# Event Timeline Generation and Summarization using PageRank


Pabani Das (Roll.No. CS1722)

M.Tech in Computer Science 2017-19

Indian Statistical Institute

*Supervisor*

Dr. Debapriyo Majumdar

In partial fulfillment of the requirements for the degree of

*Master of Technology in Computer Science*

July 8, 2019

# CERTIFICATE

This is to certify that the dissertation titled **"Event Timeline Generation and Summarization using PageRank"** submitted by **Pabani Das**, Roll Number **CS1722** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** embodies the work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

---

**Debapriyo Majumdar**
Assistant Professor,
Computer Vision and Pattern Recognition Unit,
Indian Statistical Institute,
Kolkata-700108, India.

# Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance. I would first like to thank my supervisor, **Dr. Debapriyo Majumdar**, whose expertise was invaluable in the formulating of the research topic and methodology in particular. I want to thank you for your excellent cooperation and for all of the opportunities I was given to conduct my dissertation.

I would like to acknowledge **Computer Vision and Pattern Recognition Unit, ISI Kolkata** for their wonderful cooperation. Department supported me greatly and were always willing to help me with the tools that I needed to choose the right direction and successfully complete my dissertation.

# Abstract

Given a search query our system retrieves most significant events related to the query and display them in user friendly timeline. We are using graph structure to compute relative importance of events. We are representing event and its elements as complete graph. To rank events we are linearly combining basic IR ranking and ranking of event elements. evaluation on DUC Data experimental results shows that our system performs at par with the NLP based summarization techniques and yet works as fast as a search engine.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Problem Definition

Search Engine is a System Software that is designed to carry out a search may be with a predefined database or in world wide web(Internet) in a systematic way for a particular specified information given as a search query. A Search Engine may be online or offline. If a search engine runs its search procedure through internet or World Wide Web then its a online searching. When a search engine finds the result for a given query from the fixed predefined database then its a offline searching. There are some very well known search engine for example, Google, yahoo, bing etc.

We are aiming to present system for searching historical events. Given any arbitrary search query, this system will retrieve k most significant events and display the information about them in timeline. Our target is to model a automatic event timeline generator for arbitrary search query.

Event Timeline Generation and Summarization using PageRank represents a search engine which produces the required information with event date for a given

search query. This search engine accepts query in form of question or topic query. And gives the result in the form of information about the k most significant events about the query topic, not a list of articles. A information produced as result will contain important incidents with its dates like war date, awarded at, won, realised at etc. For Example, if an user wish to search for query like 'Crimes in kolkata' then the search engine will produce top rated crimes happened in kolkata with its dates.

## 1.2 Motivation

Regular search engine provides list of most significant article ranked based on some defined ranking method. But a single article may not contains all important information related to a search query. To cover all the important information a user may have to go through a huge number of articles and search links. But still user cant ensure to have all significant information on a search topic.

When we talk about historical events, information of any historical events are vary much distributed over the internet. So to get knowledge on a specific historical topic we need to go through many articles and summarise the information. This may take a lot of time and effort to get all significant information on a specific topic in a single place. This is kind of impossible to ensure all significant information on a query topic in a single run or in a single article with regular search engine.

NLP based sophisticated techniques are likely to be slow. They work for a given topic. Our approach works as fast as a search engine is expected to return results

Our system ensure all significant information related to a query topic in a single run and displays results in a timeline view. Except showing links of articles to user our system shows the actual information of significant events related to the query.

The coverage of the system should not be limited, the user interface should be easy to use, and the response time should be as fast as the popular search engines we are used to, so that it can be actually used by users, in particular young students interested in history or current affairs, or anyone who is not an expert in some topic but is interested in knowing about the events related to the topic.

## 1.3 Difference with Traditional search Engine

### 1.3.1 Traditional Search

A Traditional Search Engine searches the given query in a systematic way for particular information. Search results are generally presented in line of results often referred as search results pages. The information may be mix of web pages, images, videos, graphics, research pages, articles etc. Traditional search engine allows internet users to search for contents via World Wide Web. When a user enters some query to search engine, the user receives the list of contents returned by the search engine which is known as Search Engine Result Page(SERP).

The main component of the SERP is the listing of results that are returned by the search engine in response to a keyword query, although the pages may also contain other results such as advertisements.The results are of two general types, organic search (i.e., retrieved by the search engine's algorithm) and sponsored

search (i.e., advertisements). The results are normally ranked by relevance to the query. Each result displayed on the SERP normally includes a title, a link that points to the actual page on the Web, and a short description showing where the keywords have matched content within the page for organic results. For sponsored results, the advertiser chooses what to display. Due to the huge number of items that are available or related to the query, there usually are several pages in response to a single search query.

When a user gets SERP as query results its very difficult for a user to get all significant information about a single query via a single article link. So, for sick of all relevant significant information about a query the user need to go through multiple links and summarize all those links to get the ultimate information.

### 1.3.2 Timeline Search

Our search engine performs a searching mechanism on a given predefined dataset. When a user enters some query the search engine find all relevant and significant information related to the query. The search engine displays all the relevant information on the query in a timeline. So an user gets all the significant event information in a single page with event date. Thus a user need not to traverse through a number of articles and analyse information and dates to get all the significant event details related to a given query.

Our search engine makes the searching and information gathering easy for a user. With this timeline search engine user will be capable to get all relevant significant information on any historical event in a single hit.

## 1.4 Similarity with Document Summarization

Our model performs some document summarization technique on given dataset. To show most important events related to given query the system needs to go through all the information related to query and find out which information is most important and relevant with given query. This is very much expected that some information may be distributed through multiple consecutive lines or through a paragraph. For example one line contains name of some very important person, second line contains some important date of some event and third line contains other important person names related with the event mentioned in a previous line. Now if an user submits a query to get information about the event then the complete information is distributed over all three lines. So we need to use some summerization technique to extract significant information and produce the desired result.

Our work is not only text summerization but something more than that. Here in this work we are combining event extraction, date extraction, entity extraction, text ranking, indexing, summarizing and searching all together to develop a system which makes searching easy and interactive for user.

## 1.5 Objectives

Here we are planning to design a search engine which gives a crisp summarized event information on some given query. This search engine shows most relevant and significant events on given query. For example,
Given Query:
**Terrorist attacks in india**
We are expecting the result:

1. June 12,2019: Five CRPF jawans were martyred and three others were injured on Wednesday on terror attack in Jammu and Kashmir.

2. An attach took place at Dantewada, Chhattisgarh and 5 lives ware sacrificed.

3. March 7, 2019: 3 people ware dead in grenade blast at Jammu Bustand in Jammu and Kashmir.

4. February 14, 2019: a convoy of vehicles carrying security personnel on the Jammu Srinagar National Highway was attacked by a vehicle-borne suicide bomber at Lethpora (near Awantipora) in the Pulwama district, Jammu and Kashmir, India.

5. September 18, 2016: There was an attack by four heavily armed militants on 18 September 2016, near the town of Uri in the Indian state of Jammu and Kashmir.

6. January 2, 2016: There was a terrorist attack committed on 2 January 2016 by a heavily armed group which attacked the Pathankot Air Force Station, part of the Western Air Command of the Indian Air Force.

7. June 4, 2015: United Liberation Front of Western South East Asia insurgents ambushed a military convoy in Chandel district on 4 June 2015, resulting in the loss of life for eighteen soldiers of the Indian Army.

8. December 23, 2014: In December 23, 2014, a series of attacks by militants resulted in the deaths of more than

76 people in India.

9. February 21, 2013: On 21 February 2013, at around 19:00 IST, two blasts occurred in the city of Hyderabad, India.

10. July 13, 2011: The 2011 Mumbai bombings were a series of three coordinated bomb explosions at different locations in Mumbai, India, on 13 July 2011 between 18:54 and 19:06 IST.

# Chapter 2

# Related Work

In *Automatic generation of overview timelines* [1] by Russell Swan and James Allan proposed a statistical model to automatically extract the most significant topics from a corpus. They developed a technique for determining relative importance of the occurrence of extracted features within text. Their interest was in using timelines as a browsing interface to a document collection.

In *Query based event extraction along a timeline* [2] author have presented a framework and a system that extracts events relevant to a query from a collection of documents and and places such events along a timeline. This paper summarizes a big collection of documents that could have been returned by a query-based search, by placing sentences that report "important" events related to the query along a timeline. For the experiment they have used articles from the first 6months of 2002 from the English Gigaword corpus which is very small and covering a very short time-span.

*DIGITALHISTORIAN: Search & Analytics Using Annotations* [3] demonstrates a retrieval system, that analyzes document collections using semantic annotations in the form of temporal expressions and named entities linked to a knowledge

graph. In [4] the statistics of frequently occurring temporal expressions in highly relevant documents was analysed given a keyword query. [5] presents a probabilistic framework that leverages semantic annotations in the form of temporal expressions, geographic locations, and named entities to analyze natural language text and determine important events.

Another work [6] first identifies time intervals of interest to the given keyword query based on pseudo-relevant documents. It then re-ranks query results so as to maximize the coverage of identified time intervals. In these discussed works whole document collections were extracted. [7] proposes methods for clustering search results by time.[8] presents an information retrieval applications in which various temporal information are exploited associated with documents to present and cluster documents along timelines.

In paper [9] the problem of generating storylines from microblogs was explored for user input queries. This problem was challenging due to the sparse, dynamic and social nature of microblogs. [10] presents an approach that exploits the headlines of online news articles instead of the articles' full text. News headlines and microblog posts containing crisp summaries of events occurring in the recent times, but they would lack information about events in the past, for which news headlines or tweets are not available.

The goal of [11] is extraction and retrieval of local events from web pages. Here events collected in the form of retrievable calendar entries that include structured information about event name, date, time and location. For example, Yago [12] is a knowledge base which contains relations and facts (including events) related to entities.

Our system includes event-based multi-document summarization.In paper [13], a novel approach is proposed to automatic generation of aspect-oriented summaries from multiple documents. [14] presents multidocument summarization to generate a summary which includes the main points from an input collection of documents with minimal repetition of similar points. [15] produces trans-temporal correlations among component summaries for timelines, using inter-date and intra-date sentence dependencies. In [15] and [16] the topics are pre-decided, a set of related documents are retrieved based on the topics.

In paper [17] author studies the abstractive multi-document summarization for event-oriented news texts through event information extraction and abstract representation. The abstractive summarization for event-oriented news texts is made by extracting fine-grained events and constructing event semantic link network as the abstract representation of source texts. They experimented on DUC 2006 and DUC 2007 datasets. In the work [18] it has been examined that how elementary discourse units (EDUs) from Rhetorical Structure Theory can be used to extend extractive summarizers to produce a wider range of human-like summaries. In paper [19], the problem of extracting summary sentences from multi-document sets by applying sparse coding techniques. Based on the data reconstruction and sentence denoising assumption, a two-level sparse representation model is presented to depict the process of multi-document summarization. In [20], a new extractive multi-document summarization method is introduced that uses ILP to jointly optimize the importance of the summary's sentences and their diversity (nonredundancy), also respecting the maximum allowed summary length. System described in [21] produces an abstractive summary for a set of topic related documents. It consists of two major components: Information extraction and summary generation. All these paper include experiments on DUC Data set.

# Chapter 3

# Proposed Approach

In this chapter we will discuss about the approach we are proposing to achieve the goal of our work. We have selected Wikipedia Articles to form our project dataset. our approach works on any dataset having information about events, but Wikipedia is a good starting point.

## 3.1 Proposed Method

We are approaching to get result for a given query based on how important events related to that query a sentence contains. To find relative importance of a sentence and importance of named entities in that sentence.

**The algorithm has the following main steps:**

1. Annotate each sentence with dates and named entities it contains.
2. For each sentence, consider that sentence as a node, and the dates and Named Entities(NEs) as nodes.
3. For each sentence there is bi-directional edge between each pair of nodes.

So for each sentence a complete graph is formed with all nodes with in that sentence.

4. Use PageRank algorithm to compute "importance" of each node in the graph. Important events will be associated with important dates and entities and other important events in the graph, thus will have higher scores.

5. Search: combine this score with usual tf.idf kind of term scoring.

### 3.1.1 Annotate Sentence

A sentence is important if it contains valuable information. How we will check if a sentence have valuable information or not? When a sentence contains some vary important named entity and date then the sentence seems to be vary important. For Example,

```
Rabindranath Tagor won Nobel prize for literature at 1913.
```

is a vary important Sentance as it contains important named entity **Rabindranath Tagor** and **Nobel** and a important year for our country **1913**.

### 3.1.2 Nodes

Each sentence elements are represented a nodes. A sentence itself is a node. Each the Named Entities and Dates exists in that sentence is represented as individual nodes. For sentence,

```
Rabindranath Tagor won Nobel prize for literature at 1913.
```

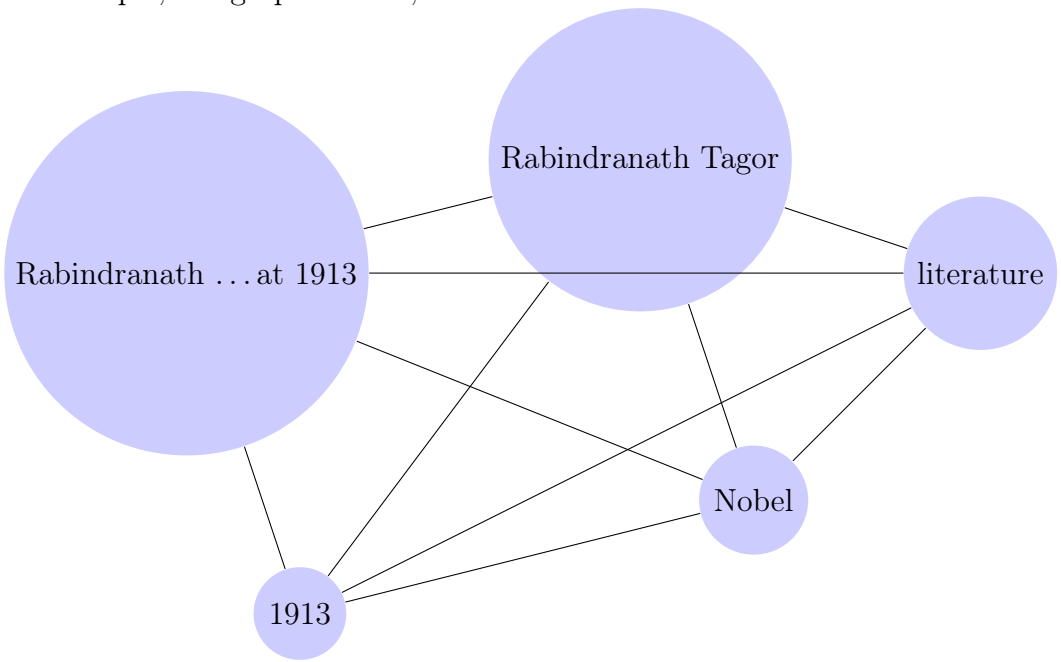
Nodes are:

```
Rabindranath Tagor won Nobel prize for literature at 1913.
Rabindranath Tagor
```

```
literature
Nobel
1913
```

### 3.1.3 Graph

We are forming a complete graph with all the nodes from a sentence. There will exists a bi-directional edge between every pair of nodes within a sentence. From above example, the graph will be,




### 3.1.4 PageRank on Graph Nodes

PageRank (PR) is an algorithm used by Google Search to rank web pages in their search engine results. PageRank is a way of measuring the importance of website pages. PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from

other websites.

PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set.

PageRank is used primarily for ranking web pages in online search results. Here we are using the PageRank technique on *Graph* described in previous section 4.2.3.

**Similarity Between PageRank on Web Pages and PageRank on graph nodes are:**

- In place of web pages, we use sentences.
- Similarity between two consecutive sentences, sentence and Named Entity belongs to this, Sentence and Date belongs to it, two Named Entity belongs to same sentence, Named Entity and Date belongs to same sentence etc is used as an equivalent to the web page transition probability.
- The similarity scores are stored in a square matrix, similar to the matrix M used for PageRank.

PageRank on text terms mainly used for Text Summarization. As our work have some summarization task also, we are using PageRank on graph Nodes to find the impotence of a sentence respect to the Named Entities within this sentence.

Importance of a sentence is decided by what type of information it caries and about whom or which. when a sentence have named entities those have great importance then the sentence must be important. Lets assume a sentence represents a important event about a vary important person. So the sentence importance

is vary high. Also if a named Entity arrives in many sentences then the importance for named entity is also vary high. PageRank assigns high value for those nodes which are connected with other nodes with high value. So Each sentence contains nodes with high rank get high importance reversely existing nodes of an important sentence also gets high rank.

### 3.1.5 Search

For searching we need to score sentences for extracting meaningful results. By combining this PageRank scores with usual tf.idf kind of term scoring sentences are given some ranking score, based on which the searching is done.

## 3.2 Indexing

Indexing forms the core functionality of the IR process since it is the first step in IR and assists in efficient information retrieval. The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus, which would require considerable time and computing power.

Here we are indexing sentence tuples. To create the index we are rewriting the tuples in a ways that each line contains three elements $Sentence_ID$, $Text$ and $Event_Value$ separated by **tab**.

```
<Sentence_ID>   <Text>   <Event_Value>
```

Example:

```
568    Rabindranath Tagor won Nobel prize for literature
at 1913.   5825.3698
```

### 3.2.1 Event Value

Event Value $S_e(q, e)$ represents the importance of a sentence. Each sentence is an Event. Each Event contains some Named Entity or Dates or both, called elements of the sentence. Here we are deciding the importance of sentence depending on the rank score $S_{PR}$ of each elements of the sentence.

$$S_e(q, e) \ = \ \sum_{i=0}^{N} S_{PR}(i)$$

where,

$N \ = \ Number\ of\ elements\ in\ event\ e$

$RankScore \ = \ PageRank\ Score\ of\ i^{th}\ element\ of\ event\ e$

## 3.3 Ranking Method

Given a query $q$, we are ranking the result based on $Lucene_Score\ S_l(q, e)$and $Event\_Value\ S_e(q, e)$ discussed in Section 4.2.1 for Event $e$. The final score S(q, e) we use for event ranking is obtained by combination of $Lucene\_Score$ and $Event\_Value$, as

$$S(q, e) = S_l(q, e) + \log\left(\frac{S_e(q, e)}{l}\right)$$

where l = length of event

Lucene_Score(`https://lucene.apache.org/core/3_5_0/api/core/org/apache/lucene/search/Similarity.html`) combines **Boolean model(BM)** of Information Retrieval with **Vector Space Model(VSM)** of Information Retrieval.

### 3.3.1 Lucene Score

Lucene combines **Boolean model(BM)** of Information Retrieval with **Vector Space Model(VSM)** of Information Retrieval.

VSM score of Event $e$ for query $q$ is the Cosine Similarity of the weighted query vectors $V(q)$ and $V(d)$:

$$cosine\_similarity(q, e) = \frac{V(q) \cdot V(e)}{|V(q)||V(e)|}$$

Where$V(q)\Delta V(e)$ is the dot product of the weighted vectors, and $|V(q)|$ and $|V(e)|$ are their Euclidean norms.

Lucene's Practical Scoring Function is,

$$Lucene\_Score = coord(q, e) \cdot queryNorm(q) \sum_{t \in q} (tf(t \in e) \cdot idf(t)^2 \cdot t.getBoost() \cdot norm(t, e))$$

where,

- tf(t in d) correlates to the term's frequency, defined as the number of times term t appears in the currently scored event e.
- idf(t) stands for Inverse Document Frequency.
- coord(q,d) is a score factor based on how many of the query terms are found in the specified event.
- queryNorm(q) is a normalizing factor used to make scores between queries comparable.
- t.getBoost() is a search time boost of term t in the query q as specified in the query text.

- norm(t,d) encapsulates a few (indexing time) boost and length factors:

  Document boost - set by calling doc.setBoost() before adding the sentence(event) to the index. Field boost - set by calling field.setBoost() before adding the field to a event. lengthNorm - computed when the event is added to the index in accordance with the number of tokens of this field in the event, so that shorter fields contribute more to the score.

# Chapter 4

# Experiments

## 4.1 Dataset

### 4.1.1 Wikipedia

Wikipedia is a multilingual online encyclopedia, based on open collaboration through a wiki-based content editing system. It is the largest and most popular general reference work on the World Wide Web and is one of the most popular websites ranked by Alexa as of June 2019. It features exclusively free content and no commercial ads, and is owned and supported by the Wikimedia Foundation, a non-profit organization funded primarily through donations.

In our experiment we are using English Wikipedia dump version of april 2019(`https://dumps.wikimedia.org/enwiki/20190301/`) as raw dataset.

**File Format:** This Wikipedia dump includes Articles, templates, media/file descriptions, and primary meta-pages. This dump is in XML format which zipped in a bz2 format.

**File Size:** Complete size of the wikipedia dump is 14.5 GB which is in Xml.bz2 zipped format. It contains 56 zipped xml files each containing millions of documents.

```
=== Origins ===
&lt;!-- Anarcho-communist Joseph Déjacque, the first person to use the term &quo
t;libertarian&quot; in a political sense and self-proclaimed advocate of liberta
rianism, needs to be added here. His work and stances on anarchism are very rele
vant to this particular section of the article. Additionally, his criticisms of
Pierre-Joseph Proudhon's mutualism are very relevant here. --&gt;
The earliest anarchist themes can be found in the 6th century BC among the works
 of [[Taoism|Taoist]] philosopher [[Laozi]]&lt;ref&gt;{{harvnb|Kropotkin|1910}}:
 ''&quot; At the same time it evidently found its expression in the writings of
some thinkers, since the times of Lao-tsze, although, owing to its non-scholasti
c and popular origin, it obviously found less sympathy among the scholars than t
he opposed tendency&quot;''&lt;/ref&gt; and in later centuries by [[Zhuang Zhou|
Zhuangzi]] and Bao Jingyan.{{sfn|Graham|2005}} Zhuangzi's philosophy has been de
scribed by various sources as anarchist.&lt;ref name=&quot;ZhuangziAnarchist&quo
t;&gt;
* {{cite web | last=Hansen | first=Chad | title=Taoism | website=plato.stanford.
edu | date=February 19, 2003 | url=http://plato.stanford.edu/entries/taoism/ | a
rchive-url=https://web.archive.org/web/20130624092211/http://plato.stanford.edu/
entries/taoism/ | archive-date=June 24, 2013 | dead-url=yes | ref=harv | access-
date=September 17, 2018|quote =&quot;The priority of dao over tiannature:sky und
erwrites the themes of dependency and relativism that pervade the Zhuangzi and u
ltimately the skepticism, the open-minded toleration and the political anarchism
 (or disinterest in political activity or involvement).&quot; [http://plato.stan
ford.edu/entries/taoism/ &quot;Taoism&quot; at the Stanford Encyclopedia of Phil
```

Figure 4.1: Wikipedia English Dump XML View

Wikipedia dataset in unstructured and distributed through different articles. But Wikipadia have a descent collection of historical data. So for our experiment we have choose wikipedia dump as our data corpus. But there is a problem to use the raw xml file as corpus. As we can see in Figure 4.1 the raw data have a lots of tags, references, url links and symbols which are meaning less and confusing for an user.So this raw xml file cant be used directly. We need to clean this xml files to make our corpus. We have already discussed the cleaning process in 3.

### 4.1.2 DUC PAST DATA

The Document Understanding Conference (DUC) is a series of summarization evaluations that have been conducted by the National Institute of Standards and Technology (NIST) since 2001. Its goal is to further progress in automatic text summarization and enable researchers to participate in large-scale experiments in both the development and evaluation of summarization systems.

Here we are using DUC PAST DATA 2004, DUC PAST DATA 2005, DUC PAST DATA 2006, DUC PAST DATA 2007. The data format is shown in Figure 4.2.

```
<DOC>
<DOCNO> XIE19980304.0061 </DOCNO>
<DATE_TIME> 1998-03-04 </DATE_TIME>
<BODY>
<HEADLINE> Report: Hate Groups Growing in United States </HEADLINE>
<TEXT>
<P>
WASHINGTON, March 3 (Xinhua) -- The number of organized hate
groups in the United States grew last year, mostly through new
chapters of established white power organizations, the Southern
Poverty Law Center said in a report released Tuesday.
</P>
<P>
The research group based in Mongomery, Alabama, said 474 hate
groups and their chapters engaged in some form of racist behavior
in 1997, The Associated Press reported.
</P>
<P>
The center used a different methodology to count the groups so
previous numbers cannot be directly compared. But if the old
methodology had been used, the latest figures would represent a 20
percent increase from 1996.
</P>
<P>
```

Figure 4.2: DUC PAST DATA 2007

**File Format:** The DUC Data represented in two parts Main and Update. Both parts have multiple files of the format shown in Figure 4.2. DUC text data contains Document No, Date Time, Category (Country), Headline and main Text. DUC mainly based on newspaper articles. DUC Main part contains the main

original articles and Update part contains next occurred events or actions on those topics.

**File Size:** number of queries in DUC 2007 PAST DATA is 1126.

## 4.2 Processing of Data

We are using Wikipedia dump and DUC PAST DATA. But the raw Wikipedia and DUC date both have some tags which is not required. Initially we had raw wikipedia data which contains article text, external links, images, references etc. We need only the text article for our experiment. Other url links, images, tags etc are useless noise for our dataset.

### 4.2.1 Cleaning

Wikipedia Dump files contains Articles, templates, media/file descriptions, and primary meta-pages. wikipedia downloaded as a *xml.zip* file. To fulfil our requirement we needed clean text file without tags, symbols and external links. Therefore we directly downloaded the zipped file then decompressed that to get *xml* file. After getting xml file we cleaned the xml file using python *wikidumpcleaner* and extracted the cleaned text corpus.

Cleaning Result:

After cleaning the wikipedia dump we get cleaned text shown in Figure 4.3

### 4.2.2 Generate Tuples

After cleaning the text we are concentrating to extract dated events from the text. Our concern is to extract those event information which contains some date

```
TITLE: Autism
Autism is a developmental disorder characterized by troubles with social interac
tion and communication, and by restricted and repetitive behavior. Parents usual
ly notice signs during the first two or three years of their child's life. These
 signs often develop gradually, though some children with autism reach their dev
elopmental milestones at a normal pace before worsening.
Autism is associated with a combination of genetic and environmental factors. Ri
sk factors during pregnancy include certain infections, such as rubella, and tox
ins including valproic acid, alcohol, cocaine, pesticides and air pollution. Con
troversies surround other proposed environmental causes; for example, the vaccin
e hypotheses, which have been disproven. Autism affects information processing i
n the brain by altering how nerve cells and their synapses connect and organize;
 how this occurs is not well understood. In the DSM-5, autism and less severe fo
rms of the condition, including Asperger syndrome and pervasive developmental di
sorder not otherwise specified (PDD-NOS), have been combined into the diagnosis
of autism spectrum disorder (ASD).
Early speech or behavioral interventions can help children with autism gain self
-care, social, and communication skills. Although there is no known cure, there
have been cases of children who recovered. Not many children with autism live in
dependently after reaching adulthood, though some are successful. An autistic cu
lture has developed, with some individuals seeking a cure and others believing a
utism should be accepted as a difference and not treated as a disorder.
Globally, autism is estimated to affect 24.8 million people as of 2015. In the 2
000s, the number of people affected was estimated at 1–2 per 1,000 people worldw
```

Figure 4.3: Cleaned Wikipedia XML

and Named entity and then generating tuple respect to each sentence. We are considering the sentences of whole paragraph which contains some date. Intuition behind considering the whole paragraph is A complete meaning full information may be distributed in several lines. If we considered only single lines which have Date or Name entity then the summarized information may be scattered and we ended up with some information result which are incomplete and meaningless. So we are generating text file from wikipedia cleaned text files, where each line contains four tuples. Our target is to lookup on each sentence and find if a sentence have any *Date* or not. If a sentence have *Date* then find all *Named Entities* from that sentence. A sentence which contains *Date* and *Named Entity* that sentence must contains information about some *Event*. Tuples are defined as:

```
[<Sentence_ID>, <Sentence_Text>, <List_Of_Named_Entities>,
 <List_Of_Dates>]
```

Example:

Sentence:

```
The Indian National Army (Azad Hind Fauj) was an armed force
formed by Indian nationalist Rash Bihari Bose in 1942 in
Southeast Asia during World War II.
```

Generated Tuple:

```
['d1p398s286', 'The Indian National Army (Azad Hind Fauj) was an
armed force formed by Indian nationalist Rash Bihari Bose in 1942
in Southeast Asia during World War II..', [['PERSON', 'Rash Bihari
Bose'], ['ORGANIZATION', 'Indian National Army'], ['ORGANIZATION',
'Azad Hind Fauj'], ['ORGANIZATION', 'Indian nationalist'],
['ORGANIZATION', 'World War II'],['GPE', 'Southeast Asia']],
['1942-01-01']]
```

sentence id **d1p398s286** means It is $286^{th}$ sentence of $398^{th}$ paragraph of $1^st$ document.

### 4.2.3 Graph Formation

The Goal behind this graph formation is, to find the relativity between Sentence, Date and Named Entity. Each Sentence have some Dates and Named Entities. If a sentence have a Date and Named entity which also exists many in another Sentences then the sentence must be very important with respect to that named entity. With this Graph we are actually trying to find out the impotence of each Named Entity and impotence of sentences respect to those Named Entity. This graph is represents the connection of each sentence with its containing *Date, Named Entities*. This graph also shows the connection of each *Named Entity* and sentences and dates related to that *Named Entity*.

Here each elements of a *Tuple* except the $Sentence_Text$ is considered as a *Node*. i.e For sentence $Sentece_ID$ is a node, each entities of $List_Of_Named_Entities$ are nodes and each Date of $List_Of_Dates$ are nodes. Between each nodes from a sentence have bidirectional edges. Each $Sentence_ID$ node have an edge with the previous sentence of that sentence from the same paragraph to indicate consecutive sentences. For Example,

**Sentence:**

```
['d1p33s23', ' Azad Hind Fauj was revived under the leadership of
Subhash Chandra Bose  in 1943.',[['ORGANIZATION',
'Azad Hind Fauj'],
['PERSON', 'Subhash Chandra Bose']], ['1943-01-01']]
```

**Nodes:**

```
d1p33s23, Azad Hind Fauj, Subhash Chandra Bose, 1943-01-01
```

**Graph Representation**

```
d1p33s22    d1p33s23
d1p33s23    d1p33s22
d1p33s23    Azad Hind Fauj
Azad Hind Fauj    d1p33s23
d1p33s23    1943-01-01
1943-01-01    d1p33s23
Azad Hind Fauj    1943-01-01
1943-01-01    Azad Hind Fauj
1943-01-01    Subhash Chandra Bose
Azad Hind Fauj    Subhash Chandra Bose
```

```
d1p33s23    Subhash Chandra Bose
```

In the above Graph each line represents an edge where each nodes are separated by **tab**.Node **d1p33s22** is the previous sentence of sentence **d1p33s23**. Format of the Graph file is .tsv. Here we are connecting two consecutive sentences because there may be some cases where a sentence is incomplete without the previous sentence.

For Example,

```
In 1871, at the age of eight, Narendranath enrolled at Ishwar
Chandra Vidyasagar's Metropolitan Institution, where he went
to school until his family moved to Raipur in 1877. In 1879,
after his family's return to Calcutta, he was the only student
to receive first-division marks in the Presidency College
entrance examination.
```

Here the second sentence have incomplete information alone. This sentences gives a complete information when the first and second sentences are shown together. Considering this kind of co-referencing cases we the connecting two consecutive sentences with a bidirectional edge.

```
d1p33s22    d1p33s23
d1p33s23    d1p33s22
```

## 4.3 Evaluation Strategy

Our goal is to model the search engine on wikipedia data. But wikipedia doesn't have any baseline implementation on query searching or summarization task. So we have no quality scale to evaluate our method on wikipedia.

To evaluate our method we have prepared the **DUC Past Data 2007**. We have processed DUC 2007 data and prepared a query searching demonstration.

**ROUGE Score:** ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation. We are using ROUGE F1-measure which is harmonic average of precision and recall.

$$recall = \frac{number\_of\_overlapping\_words}{total\_words\_in\_reference\_summary}$$

$$precision = \frac{number\_of\_overlapping\_words}{total\_words\_in\_system\_summary}$$

$$ROUGE\ F1\_measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

The topics and events summaries used for evaluation in [5] are publicly available, but the event descriptions in those summaries are long, very different from the type of crisp event descriptions we need. The ROUGE [22] measures are commonly used for evaluating summaries. Essentially, *Rouge n Summary score* (for $n \geq 1$) is the fraction of matched n-grams between a text and a gold standard.

## 4.4 Results

Our system system Top k events those are most significant and relevant respect to the query.Here k is an user defined number which indicates number of most significant events user wants in result. For each run we are taking different k values and evaluating the Rouge score. We are also comparing generated results for

four different scoring method.We are comparing ground truth(reference) with our results(hypothesis). Here we are using Rouge n-gram Summery Score F-Measure for unigram, bigram, trigram, and 4-gram to Evaluate the query results. Results using different ranking score:

| Rank Score | Rouge-1 | Rouge-2 | Rouge-3 | Rouge-4 |
|---|---|---|---|---|
| Lucene | 0.54964 | 0.50990 | 0.32674 | 0.20040 |
| Lucene + log((Event Value)/length) | **0.568110** | **0.53736** | **0.41456** | **0.25913** |
| Lucene + (Event Value/length) | 0.57038 | 0.51199 | 0.38049 | 0.24792 |
| Lucene + Event Value | 0.51768 | 0.46560 | 0.34262 | 0.20952 |

Table 4.1: Rouge 4-gram summary score f-measure of query results. Different ranking scores are used

Here we are extracting most significant $k$ events. Effectiveness of search result is dependent on $k$ value. So we have scored the results for different $k$ values.

| Rank Score | k=100 | k=50 | k=20 |
|---|---|---|---|
| Lucene Score | 0.24437 | 0.25701 | 0.20884 |
| Lucene Score + log((Event Value)/length) | **0.27468** | **0.30438** | **0.26729** |
| Lucene Score + (Event Value/length) | 0.25893 | 0.273025 | 0.21185 |
| Lucene Score + Event Value | 0.28754 | 0.23002 | 0.20301 |

Table 4.2: Average Rouge F1-measure comparing 4-gram over 20 queries for different scoring and different k value.

We are evaluating our accuracy of results with respect to DUC past Data 2007.

Given a query:

**Supremacist Wanted Everyone Dead**

Ground Truth:

*MOBILE, Ala. (AP) – A white supremacist arrested after buying hand grenades from an undercover agent said he wanted to send mail bombs to Washington and*

*Montgomery, authorities said.*

*Chris Scott Gilliam said he didn't want to be like the Unabomber, who killed three person and wounded several others, a federal agent testified at a federal court hearing.*

*"He wanted to kill everybody," David Pasqualotto, a special agent with the Bureau of Alcohol, Tobacco and Firearms, said Friday.*

*There was no immediate indication what agencies or people might have been targeted in the national and state capital cities. An ATF representative did not immediately return calls for comment Saturday.*

*Gilliam, 27, was charged with possessing an unregistered firearm found at his home in Foley along with what agents said was apparently a silencer. Defense attorney Gary Arm said at the hearing that Gilliam legally owned the firearm and has no criminal record.*

*A federal magistrate ordered him held pending the results of a psychological evaluation.*

*If convicted, Gilliam, could face six or seven years in prison, said Assistant U.S. Attorney Greg Bordenkircher. No charges were immediately filed in connection with the grenade purchase.*

*Pasqualotto testified that Gilliam met with him Thursday, arriving with a loaded, cocked .45-caliber pistol on his hip and his 20-month-old son in his arms.*

*"I knew he wasn't quite right," Pasqualotto said.*

*The investigation started three weeks earlier when Gilliam contacted a man about obtaining C4 high explosive, Pasqualotto testified.*

*He said Gilliam also told the informant someone should kill the FBI sniper who killed the wife of white supremacist Randy weaver during an 11-day standoff in 1992 at Ruby Ridge, Idaho, along with civil rights lawyer Morris Dees of the Montgomery-based Southern Poverty Law Center.*

**Our Result**:

*Rudolph remains on theFBI's list of the 10 most-wanted fugitives. WASHINGTON, October 14 (Xinhua) – U.S. Federal authorities onWednesday charged Eric Robert Rudolph, one of the FBI's 10most-wanted fugitives, with bombings including one blasted inthe Centennial Park in Atlanta during the 1996 Olympics. The Aryan Nations compoundhas been a base for white supremacists for more than 20 years.*

*ISTANBUL, August 22 (Xinhua) – Some 12,040 people have so farbeen confirmed dead in Tuesday's killer earthquake in northwestTurkey and 33,495 others were reported as injured, the GovernmentCrisis Center of Turkey announced Sunday morning.*

*The King family originally wanted the full $30 million, but whenLibrary officials balked, the family agreed to reduce the paymentrequest to $20 million and take a $10 million tax deduction.*

*At the end, 13 people were dead.*

*His name was added to the FBI's 10 Most Wanted list this year,and a 1 million U.S. dollars award has been offered forinformation leading to his conviction. "The Klan of yesteryear is dead," Shelton said in a 1995interview.*

*The FBI has posted a $1 million reward for Rudolph's capture,and he has been placed on the agency's 10-most-wanted list.*

*20 testimony, Lewinsky is asked by a grand juror if there isanything she wanted to add or clarify.*

*Geddie said Simpson had originally contacted Walters about doingan interview for her ABC News program, "20/20," but said,"Barbara did not want to interview him on that show."*

*Rudolph was placed on the FBI's 10 most wanted list and a reward of $1 million was posted for information leading directly to his arrest.*

*Thestandoff ended 11 days later, with Weaver wounded and his wife andson dead. King, a 24-year-old white supremacist, did have something tosay.*
*But not everyone thinks Rudolph, an experienced backwoodsman whowould have just celebrated his 33rd birthday, is still hidden inthese woods.*
*In early May, the FBI placed Rudolph on its list of the10 most wanted fugitives and offered a $1 million reward forinformation leading to his arrest.*
*She never wants to visit Atlanta again, not even for a memorialservice July 30 for Barton's victims.*
*Of the 11 wolves brought to the wilds of Arizona last year, fivewere shot dead, one is missing and presumed dead and three wererecaptured for their own protection. Butler himself and 12 other white supremacist leaders werearrested in 1987 on federal sedition charges but were acquitted attrial in Fort Smith, Ark. The Army wants to begin buying 40 interceptor missiles as afirst installment on a system that is projected to cost 15 billiondollars to acquire and 18 billion dollars to operate over 20years, the Associated Press reported Wednesday.*

There are many summarization work on DUC Data. We are evaluating our accuracy of results with respect to other summarization tasks on DUC past Data 2007.

| System | Rouge-4 Score |
|---|---|
| Our System | 0.25913 |
| EDU [18] | 0.35 |
| ILP 2 [20] | 0.17603 |
| ESLN with Coherence [17] | 0.16137 |
| BSU [21] | 15.632 |
| MDS-Sparse-div [19] | 0.11669 |
| NIST Baseline | 0.11114 |

Table 4.3: Comparing Rouge Score with other works

## 4.5 Challenges

Our main challenge was to get the wikipedia data ready. cleaned wikipedia data is approximately vary large. When we generated the graph its turns to larger size. It was impossible to calculate PageRank for this huge graph. When the matrix is forming forming to compute page rank the matrix size increases to unmanageably large. So to compute page ranks we split the wikipedia graph randomly in 50 parts. Then we calculated Pagerank for each part and get the final rank score by adding all the pagerank score for each node from each of 50 calculated Pagerank.

# Chapter 5

# Conclusion And Future Work

The proposed Event Extraction and summarization approach is effective in information extraction and achieves good performance on DUC datasets. We have created a demo version using **DUC Past Data 2007** to evaluate our proposed approach. We have implemented this system on wikipedia. this system can be developed on any data set which have event information. We can find that the approach is very effective for summarizing texts that mainly describe events. Our goal is to implement the Timeline generation system and present the system in a easy and user friendly GUI.

There is a tail of work which we have to do to get our goal:

- Design an easy and friendly GUI to make the searching easy.
- Use YAGO to get more significant crisp information about well known named entities.
- Evaluation of timeline search on Wikipedia.
- Include more data, or build a flexible data import pipepline.

# References


[1] R. Swan and J. Allan, "Automatic generation of overview timelines," *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. ACM, 49–56, 2000. 8

[2] H. L. Chieu and Y. K. Lee, "Query based event extraction along a timeline," *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. ACM, 425–432, 2004. 8

[3] J. S. Dhruv Gupta and K. Berberich, "Digitalhistorian: Search & analytics using annotations," *HistoInformatics at DH*, pp. 5–10, 2016. 8

[4] D. Gupta and K. Berberich, "A probabilistic framework for time sensitive search," *12th NTCIR Conference on Evaluation of Information Access Technologies*, pp. 225–232, 2016. 9

[5] J. S. Dhruv Gupta and K. Berberich, "Eventminer: Mining events from annotated documents," *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pp. ACM, 261–270, 2016. 9, 27

[6] D. Gupta and K. Berberich, "Diversifying search results using time," *European Conference on Information Retrieval*, pp. Springer, 789–795, 2016. 9

[7] M. G. Omar Alonso and R. Baeza-Yates, "Search results using timeline visualizations," *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. ACM, 908–908, 2007. 9

[8] M. G. Omar Alonso and R. Baeza-Yates, "Clustering and exploring search results using timeline constructions," *In Proceedings of the 18th ACM conference on Information and knowledge management*, pp. ACM, 97–106, 2009. 9

[9] J. L. D. W. Y. C. Chen Lin, Chun Lin and T. L, "Generating event storylines from microblogs," *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. ACM, 175–184, 2012. 9

[10] M. A. Giang Tran and E. Herder, "Timeline summarization from relevant headlines," *European Conference on Information Retrieval*, pp. Springer, 245–256, 2015. 9

[11] M. B. John Foley and V. Josifovsk, "Learning to extract local events from the web," *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. ACM, 423–432., 2015. 9

[12] G. K. Fabian M Suchanek and G. Weikum, "Yago: a core of semantic knowledge," *Proceedings of the 16th international conference on World Wide Web*, pp. ACM, 697–706, 2007. 9

[13] W. G. Peng Li, Yinglin Wang and J. Jiang, "Generating aspect-oriented multi-document summarization with event-aspect model," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. Association for Computational Linguistics, 1137–1146, 2011. 10

[14] M.-Y. K. Jun-Ping Ng, Yan Chen and Z. Li, "Exploiting timelines to enhance multi-document summarization," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 923–933, 2014. 10

[15] C. H.-X. W. X. L. Rui Yan, Liang Kong and Y. Zhang, "Timeline generation through evolutionary trans-temporal summarization," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. Association for Computational Linguistics, 433–443, 2011. 10

[16] J. O.-L. K. X. L. Rui Yan, Xiaojun Wan and Y. Zhang, "Evolutionary timeline summarization: a balanced optimization framework via iterative substitution," *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. ACM, 745–754, 2011. 10

[17] H. Z. Wei Li, Lei He, "Abstractive news summarization based on event semantic link network," *The 26th International Conference on Computational Linguistics: Technical Papers*, pp. 236–246, 11 2016. 10, 31

[18] A. S. Junyi Jessy Li, Kapil Thadani, "The role of discourse units in near-extractive summarization," *Association for Computational Linguistics*, pp. 137–147, September 2016. 10, 31

[19] Z.-H. D. He Liu, Hongliang Yu, "Multi-document summarization based on two-level sparse representation model," *Association for the Advancement of Artificial Intelligence*, 5. 10, 31

[20] G. L. Dimitrios Galanis and I. Androutsopoulos, "Extractive multi-document summarization with integer linear programming and support vector regression," *COLING 2012*, pp. 911–926, 2012. 10, 31

[21] W. Li, "Abstractive multi-document summarization with semantic information extraction," *Conference on Empirical Methods in Natural Language Processing*, pp. 1908–1913, 2015. 10, 31

[22] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol. 1, pp. Association for Computational Linguistics, 71–78, 2003. 27

[23] A. H. Rakesh Agrawal, Sreenivas Gollapudi and S. Ieong, "Diversifying search results," *Proceedings of the second ACM international conference on web search and data mining*, pp. ACM, 5–14, 2007.

[24] T. G. Jenny Rose Finkel and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. Association for Computational Linguistics, 363–370, 2005.

[25] R. M. Lawrence Page, Sergey Brin and T. Winograd, "The pagerank citation ranking: Bringing order to the web," *Technical Report. Stanford InfoLab*, 1999.