_Toc4187788)

[References 36](#_Toc4187789)

# Chapter 1

# Introduction

Nearest search is a form of closeness search. It is used to find the nearest point in the dataset to the given point. The closeness is expressed in terms of a dissimilarity function that is the less similar the objects, the larger the function values. The metric used for calculating the dissimilarity is mainly a distance metric. The dissimilarity is measured using the Euclidean distance, cosine distance, Manhattan distance or other distance metric.

Application of nearest neighbour search:

* Nearest Document Search
* Pattern Recognition
* Spell Check
* Cluster Analysis
* Plagiarism Detection etc.

# Objective

News changes every day and thus news data is expanding tremendously. Searching a similar news article to one particular news from the newspaper or internet is yet again a tedious task. In addition, to understand the sentiment of a news, one needs to read the entire article of it. Moreover, in this busy life one does not have much time to read the entire news article, which is excessively long. The given business problem is to save the users time efficiently by providing the three similar news articles of any particular news and by providing the sentiment of the news even before reading the news by the user. It also provides a summary of the news, which then lets the user to decide whether the news interests him to read the entire news or not hence saving users time. It also provides with the most commonly occurring word in the news article.

## Analytics problem

According to the given business objective, the analytics problem can be summarised as follows**:**

* Fetching the three most nearest neighbour of one news article based of six different categories of news.
* Providing the summary of the news article of any category.
* Providing the sentiment of the news of any category.
* Providing the most frequently occurring word in the news article.

## Why is it important to solve the problem?

Time is money! So sometimes people avoid reading newspaper, as they do not have that much time in their day-to-day routine. Hence, this project is designed in order to keep people updated from the day-to-day news also by saving their precious time. This application will help the user to fetch three similar news to any news article from five different categories of news. This application will also provide a sentiment analysis check of any news. It also provides the summarization of the entire news in few lines and saving ample time of the users

# [Chapter 2](#_1t3h5sf)

## 2.1. Natural Language Processor

Computers are excellent in working with structured data like database tables. They can process the data much faster than humans can. However, working with unstructured data is difficult for computers. Data generated from conversation, tweets, news data and other social media are unstructured in nature. Unstructured data cannot be stored into traditional database like rows and columns. It is messy and hard to manipulate. To handle this type of unstructured data natural language processing was introduced in machine learning.

Natural Language Processor (NLP) is a field of Artificial Intelligence that gives the machine the ability to read, understand and derive meaning from human language. NLP is to use tools, techniques and algorithms to process and understand human readable language such as text, speech etc. NLP based applications are like Alexa, Siri etc.

Three parts of NLP are:

* Speech Recognition: The translation of spoken language into text
* Natural Language Understanding: The ability of computers to understand human language
* Natural language generation: the generation of natural language by computers

## 2.1.1 Challenges in NLP

Human language is complex. Understanding human language is very difficult because of its complexity. For example, there is an infinite number of different ways to arrange words in a sentence. Also, words can have several meanings and contextual information is necessary to correctly interpret sentences. Every Language is more or less unique and ambiguous. Just take a look at the following newspaper headline “The Pope’s baby steps on gays“. This sentence clearly has two very different interpretations, which is a pretty good example of the challenges in NLP.

### 2.2 Basics of NLP

Human language is complex. Understanding human language is very difficult because of its complexity. For example, there is an infinite number of different ways to arrange words in a sentence. Also, words can have several meanings and contextual information is necessary to correctly interpret sentences. Every Language is more or less unique and ambiguous. Just take a look at the following newspaper headline “The Pope’s baby steps on gays“. This sentence clearly has two very different interpretations, which is a pretty good example of the challenges in NLP.

## 2.1.1. Bag of Words

To represent text data to machine learning algorithm we need a way. Bag of words is the most common model that counts all the words in a particular text corpus. Basically it creates an occurrence matrix for the sentence or document, disregarding grammar and word order.  It is a way of extracting features from the text for use in machine learning algorithms. In this approach, we use the tokenized words for each observation and find out the frequency of each token.

Example:” Today is a shiny day”

“Yesterday was Sunday”

Here each sentence is tokenized into words. Like “Today”, “is”, “shiny”, “day”, “Yesterday

“, “was”, “Sunday”.

Conting the words and creating the vector.:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Today | Is | A | shiny | day | Yesterday | was | Sunday |
| Today is a shiny day | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Yesterday was Sunday | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

This process of converting NLP text into numbers is called vectorization in machine learning.

## 2.2.2 TF-IDF Vectorizer

TF-IDF stands for term frequency-inverse document frequency. TF-IDF weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

* Term Frequency (TF)

is a scoring of the frequency of the word in the current document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. The term frequency is often divided by the document length to normalize.

TF (t) =

* Inverse Document Frequency

It is a scoring of how rare the word is across documents. IDF is a measure of how rare a term is. Rarer the term, more is the IDF score.

IDF (t) =

Hence, TF-IDFscore=TF\*IDF

## 2.2.3 Tokenization

Converting the sentecne into words is called as tokenization. That is cutting down each sentences into pieces called as tokens. Tokenization also helps in removing the punctuation. Tokenization is a basic need while dealing text data in NLP.

Example: “NLP is awesome”

The tokens of this sentence will be “NLP”, “is”, “awesome”.

Thus it is a process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.

## 2.2.4 Removal of Stop Words

In common day to day language we use a lot of words like: the, is, at. These are mostly pronouns, prepositions in English language. These are called as stop words. These words add no value to the sentences. Hence in NLP it is always advisable to remove the stop words from the text. These stop words can be saftly removed by carrying out a lookup in a pre-defined list of keywords, freeing up database space and improving processing time.

There is no universal list of stop words. These can be pre-selected or built from scratch.Or we can use the stop word library from nltk package.

But the stop word removal can sometimes also lead to removing the relavant information and then modify the context.  For example, while performing a sentiment analysis we might throw our algorithm off track if we remove a stop word like “not”. Under these conditions, we might select a minimal stop word list and add additional terms depending on your specific objective.

## 2.2.5 Stemming

Stemming is a process of removing the end or beginning of the word with the intention of removing suffix or prefix. It is a heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time.  Stemming is important in natural language processing. stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate” This indiscriminate cutting can be successful in some occasions, but not always, and that is why we affirm that this approach presents some limitations.

Some Stemming algorithm are:

* **Potter’s Stemmer algorithm**

It is one of the most popular stemming methods proposed in 1980. It is based on the idea that the suffixes in the English language are made up of a combination of smaller and simpler suffixes.

Advantage**:** It produces the best output as compared to other stemmers and it has less error rate.

Limitation**:**  Morphological variants produced are not always real words.

* **Lovin’s Stemmer**

It is proposed by Lovins in 1968, that removes the longest suffix from a word then word is recoded to convert this stem into valid words.

Advantage**:** It is fast and handles irregular plurals like 'teeth' and 'tooth' etc.

Limitation**:** It is time consuming and frequently fails to form words from stem.

* **Dawson Stemmer**

It is extension of Lovins stemmer in which suffixes are stored in the reversed order indexed by their length and last letter.

Advantage**:** It is fast in execution and covers more suffices.

Limitation**:** It is very complex to implement.

* **Krovetz Stemmer**

It was proposed in 1993 by Robber Krovetz. Following are the steps:

1. Convert the plural form of the word to singular form
2. Convert the past tense of a word to its present tense and remove the suffix ‘ing’.

Advantage**:** It is light in nature and can be used as pre-stemmer for other stemmers.

Limitation**:** It is inefficient in case of large documents.

* **Xerox Stemmer**

Advantage**:** It works well in case of large documents and stems produced are valid.

Limitation**:** It is language dependent and mainly implemented on english and over stemming may occur.

* **N-Gram Stemmer**

An n-gram is a set of n consecutive characters extracted from a word in which similar words will have a high proportion of n-grams in common.

Advantage: It is based on string comparisons and it is language dependent.

Limitation: It requires space to create and index the n-grams and it is not time efficient.

## 2.2.6 Lemmatization

Lemmatization helps to reduce the words to its base form and group them together different form of the same word. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.The key to this methodology is linguistics in lemmatization. At time Lemmatization is preferred over Stemming because lemmatization does morphological analysis of the words.

Example: running”, “runs” and “ran”are all forms of the word “run”. So the lemma of these words will be run.

**Difference between stemming and Lemmatization:**

The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

However, the two words differ in their flavor. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.

Example:

|  |  |  |
| --- | --- | --- |
| **Word** | **Stemming** | **Lemmatization** |
| Caring | Car | Care |

In this project, we have used almost all the procedures of NLP in our news data. Each news is initially separate based on their categories. Then we have performed the cleaning on the news data like:

* Removing the punctuations
* Removing the digits
* Converting every text to lower case
* Removing the stop words from the text
* Performing lematization on the text
* Converting to TF-IDF vector

After cleaning the news data we have applied the KNN model to search for the nearest news based on a perticular news.

# Chapter 3

## .Nearest Neighbour Search

This is for retrieving documents of interest. Searching for an article, which is similar to the article of your choice. To search for the nearest neighbour we use algorithms like KNN(K nearest neighbour), KD-Tree, Locality Sensitive Hashing. Distance metrics used for the nearest neighbour. The distance metric used to calculate nearest document is cosine similarity metric.

## KNN (K nearest neighbour)

KNN is a supervised algorithm, which is mainly used in machine learning also in data mining.

It is a non-parametric approach. It is a classification algorithm and the learning is based on how similar the data is from other. KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970’s. It is much popular as it is easy to use and results are clearly understandable. The k in KNN is refers to the number of nearest neighbour used to predict or classify from the entire dataset.

**When do we use kNN algorithm?**

* KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry.
* To evaluate any technique one should generally look at three important aspects 1. Ease to interpret output 2. Calculation time 3. Predictive Power

## Strength and Limitation of KNN

Advantage:

* Robust in nature
* Effective in large dataset
* Can be applied to data of any distribution
* Simple and intuitive

Disadvantage:

* Need to determine the value of k is tricky
* Choosing the correct distance metric is sometimes a challenge
* Computational cost is high
* Performance of the algorithm depends on the number of dimensions used

## 3.3 Distance Metric

Distance metrics are a method to find distance between a new data point and existing training dataset.

* Euclidian Distance 2

Euclidian distance is the most common type of distance metric. Mostly when the data type is dense or continuous then it is the best proximity measure. It finds the distance between two data point.

* Manhattan Distance

Manhattan distance is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates. It is the total sum of the difference between x-coordinate and y-coordinate.

* Minkowski Distance q)⅟q

Minkowski distance is a generalized metric form of Euclidian distance and Manhattan distance.

For q≥1, the minkowski distance metric is result of minkowski inequality.

Synonyms of minkowski:

* When q=1, it is Manhattan distance. It can also be called as city block distance
* When q=2, it is Euclidian distance.
* When q=∞, it is the chebyshev distance
* Cosine Similarity Distance

In this project cosine similarity distance metric has been used to calculate the nearest neighbour. Cosine similarity metric is a normalized dot product of two data. It measures the similarity between two non-zero vectors of an inner product space that measures the cosine angle between them. Thus if two vectors with same orientation have a cosine similarity of 0 degree then they are highly similar and cosine similarity is one. If the angle between two vectors is 90 degree then the two data points are highly dissimilar and have a similarity of zero. If two vector are diametrically opposed, have a similarity of -1.

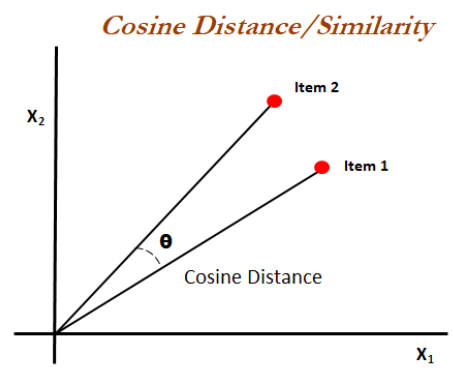


Figure 1: Cosine similarity

Cosine similarity is particularly used in positive space, where the outcome is nearly bounded in [0,1]. The reason why cosine similarity is popular because it is very efficient to evaluate, mainly for sparse vector

## 

## 3.4 KNN Algorithm:

**Define K**

**Compute the distance between one of the data point to the other data point from the same dataset**

**Sort the distance**

Figure 2: KNN Algorithm

**Apply simple majority**

**Take k nearest neighbours**

## 3.5 Pseudo Code for KNN

1. Load the data

2. Initialize value of k

3. **for** all the unknown samples UnSample (i):

**for** all the known samples Sample (j):

Compute the distance between UnSample (i) and Sample (j)

**End**

Sort the calculated distances in ascending order based on distance values

Find the k smallest distances locate the corresponding samples Sample (j1),..., Sample (jk)

Assign UnSample (i) to the class which appears more frequently

**End**

## 3.6 Problem with KNN

KNN is a supervised algorithm used for classification and regression problems for non-parametric methods. Non parametric methods do not assume a parametric form of f(x). When the value of k is small it provides more flexible fit which has low bias and high variance. Here the variance is high because it is entirely dependent on one observation. A large value of k will provide more smooth and less variable fit.

A parametric approach will outperform the non-parametric approach if the parametric form that has been selected is closed to the true form of f. It is observed that KNN performs slightly worse than linear regression when the actual relationship is linear, but much better than linear regression for nonlinear situation. Thus a conclusion might be drawn from this that KNN should be favored over linear regression. But in reality even when the true relationship is not linear, KNN might still provide inferior results compared to linear regression.

* Considering a highly non-linear situation, when number of predictors are less like 1 or 2 then KNN performs better when compared to linear regression but as the number of predictor increases, linear regression outperforms KNN. Thus the decrease in performance as dimension increases is a common problem of KNN.
* Scaling issues: Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes.

Examples: Height of a person may vary from 4’ to 6’

Example: In a data set which has 100 observations and one predictor, this provides enough information to estimate f(x). Now if distributing this 100 observations over 2 predictors this will result in a situation where a given observation has no nearby neighbor. This means the kth observation that are nearest to a given test observation x, may be very far away from x in p dimensional space where p is a large value. Thus it leads to a poor fit of KNN. Hence this is also coined as “curse of dimensionality”.

Also one of the major problem with KNN is its interpretability. Unlike other supervised algorithms KNN is hard to interpret.

2. KNN algorithm is a lazy learner:

K-NN is a lazy learner because it doesn’t learn a discriminative function from the training data but “memorizes” the training dataset instead.

For example, the logistic regression algorithm learns its model weights (parameters) during training time. In contrast, there is no training time in K-NN. Although this may sound very convenient, this property doesn’t come without a cost: The “prediction” step in K-NN is relatively expensive! Each time we want to make a prediction, K-NN is searching for the nearest neighbor(s) in the entire training set.

3.KNN can have poor run-time performance when the training set is large. It is very sensitive to irrelevant or redundant features because all features contribute to the similarity and thus to the classification.

4. Distance based learning is not clear which type of distance to use and which attribute to use to produce the best results.

5. Computation cost is quite high because we need to compute distance of each query instance to all training samples.

Word Cloud are a way of displaying how important words are in a corpus or text. It identifies the word which frequently arrives in the text by giving that particular word a greater space to occupy in the image. It help us get an intuition about what the collection of texts is about.

In this project we have used KNN model using cosine similarity distane on the cleaned text data of news (cleaning of the data is done using natural language processor in python using different packages like nltk, stopwords, etc this is being explained in detail in the next chapter) and obtained the three nearest neighbouring news article for any perticular article for each different category of news.

KNN Model:



# Chapter 4

# Word Cloud

Word Cloud are a way of displaying how important words are in a corpus or text. It identifies the word which frequently arrives in the text by giving that particular word a greater space to occupy in the image. It help us get an intuition about what the collection of texts is about.

**Uses of word cloud:**

* Helps to take a quick look at the word distribution of a collection of texts
* Clean the texts and want to see what are some frequent stopwords you want to filter out
* See the differences between frequent words between two or more collections of texts

Python has a inbuilt library wordcloud which helps to generate the word cloud easily.



Figure 4.1 Word Cloud

In this project wordcloud package has been installed and for every category of the news wordcloud has been generated. Thus every news generates its own wordcloud and giving its most occuring words a higher space in the image.

# Text Summarization

## Objective of Text Summarization

­­­­­ In this new era, where tremendous information is available on the Internet, it is most important to provide the improved mechanism to extract the information quickly and most efficiently. It is very difficult for human beings to manually extract the summary of a large documents of text. There are plenty of text material available on the Internet. So there is a problem of searching for relevant documents from the number of documents available, and absorbing relevant information from it.In order to solve the above two problems, the automatic text summarization is very much necessary.

With push notifications and article digests gaining more and more grip, the task of generating intelligent and accurate summaries for long pieces of text has become a popular research as well as industry problem.

## 4.2.2 Overview

There are two fundamental approaches to automatic text summarization: **Extractive** and **Abstractive**. The former extracts words and word phrases from the original text to create a summary. The later learns an internal language representation to generate more human-like summaries, paraphrasing the intent of the original text.

The problem of extracting a sentence that represents the contents of a given document or a collection of documents is known as extractive summarization problem.

Difference between these two approaches as an example follows:

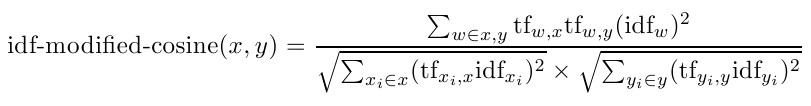
**Extractive summaries**

An increase in police activities ISIS has been operating for years; Now its action in Iraq are promoting the US to target its fighter with airstrikes there are to threaten more such strikes in Syria.

**Abstractive summaries**

Airstrikes says it is one of the U.S. military programme.

In this project we are following ‘extractive summaries’ approach. Every sentence represents one node, and the edges are similarity relationship between sentences in the corpus. They measure similarity between sentences by considering every sentence as bag-of-words model. This means that the similarity measure between sentences is computed by frequency of word occurrence in a sentence. The basic measurement is using TF-IDF formulation, where tem frequency (TF) contributes to the similarity strength as the number of word occurrences is higher. On the other hand, the inverse document frequency regards low frequency words inversely contributes to higher value to the measurement. This TF-IDF formulation is then used as  a measurement for similarity between sentences by using it in this idf-modified-cosine formula

 --- Idf-modified-cosine equation

This formula is measuring the ‘distance’ between two sentences x and y. the more similar two sentences, the more ‘closer’ they are to each other.

This similarity measure is then used to build a similarity matrix, which can be used as a similarity graph between sentences. The LexRank

Algorithm measure the importance of sentences in the graph by considering its relative importance to its neighbouring sentences, where a

Positive contribution will raise the importance of a sentence’s neighbour, while a negative contribution will lower the importance value of a

Sentence’s neighbour.

Summarization application

* Outline or abstracts of any documents, or news article
* Summarise of email threads
* Action items of a meeting
* Simplifying text by compressing sentences

## 4.2.3Stages

Content selection

Information ordering Sentence Realisation

Removing redundancy

Figure 4.2.3 Stages

## 4.2.4 EXTRACTIVE SUMMARIZATION METHODS

A. Term Frequency-Inverse Document Frequency (TF-IDF) method:

B. Cluster based method

C. Graph theoretic approach

D. Machine Learning approach

E. Text summarization with neural networks

F. Automatic text summarization based on fuzzy logic

## 4.2.4.1. Graph theoretic approach: LexRank

In this technique, there is a node for every sentence. Two sentences are connected with an edge if the two sentences share some common words, in other words, their similarity is above some threshold. This representation gives two results :The partitions contained in the graph (that is those sub-graphs that are unconnected to the other sub graphs), form distinct topics covered in the documents. The second result by the graph-theoretic method is the identification of the important sentences in the document. The nodes with high cardinality (number of edges connected to that node), are the important sentences in the partition, and hence carry higher preference to be included in the summary.

S1 -> {(computation, 0.1), (process, .15)…}

S2 -> {(computation, 0.1), (process, 0.05)}

S3 -> ……………….

Text Document

Computation is a process following a well-defined model……A computation can be seen as a purely physical phenomena………

Machine readable format

Document representation

Underlying Hypothesis

Sentences that convey the theme of the Document are more similar toeach other

S1 S2

Si S3

Document

Finding the most salient sentences

Figure 4.2.4.1 LexRank Approach

**Problem**

Abstractive method is similar to summaries made by humans. Abstractive summarization as of now requires heavy machinery for language generation and is difficult to replicate into the domain specific area.

## Sentiment Analysis

Sentiment analysis is a type of data mining that measures the inclination of people’s opinions through natural language processing (NLP), computational linguistics and text analysis, which are used to extract and analyze subjective information from the Web - mostly social media and similar sources. The analyzed data quantifies the general public's sentiments or reactions toward certain products, people or ideas and reveal the contextual polarity of the information.

Sentiment analysis is also known as opinion mining.

The aim of sentiment analysis is to get the subjective opinion of a document or collection of documents, like blog posts, reviews, news articles and social media feeds like tweets and status updates.

Sentiment analysis is used to track the following:

* Brand reception and popularity
* New product perception and anticipation
* Company reputation
* Flame/rant detection

For our project we have useader VADER (Valence Aware Dictionary for sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. VADER text sentiment analysis uses a human-centric approach, combining qualitative analysis and empirical validation by using human raters and the wisdom of the crowd.

4.3.1 What is VADER?

VADER (Valence Aware Dictionary for sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. VADER text sentiment analysis uses a human-centric approach, combining qualitative analysis and empirical validation by using human raters and the wisdom of the crowd.

4.3.2 How does VADER works?

VADER sentiment analysis returns a sentiment score in the range -1 to 1, from most negative to most positive.

The sentiment score of a sentence is calculated by summing up the sentiment scores of each VADER-dictionary-listed word in the sentence. Cautious readers would probably notice that there is a contradiction: individual words have a sentiment score between -4 to 4, but the returned sentiment score of a sentence is between -1 to 1.

They’re both true. The sentiment score of a sentence is the sum of the sentiment score of each sentiment-bearing word. However, we apply a normalization to the total to map it to a value between -1 to 1.

The normalization used by Hutto is

\large \dfrac{x}{\sqrt{x^2 + \alpha}}

where x is the sum of the sentiment scores of the constituent words of the sentence and \alpha is a normalization parameter that we set to 15. The normalization is graphed below.

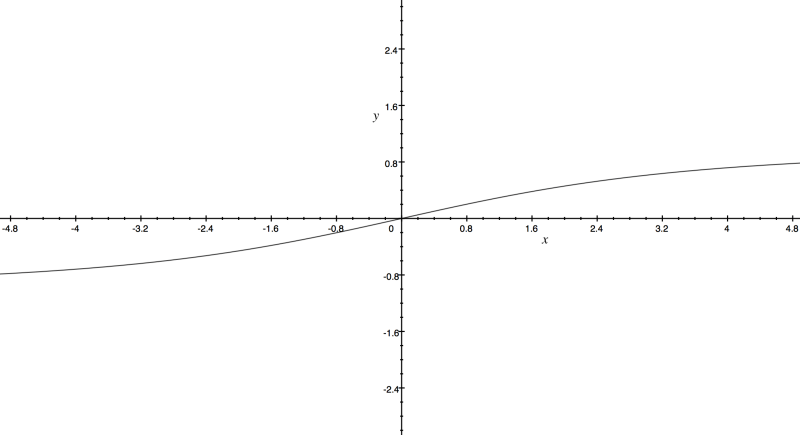


Figure 4.3 Normalization Graph

We see here that as x grows larger, it gets more and more close to -1 or 1. To similar effect, if there are a lot of words in the document you’re applying VADER sentiment analysis to, you get a score close to -1 or 1. Thus, VADER sentiment analysis works best on short documents, like tweets and sentences, not on large documents.

erm Frequency-Inverse Document Frequency (TF-IDF) method:

B. Cluster based method:

C. Graph theoretic approach:

D. Machine Learning approach:

E. Text summarization with neural networks :

F. Automatic text summarization based on fuzzy logi

A. Term Frequency-Inverse Document Frequency (TF-IDF) method:

B. Cluster based method:

C. Graph theoretic approach:

D. Machine Learning approach:

E. Text summarization with neural networks :

F. Automatic text summarization based on fuzzy logi

A. Term Frequency-Inverse Document Frequency (TF-IDF) method:

B. Cluster based method:

C. Graph theoretic approach:

D. Machine Learning approach:

E. Text summarization with neural networks :

F. Automatic text summarization based on fuzzy logi

Term Frequency-Inverse Document Frequency (TF-IDF) method:

B. Cluster based method:

C. Graph theoretic approach:

D. Machine Learning approach:

E. Text summarization with neural networks :

F. Automatic text summarization based on fuzzy logic

Term Frequency-Inverse Document Frequency (TF-IDF) method:

B. Cluster based method:

C. Graph theoretic approach:

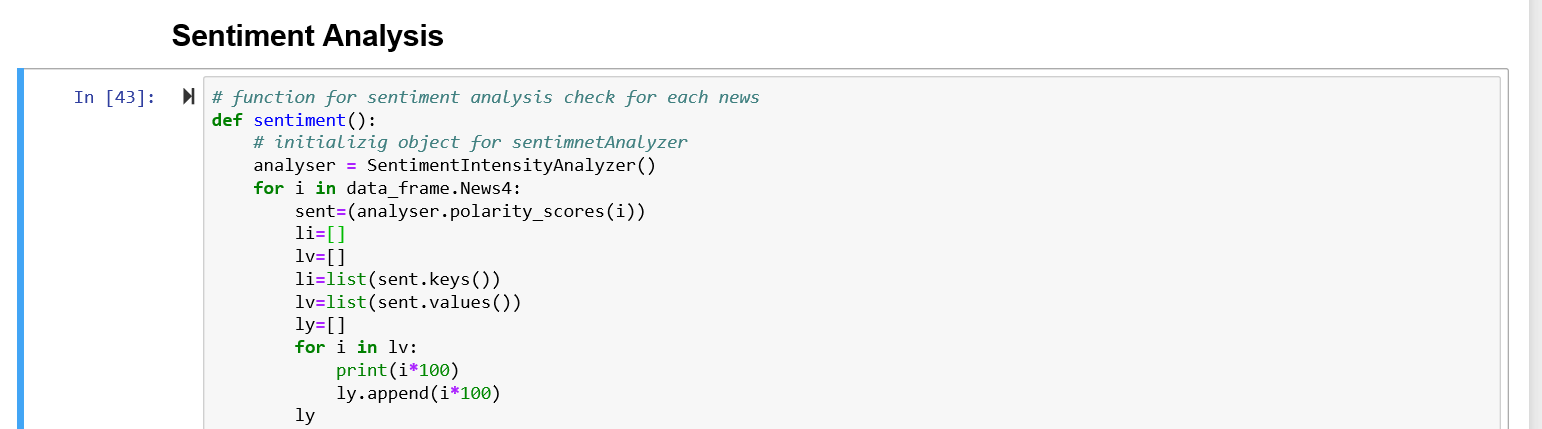
D. Machine Learning approach:

E. Text summarization with neural networks :

F. Automatic text summarization based on fuzzy lo

# Code for Sentiment Analysis

We have created function sentiment() that can perform sentiment analysis on each news and gives a doughnut plot of how pleasant, depressing and neutral content in the news.



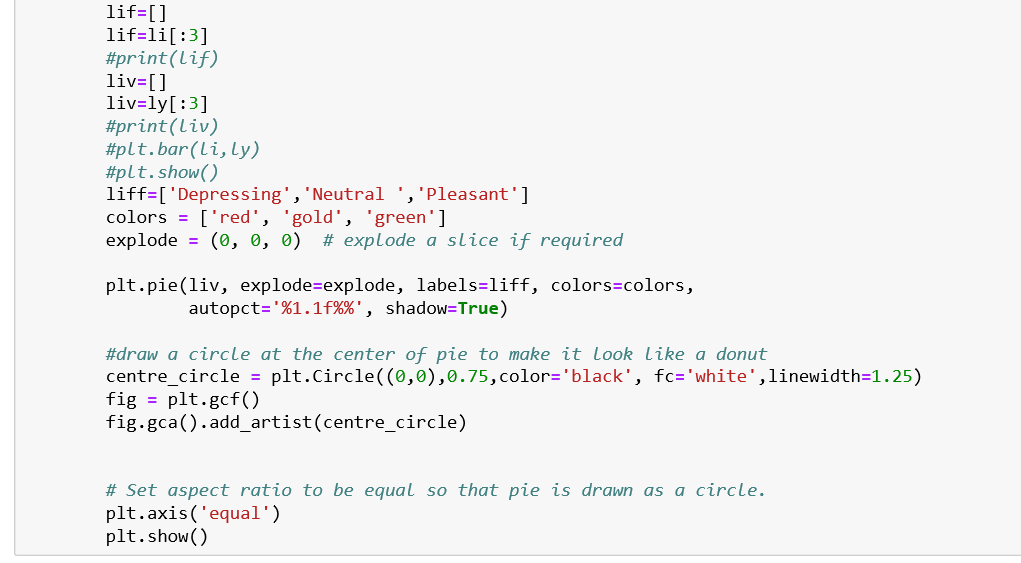


Figure 4.4 Code screenshot for sentiment analysis

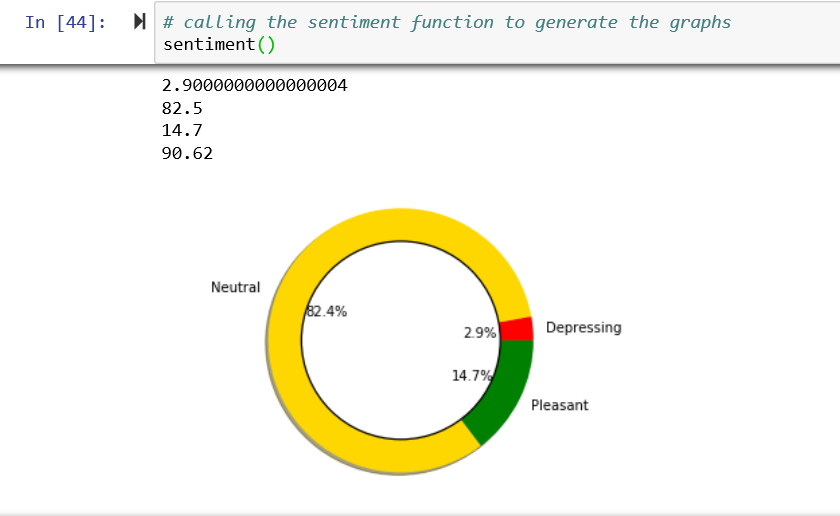


Figure 4.4 Sample doughnut plot of a news

# Chapter 5

## Conclusion

Thus, in this project we see blend of technique’s of advanced analytics like natural language processor along with traditional machine learning model like KNN. How a new recommendation engine works and how it fetches out the nearerst neighbouring news of any particular news aticle from the five different categories of new. The application also generates out the sentiment of any news. It also genereates a summary of any news article which gives a brief about the entire news article. It also fetches out the frequently occuring words for any news article.

We have used tkinter to build the user interface for the application in python. Also, we have used various other packages like nltk, matplotlib, knn, pandas, stopwords, lemmatizer, wordcloud, vadersentiment, lexranksummarizer etc to build the application in python.

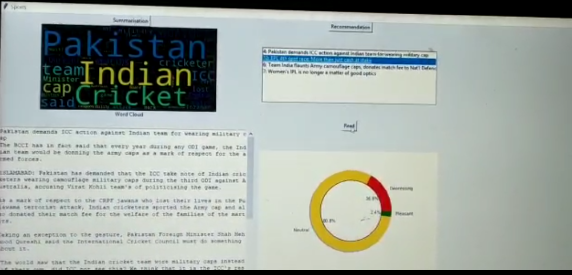


Fig 5.1 Screenshot from the application

# Chapter 6

## Future Work

This chapter deal with the work which we feel that we missed out because of the time contraints, which is cost sensitive learning on the data, which we are keeping it in the future work.

* Diffrentiating more categories for news
* Building for other relevant text data like Book recommendation engine
* Applying few more techniques of natural language processor

Also, other traditional modelling techniques like LDA could be performed on similar kind of the data set. And for its nearest neighbour for any particular news.

Currently we have created documents for our convenience, so we have chosen the small size of documents. But in real time documents may vary in size. There we can face challenges while using Vader sentiment analysis.

Most of the news websites have not allowed us to scrape or copy documents.Main drawbacks were availability of documents.

As the choice of the documents is as per our interest, so we have created small dataset for this project. KNN works well when the dataset is small. But in real time, when the dataset is large enough, KNN seems to fail, and we may not get optimum results.

There we can use different methods like kdtree, k-means clustering etc.

# Chapter 7

## References

http://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/

https://en.wikipedia.org/wiki/Cosine\_similarity

https://datascience.stackexchange.com/questions/20075/when-would-one-use-manhattan-distance-as-opposite-to-euclidean-distance

https://en.wikipedia.org/wiki/Minkowski\_distance

https://towardsdatascience.com/tagged/nlp

https://stackoverflow.com/questions/1787110/what-is-the-true-difference-between-lemmatization-vs-stemming

https://en.wikipedia.org/wiki/Lemmatisation

https://discuss.analyticsvidhya.com/t/use-of-different-distance-measures-in-knn-algorithm/2601

https://towardsdatascience.com/introduction-to-k-nearest-neighbors-3b534bb11d26

https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d

https://towardsdatascience.com/k-nn-getting-to-know-your-nearest-neighbors-b60399dc0f32

https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

https://www.geeksforgeeks.org/k-nearest-neighbours/

http://people.csail.mit.edu/dsontag/courses/ml13/slides/lecture11.pdf

https://en.wikipedia.org/wiki/Nearest\_neighbor\_search

<https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f>

<https://blog.nus.edu.sg/soctalent/2010/02/11/a-brief-summary-of-lexrank-graph-based-lexical-centrality-as-salience-in-text-summarization/>

https://www.researchgate.net/publication/257947528\_Text\_SummarizationAn\_Overview