

# EVENT EXTRACTION FROM DIGITAL MEDIA

Ph.D. THESIS

*by*

SWATI GUPTA



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
ROORKEE - 247 667 (INDIA)  
SEPTEMBER, 2018**



# EVENT EXTRACTION FROM DIGITAL MEDIA

A THESIS

*Submitted in partial fulfillment of the  
requirements for the award of the degree*

*of*

**DOCTOR OF PHILOSOPHY**

*in*

**COMPUTER SCIENCE & ENGINEERING**

*by*

**SWATI GUPTA**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
ROORKEE - 247 667 (INDIA)  
SEPTEMBER, 2018**





**@INDIAN INSTITUTE OF TECHNOLOGY ROORKEE, ROORKEE-2018  
ALL RIGHTS RESERVED**





# INDIAN INSTITUTE OF TECHNOLOGY ROORKEE ROORKEE

## CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled “**EVENT EXTRACTION FROM DIGITAL MEDIA**” in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Computer Science and Engineering of Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out during the period from December, 2014 to September, 2018 under the supervision of Dr. Sugata Gangopadhyay, Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee and Dr. Biplab Banerjee, Assistant Professor, Centre of Studies in Resources Engineering, Indian Institute of Technology Bombay, India.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

(**SWATI GUPTA**)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

(**Sugata Gangopadhyay**)  
Supervisor

(**Biplab Banerjee**)  
Supervisor

The Ph.D. Viva-Voce Examination of **SWATI GUPTA**, Research Scholar, has been held on ....., **2019**.

**Chairman, SRC**

**External Examiner**

This is to certify that the student has made all the corrections in the thesis.

(**Sugata Gangopadhyay**)  
Supervisor

(**Biplab Banerjee**)  
Supervisor

Date:

**Head of the Department**







*This Thesis is dedicated to  
The Almighty,  
My Teachers,  
My Friends &  
My Family Members.*



# Acknowledgements

First and foremost, I praise my Eiest Dev, the Almighty, for providing me this opportunity and granting me the capability to proceed successfully. This thesis appears in its current form due to the assistance and guidance of several people. It is a pleasure to thank all those who made this thesis possible.

It is difficult to overstate my gratitude to my Ph.D. supervisors, Dr. Sugata Gangopadhyay (Professor, Department of Computer Science and Engineering, IIT Roorkee) and Dr. Biplab Banerjee (Assistant Professor, Centre of Studies in Resources Engineering, IIT Bombay). With their enthusiasm, inspiration, and great efforts to explain things clearly and simply, they helped to make *event extraction task* fun for me. I am also thankful to them for providing me an excellent work space, necessary resources, and friendly environment to carry out my research in a lucid manner. They helped me get on the road to *LaTeX* and provided an experienced ear for my doubts about writing a thesis. Throughout my research period, they provided invaluable inspiration, incessant encouragement and appreciation, sound advice, good teaching, good company, and lots of good ideas. Their dedication towards work, the method of problem-solving, and helping nature will continue to inspire me for the rest of my career.

I would like to express my deep sense of gratitude to my research committee members Dr. Rajdeep Niyogi (Associate Professor, Computer Science and Engineering), and Dr. Bhavesh Balja (Associate Professor, Electrical Engineering) IIT Roorkee for their continuous support and invaluable suggestions. I am also thankful to Dr. Dhaval Patel, research staff member at IBM TJ Watson, New York, USA, for his support and guidance. Also, a special thanks to our departmental faculties for their kind and helpful nature that motivated me to work.

## Acknowledgements

---

In my daily work, I have been blessed with a friendly and cheerful group of colleagues. I am thankful to Dr. Pankaj Pratap Singh, Dr. Shitala Prasad, Dr. Niyati Baliyan, Jayendra Barua, Keerti, Sujata Swain, Anshul Arora, Vikas Chouhan, Brijraj Singh, Nikita Jain, Radhika Gour, Shivani Sharma and Swati Bajaj for spending their valuable time for providing a good atmosphere and useful discussions. I will really miss the funny and healthy moments spent at Nescafe and hostel canteen with them.

I am extremely thankful to the buddies of UGPG Lab for their painstaking involvement during the research work and the joyful gatherings during the stay at IIT Roorkee. I thank the support of Dr. Raj Kumar Saini, Prateek Keserwani, Tofik, and Pallavi Kaushik for their help during the preparation of the thesis.

Finally, I extend my gratitude to my loving grandparents Shri R. P. Gupta and Smt. Shanti Gupta, my parents Shri R. C. Gupta and Smt. Anita Gupta, my elder sister Dr. Shweta, my brother in law Dr. Abhishek, my younger sister Shiwani, younger brother Naman, and all my relatives for their love, consistent support and patience even during hard times of the study. I am short of words, to express my loving gratitude to my cousins Aparna, Anjali, Anirudha, and Shivam, for their innocent smiles which inspired me during entire work.

**SWATI GUPTA**

# Author's Publications

## Journals

1. **Swati Gupta** and Dhaval Patel, “*NE<sup>2</sup> : named event extraction engine*,” Knowledge and Information Systems, May 2018.
2. **Swati Gupta** and Biplab Banerjee “*Unsupervised Event Detection using Self-learning based Max-margin Clustering: Analysis on Streaming Tweets*,” IETE Journal of Research, August 2018.

## Conferences

1. Nikita Jain, **Swati Gupta**, and Dhaval Patel. 2016 . “*E<sup>3</sup>: Keyphrase based News Event Exploration Engine*”. In Proceedings of the 27<sup>th</sup> ACM Conference on Hypertext and Social Media (HT’16). ACM, New York, NY, USA, 327–329.
2. **Swati Gupta**, Sagun Sodhani, Dhaval Patel, and Biplab Banerjee, “*News Category Network based Approach for News Source Recommendations*”. In Proceedings of the 7<sup>th</sup> International Conference on Advances in Computing, Communications and Informatics (ICACCI’18). Bangalore, India.



# List of Abbreviations



<b>ADRs</b>	<b>A</b> dverse <b>D</b> rug <b>R</b> eactions
<b>AMR</b>	<b>A</b> bstract <b>M</b> eaning <b>R</b> epresentations
<b>API</b>	<b>A</b> pplication <b>P</b> rogram <b>I</b> nterface
<b>BiLSTM</b>	<b>B</b> idirectional <b>L</b> ong <b>S</b> hort <b>T</b> erm <b>M</b> emory
<b>CNN</b>	<b>C</b> onvolutional <b>N</b> eural <b>N</b> etwork
<b>CRF</b>	<b>C</b> onditional <b>R</b> andom <b>F</b> ield
<b>DBSCAN</b>	<b>D</b> ensity- <b>b</b> ased <b>S</b> patial <b>C</b> lustering of <b>A</b> pplications with <b>N</b> oise
<b>EHRs</b>	<b>E</b> lectronic <b>H</b> ealth <b>R</b> ecords
<b>EMM</b>	<b>E</b> urope <b>M</b> edia <b>M</b> onitor
<b>GDELT</b>	<b>G</b> lobal <b>D</b> ata on <b>E</b> vents, <b>L</b> ocation and <b>T</b> one
<b>GMM</b>	<b>G</b> aussian <b>M</b> ixture <b>M</b> odel
<b>HDP</b>	<b>H</b> ierarchical <b>D</b> irichlet <b>P</b> rocess
<b>IDF</b>	<b>I</b> nverse <b>D</b> ocument <b>F</b> requency
<b>IE</b>	<b>I</b> nformation <b>E</b> xtraction
<b>IR</b>	<b>I</b> nformation <b>R</b> etrieval

## List of Abbreviations

---

<b>LEEV</b>	<b>L</b> atent <b>E</b> vent <b>E</b> xtraction and <b>V</b> isualization
<b>LEM</b>	<b>L</b> atent <b>E</b> vent <b>M</b> odel
<b>LR</b>	<b>L</b> ogistic <b>R</b> egression
<b>LSTM</b>	<b>L</b> ong <b>S</b> hort <b>T</b> erm <b>M</b> emory
<b>MEC</b>	<b>M</b> aximum <b>E</b> ntropy <b>C</b> lassifier
<b>NB</b>	<b>N</b> ave <b>B</b> ayes
<b>NER</b>	<b>N</b> amed <b>E</b> ntity <b>R</b> ecognition
<b>NLP</b>	<b>N</b> atural <b>L</b> anguage <b>P</b> rocessing
<b>POS</b>	<b>P</b> art of <b>S</b> peech
<b>RFC</b>	<b>R</b> andom <b>F</b> orest <b>C</b> lassifier
<b>RNN</b>	<b>R</b> ecurrent <b>N</b> eural <b>N</b> etworks
<b>SGD</b>	<b>S</b> tochastic <b>G</b> radient <b>D</b> escent
<b>STICS</b>	<b>S</b> earching with <b>S</b> trings, <b>T</b> hings and <b>C</b> ats
<b>SVC</b>	<b>S</b> upport <b>V</b> ector <b>C</b> lassification
<b>SVM</b>	<b>S</b> upport <b>V</b> ector <b>M</b> achine
<b>TEES</b>	<b>T</b> urku <b>E</b> vent <b>E</b> xtraction <b>S</b> ystem
<b>TF</b>	<b>T</b> erm <b>F</b> requency
<b>t-SNE</b>	<b>t</b> - <b>D</b> istributed <b>S</b> tochastic <b>N</b> eighbor <b>E</b> mbedding
<b>URL</b>	<b>U</b> niform <b>R</b> esource <b>L</b> ocator



# Abstract

Nowadays, digital media has become a source of huge amount of up-to-date information which increases exponentially day by day. The information is concealed with unstructured data and the end user cannot directly access the desired information from it. The solution to deal with this problem is to collect important facts from unstructured data and store them in such a way that it can help end user to serve their queries. The procedure of specifically organizing and consolidating data that is explicitly expressed or implied in one or more natural language documents, is known as Information Extraction (IE). Generally, the information in digital media is reported in the form of events. Events can be represented as several types such as with its specific names, change of state, situations, actions, relations, etc. Standard dataset such as ACE format represents events as a triplet of event mention, event trigger, event arguments and divides the events among eight categories. Thus, event extraction is an important task of information extraction as it helps in developing various systems like story-telling, news event exploration, social media information fusion, question answering, etc.

To tackle the information overload issue, this thesis focuses on extracting information from news media and social media (Twitter) in terms of events and related key-phrases. In particular, the subsequent problems are addressed:

- Named event extraction from news headlines dataset using a knowledge-driven approach. The knowledge-driven approach uses patterns or templates that define the expert domain-specific knowledge. The named events are enriched with their type, categories, popular durations, and popularity. The system utilizes the syntactic and semantic patterns of headlines to identify the named events. Named events are short

phrases that represent the name of events like *2016 Rio Olympic Games*, *2G Case*, and *Adarsh Society Scam*. Named events are categorized into candidate-level and high-level categories using URL information, and popular durations of named events are extracted using temporal information of news headlines.

- Key-phrase extraction from news content for the purpose of offering the news audience a broad overview of news events, with especially high content volume. Given an input query, the system extracts key-phrases and enriches them by tagging, ranking, and finding the role for frequently associated key-phrases. The system utilizes the syntactic and linguistic features of text to extract the key-phrases from the news media content (text).
- Event extraction from a large-scale Twitter repository using an unsupervised approach. The amount of acquired data from streaming media like Twitter is vast in nature. It contains readily available information regarding important events taking place during the time span. Hence, it is indeed difficult to deploy supervised learning strategies for analyzing the tweets for meaningful information extraction. On top of that, the tweets are unstructured in nature given the diversities of the end-users who put the tweets. A self-learning max-margin clustering approach which deploys the notion of Support Vector Machine (SVM) in an unsupervised setup is used to cluster semantically similar tweets.

In this thesis, machine learning algorithms and Natural Language Processing (NLP) tools are used to extract the data from news media and Twitter. For each of the previously mentioned subjects, significant literature is studied thoroughly and the limitations of some existing methods are highlighted. The main motive to select the problems defined in this thesis is to prepare the methods that solve those limitations to the feasible extent. News media data (headlines, articles, meta keywords, etc.) and Twitter data are used to evaluate the performance of the proposed methods with respect to relevant state-of-the-art methods.

**Keywords:**

Named Events, News Events, Key-phrases, Categories, Popular Durations, Tweet Clustering, News Media, Twitter Data, Max-Margin Clustering.





# Table of Contents

<b>Acknowledgements</b>	<b>i</b>
<b>List of Abbreviations</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Event Detection and Representations of Events . . . . .	3
1.2 Motivation and Research Gaps . . . . .	4
1.3 Problem Statements . . . . .	5
1.4 Research Contribution . . . . .	6
1.5 Organization of the Thesis . . . . .	8
<b>2 Literature Review</b>	<b>11</b>
2.1 Key-phrase Extraction . . . . .	11
2.1.1 Key-phrase Mining . . . . .	11
2.1.2 Topic Learning . . . . .	12
2.1.3 Limitations . . . . .	13
2.2 Event Extraction . . . . .	14
2.2.1 Data-driven Event Extraction . . . . .	15
2.2.2 Knowledge-driven Event Extraction . . . . .	19
2.2.3 Hybrid Event Extraction . . . . .	21
2.2.4 Limitations . . . . .	22
2.3 News Event Exploration . . . . .	22

## TABLE OF CONTENTS

---

2.3.1	Limitations . . . . .	26
2.4	Summary . . . . .	26
<b>3</b>	<b>Named event Extraction from News Headlines</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Proposed Approach . . . . .	32
3.2.1	Headline Dataset . . . . .	33
3.2.2	Named Event Extractor . . . . .	34
3.2.3	Named Event Category Extractor . . . . .	41
3.2.4	Named Event Duration Extractor . . . . .	44
3.3	Experiments . . . . .	48
3.3.1	Quantitative Evaluation . . . . .	48
3.3.2	Qualitative Evaluation . . . . .	56
3.4	Summary . . . . .	62
<b>4</b>	<b>Infobox Mining of Events</b>	<b>67</b>
4.1	Introduction . . . . .	67
4.2	Proposed Approach . . . . .	68
4.2.1	Data Collection . . . . .	69
4.2.2	Key-phrase Extraction . . . . .	69
4.2.3	Key-phrase Enrichment . . . . .	71
4.3	Experiments . . . . .	74
4.3.1	Richness of Key-phrases . . . . .	76
4.3.2	Meaningfulness of Key-phrases . . . . .	79
4.3.3	Correctness of Key-phrases . . . . .	79
4.4	Summary . . . . .	80
<b>5</b>	<b>Event Extraction from Streaming Tweets</b>	<b>81</b>
5.1	Introduction . . . . .	81
5.2	Proposed System Overview . . . . .	83
5.2.1	Tweet Pre-processing . . . . .	83

5.2.2	Tweet Embedding . . . . .	87
5.2.3	Unsupervised Event Detection . . . . .	90
5.3	Experiments . . . . .	94
5.3.1	Data Collection and Statistics . . . . .	94
5.3.2	Experimental Design . . . . .	95
5.3.3	Event Detection . . . . .	96
5.4	Summary . . . . .	100
<b>6</b>	<b>Conclusion and Future Works</b>	<b>101</b>
<b>A</b>	<b>News Recommendation System</b>	<b>103</b>
A.1	Proposed Approach . . . . .	105
A.1.1	Overall Architecture . . . . .	105
A.1.2	Preliminaries . . . . .	106
A.1.3	URL Tagging . . . . .	107
A.1.4	Extract News Category URLs . . . . .	107
A.1.5	Building News Category Network . . . . .	108
A.2	News Source Ranking . . . . .	109
A.2.1	Traffic-based Website Importance . . . . .	109
A.2.2	Social Media-based Popularity . . . . .	109
A.2.3	Category-wise Article Freshness Score . . . . .	110
A.3	Experiments . . . . .	110
A.3.1	Efficiency of News Category URL Extraction . . . . .	111
A.3.2	Analysis of News Categories in News Category Network . . . . .	112
A.3.3	Analysis of News Source Ranking in News Category Network . . . . .	113
A.4	Summary . . . . .	114
	<b>Bibliography</b>	<b>115</b>





# List of Figures

1.1	Pipelining structure for knowledge discovery. . . . .	3
1.2	Overview of the proposed works of this thesis. . . . .	6
2.1	Classification of key-phrase extraction approaches. . . . .	12
2.2	Classification of event extraction approaches. . . . .	14
3.1	Overview of proposed system: Named Event Exploration Engine. . . . .	32
3.2	Overview of named event extractor. . . . .	34
3.3	Path for tokens and its root node. . . . .	39
3.4	Zipf plot for candidate-level categories on log-log scale. . . . .	42
3.5	Month-wise time series of named events. . . . .	47
3.6	Distribution of headlines containing the prominent features: colon, quotes, capitalized words. . . . .	50
3.7	Zipf plot for discovered named events on log-log scale. . . . .	51
3.8	Distribution of named event’s categories. . . . .	53
3.9	Month-wise time series for recurrent named events. . . . .	55
3.10	Month-wise time series for durative named events. . . . .	55
3.11	Number of correct named events out of selected top 10 named events of each method. . . . .	57
3.12	Month-wise precision and recall of discovered named events against the manually annotated dataset. . . . .	58
3.13	F1 Measure of named event’s categories. . . . .	61
4.1	Overview of the proposed $E^3$ System. . . . .	68

## LIST OF FIGURES

---

4.2	Infobox of key-phrases with respect to query “ <i>Bihar Election</i> ” . . . . .	75
4.3	Distribution of extracted key-phrases for $E^3$ , KEA and ToPMine key-phrase extraction techniques. . . . .	78
4.4	Precision and Recall of extracted key-phrases. . . . .	80
5.1	Overview of the proposed method. . . . .	84
5.2	Steps to process raw tweets. . . . .	86
5.3	Vector Space for 8 tweets ( $t_1$ to $t_8$ ) listed in Table 5.1. . . . .	88
5.4	Tweets posted against days of June 2017. . . . .	94
5.5	Clusters of 61, 115 tweets using DBSCAN clustering algorithm against $\epsilon = 0.05$ and $MinPnt = 10$ . . . . .	95
5.6	Clusters of 61, 115 tweets against the different $\epsilon$ values. . . . .	97
5.7	Precision and Number of detected events against the various $\epsilon$ values. . . . .	98
A.1	Overall architecture of the system. . . . .	106
A.2	Example of small news category network. . . . .	108
A.3	Precision and Recall of proposed approach. . . . .	112
A.4	News categories returned for <a href="http://www.thehindu.com">http://www.thehindu.com</a> . . . . .	113

# List of Tables

2.1	Summary of the related work for event extraction. . . . .	23
3.1	Sample of named events. . . . .	31
3.2	Sample headline dataset. . . . .	34
3.3	Seed key words list. . . . .	40
3.4	Sample URLs associated with named event “ICC World Cup 2015”. . . . .	41
3.5	Sample URLs associated with named event “Piku review”. . . . .	42
3.6	Named events and their categories. . . . .	44
3.7	Output of key-phrases extraction step. . . . .	49
3.8	Discovered named events with their support. . . . .	50
3.9	Distribution of discovered named events. . . . .	51
3.10	Sample of discovered named events. . . . .	52
3.11	Sample of named events for which Dmoz taxonomy does not provide a high-level category. . . . .	53
3.12	Category wise named events. . . . .	54
3.13	Sample of discovered named events. . . . .	54
3.14	Named Events. . . . .	60
3.15	Named events with support, categories, popular durations and its type. . . . .	63
4.1	Results of type discovery phase for given query “Bihar Election”. . . . .	73
4.2	Novel and Active Key-phrases associated with query “Bihar Election”. . . . .	74
4.3	Comparison of extracted key-phrases from $E^3$ against Google Trend queries. . . . .	79
5.1	Sample of detected events. . . . .	89

## LIST OF TABLES

---

5.2	Sample of raw tweets. . . . .	93
5.3	Comparison of the self-learning-based max-margin clustering with the other existing clustering techniques against the considered performance metrics. . .	99
A.1	pre-marked strings. . . . .	107
A.2	Example of news source category url. . . . .	109
A.3	Article freshness score of news category URL. . . . .	111
A.4	Example of news source category url. . . . .	112
A.5	Top 5 category links for “sports” category. . . . .	114
A.6	Results using Google News for sport Category. . . . .	114



# Chapter 1

## Introduction

Digital media is a dynamic and emerging source, that generates a huge amount of information on daily basis. Digital media includes social media (e.g., Twitter<sup>1</sup>, Facebook<sup>2</sup>, LinkedIn<sup>3</sup>), news media, television and images (e.g., satellite, natural), etc. Social networking service such as Twitter contains information in the form of tweets, comments, and followers. News media contains information in the form of news headlines, news articles, comments, etc. An image of an object shows its visual perception and there are many online sources available for satellite images (e.g., Google earth, GF-2). Television shows real-time events like *IIFA awards, matches*, etc. According to news media monitoring systems, such as Europe Media Monitor (EMM) [116], Global Data on Events, Location and Tone (GDELT) [58], more than 10K unique news are published on daily basis. According to [80], Twitter generates 50M tweets per day. Till December of 2017, Twitter has on an average 328 million monthly active users<sup>4</sup>, which clearly depicts its versatility. Digital media has become a source of huge amount of unstructured data but the end user can not directly access the desired information from it. Therefore, an efficient and high computing mechanism is required to process and handle the issues of information overload [34, 48] and also provide an automated way to extract information from it.

Existing works on digital media content are centered towards three stages of data pro-

---

<sup>1</sup><https://twitter.com/>

<sup>2</sup><https://www.facebook.com/>

<sup>3</sup><https://www.linkedin.com/home>

<sup>4</sup><https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>

---

cessing: (1) Data extraction, (2) Data management [60, 64, 113] and (3) Knowledge discovery [59, 99, 130] and building knowledge base. Data extraction from news media includes extraction of news headlines, news articles, comments, etc. For Twitter, data extraction includes tweets, followers, comments, commentator information, etc. Several APIs are freely available to crawl the data from digital media. As digital media is a source of huge amount of data, efficient techniques are required to manage it that helps end user to fetch data in real time. Lucene [83] is an indexing technique, that indexes data to make fetching time very fast. Google News<sup>5</sup> collects news articles from more than 4500 news sources and indexes them. Google News provides the facility to search, and sort the articles by date and publication time based on user interest. Storing huge amount of unstructured data into a structured format and processing it later is a very complex task. An alternate solution is to collect important facts from unstructured data and store it in such a way that it helps end users to serve their queries. One example of knowledge discovery from unstructured data is Wikipedia infobox. Wikipedia<sup>6</sup> is a free encyclopedia that covers a topic as an article. Wikipedia is human edited and most of the times wiki page of an event is news article of that event. Infobox content in Wikipedia is structured content that is generated by collecting knowledge from articles and storing it in a specific format.

In recent past, a lot of research work has been done on knowledge discovery from digital media. Discovered knowledge from digital media is useful to generate various type of information such as event detection [8, 9, 66, 69, 70, 77, 98], headline generation [3, 123], story telling [103], building knowledge graphs [66], relation extraction [39, 81, 106, 134, 136], document summarization [79, 110, 111], document classification [27, 28], object recognition [20, 47, 54, 101], image captioning [87, 135], object tracking [45, 121], biometrics analysis [2, 93, 125], image classification [71, 137], activity recognition [12, 13, 112, 126], etc.

In our work, we focus on knowledge discovery from digital media. Figure 1.1 narrates the general pipelining structure used for knowledge discovery from digital media. Once the data is captured, it is needful to pre-process them to deal with noisy content (associated with

---

<sup>5</sup><https://news.google.co.in/>

<sup>6</sup><https://en.wikipedia.org>

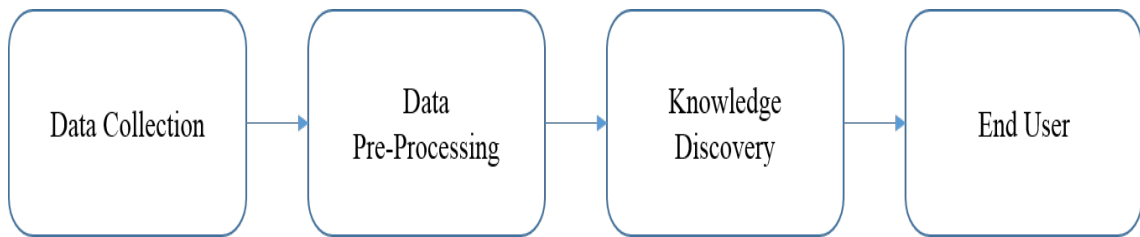


Figure 1.1: Pipelining structure for knowledge discovery.

data). After pre-processing, text mining or machine learning based approaches are used for knowledge discovery. Most of the times, digital media provides wrapped statistics on past, ongoing, and future events with their location and time but some of this information may die over a period of time. An end user is always interested to get information about what has happened in the past, what is happening and what is about to happen in this world irrespective of time limit (past, present, future).

## 1.1 Event Detection and Representations of Events

The problem of Event Detection from unstructured data has been studied extensively under the umbrella of data mining, text mining and image processing communities. The sources of unstructured text for event detection include news articles [46, 89], social media content [10, 70], and search query logs [57]. According to recent works, event can be typified as a sentence [44, 70, 97, 120], as a cluster [89] and as a category [103]. EVIN [66] represents an event as a cluster of news articles and the name of events is obtained on the basis of its semantic class such as *elections*, *championships*, etc. Authors in [105] describe events as what has occurred, who or what was involved with the time interval of the occurrence. It may also include the location of an event. An event in ACE<sup>7</sup> format is represented as a triplet of event mention, event trigger, and event argument. The 2005 ACE evaluation had 8 types of events, with 33 subtypes. Kira *et al.* [97] extract events in the form of triplet  $\langle \text{Set of the object, relation over objects, time interval} \rangle$ . Foley *et al.* [44] describe events as three “W” (*What, When, Where*). Twevent [70] represents the event as a segment of a tweet. HEADY [3] represents an event as a headline. Some authors describe an event as *situations*

<sup>7</sup><http://projects.ldc.upenn.edu/ace>

such as flood, earthquake, etc. Some authors select *actions* as an event such as buy, give, offer, etc. Others describe an event as a change of *states* such as kidnapped, throw, etc.

The researchers of the data mining, text mining and image processing communities follow clustering, classification, information extraction, relation extraction, image captioning and other machine learning approaches for event detection.

## 1.2 Motivation and Research Gaps

By reviewing the literature available on event detection and key-phrase extraction from digital media, we come up with the following research gaps along with the motivation behind selecting the problem statements:

- News media covers all the aspects of noteworthy events and publishes news in the form of news articles and news headlines. Generally, news articles talk about events in detail and news headlines represent the brief introduction of the news articles. We noted that the existing work focuses on discovering events of named event from news contents, but not the named event. For example,  $\langle \text{ICC World Cup 2015, held at, Australia} \rangle$  is an event for ICC World Cup 2015. Moreover, none of the existing work focuses on discovering named events from the news headlines. The specific name of the event helps the end users in exploring the related information.
- News media is publishing ideas, events, and opinions in an increasingly wide range of data formats such as news articles, headlines, videos, tweets, hashtags, and others. The explosion of big news data has sparked the text and data mining research communities to focus on developing systems for news data exploration and analysis. Such available systems provide up-to-date news information in real time, but they overload the user with the large amounts of results. Moreover, there is a need for a system that enables readers to get a broad overview of the news data which is generated in response of a user query (event).
- Twitter is one of the globally used major micro-blogging services. Twitter generates data as tweets, comments, and followers with the limitation of 140 characters. So



tweets consist of short form of words, smilies, emojis, emocations and generally does not follow any grammatical structure. Everything posted by users on Twitter is generally targeted on events happening around them. The existing models extensively use supervised strategies to detect the events from tweets, apart from manually fixing informative keywords for different events.

### 1.3 Problem Statements

The aim of the proposed research is to develop methods that extract the events and its related information from digital media (News data, Twitter data). Figure 1.2 depicts a broad overview of the research performed at different stages of this thesis. After analyzing the different representations of events (based on literature), the following problem statements are addressed:

1. Named event discovery with its type, categories, support, and popular durations from news headlines dataset. News headlines dataset consists of news headlines, headline URLs, and temporal information of headlines.

**Note:** We focus on discovering events with its specific name (named events). Named events are short and meaningful phrases that represent the name of events. We include specific names of elections, disasters, annually celebrated events, reality shows, live concerts, and events related to movies, politics, society, and business as named events. Some examples of named events are *Bihar elections 2015*, *World Chess Championship 2014*, *Auto Expo 2016*, *Star Wars*.

2. Infobox Mining of named events from news media (headlines, keywords, meta description, articles, publication date, and URLs related to an event). Infobox contains person entities, location entities, organization entities, and key-phrases related to given query (named events).
3. Event detection from streaming tweets (text data) using an unsupervised learning approach. We employ a self-learning max-margin clustering based approach which deploys the notion of SVM in an unsupervised setup.

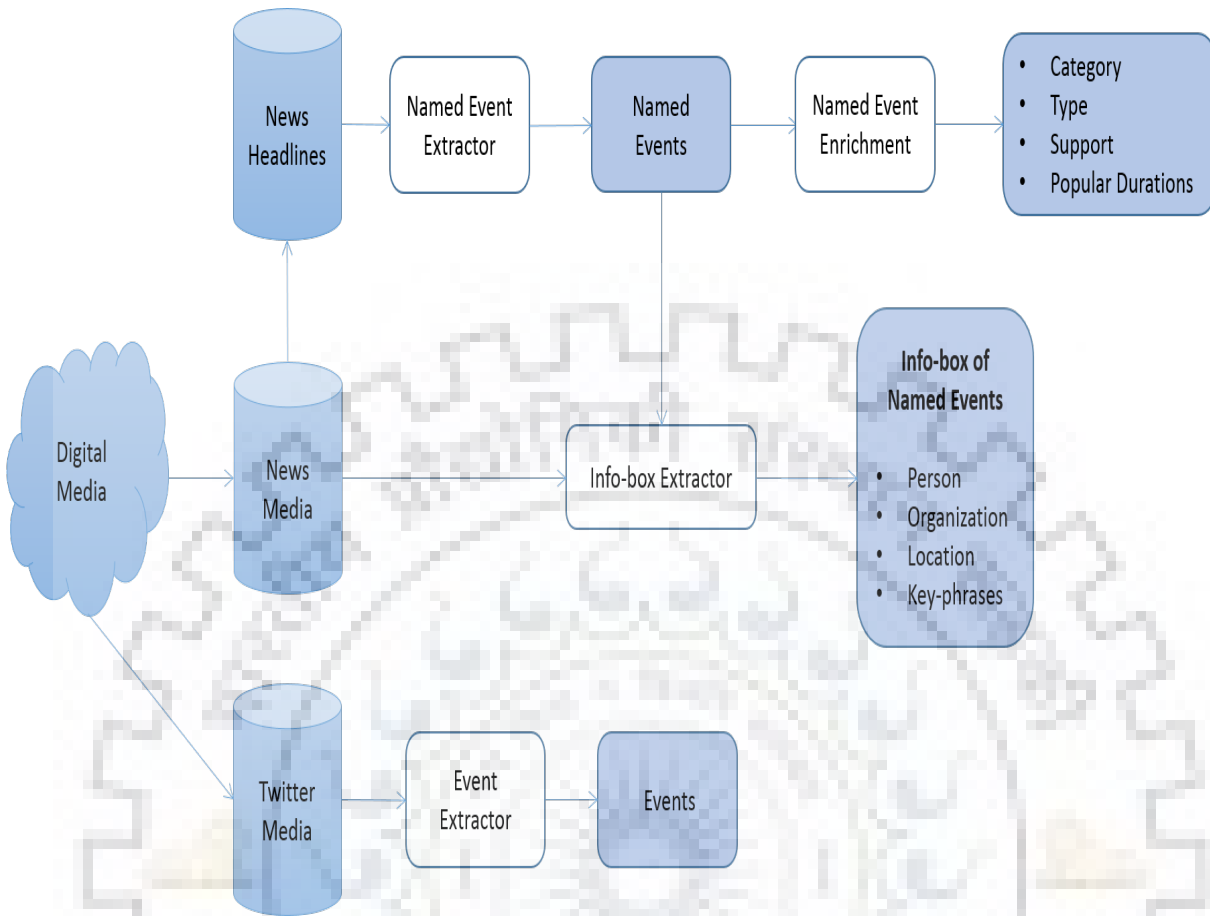


Figure 1.2: Overview of the proposed works of this thesis.

## 1.4 Research Contribution

- Problem Statement 1:** As news headlines are short and contain only essential part of the associated articles, the existing NLP [19, 118] or open IE [39, 134, 136] based methods are not suitable to discover named events from news headlines. Thus, named event discovery from news headline is a challenging problem. To deal with cited issues, we develop a filter-and-refine based technique for named event discovery from news headlines. Filter step uses prominent features of news headlines to extract the key-phrases. Refine step employs a syntactic pattern-based method for generating named events from key-phrases. The syntactic pattern includes day-based, number-based, and seed word-based pattern templates. We introduce a URL information based named event categorization method. We seek to list the type of named events (recur-

rent, durative and normal) using popular durations of named events that are obtained by leveraging the temporal information (publication date) of the news headlines.

Our system generates 75,689 number of named events by analyzing 6.5 million news headlines collected during January 2014 to August 2016 from English news sources. Out of 75,689 named events, 62,950 (82%) are categorized and popular durations are extracted for 73,288 (96.8%) number of named events. For performance assessment, the extracted named events are compared with the help of user study, availability of Google pages, and meta keywords of news. Categories of named events are compared based on user study and manually annotated dataset. We compare extracted popular durations of named events with the Google Trends. Based on performed experiments, our proposed system  $NE^2$  has 68.0% of accuracy for named events, 71.6% for the named events category, and 78.4% for the named events popular duration.

- **Problem Statement 2:** As news media data is growing exponentially day by day, the efficient methods are required to extract the desired information in real time. To deal with the information overload issue, we develop an approach to extract informative and interesting key-phrases using linguistic and syntactical features of the text. The linguistic and syntactical features help in extracting the information from heterogeneous news sources.

The proposed method is tested by various types of the input query ranging from general topics (e.g., Election, ISIS) to specific topics (e.g., Paris Attack, Gravitational Wave) and compared the results with KEA [129], ToPMine [37], and Micro-gram [132]. We found that our method outperforms existing approaches in terms of quality and quantity of key-phrases generated.

- **Problem statement 3:** Since the typical posts on Twitter (also known as a tweet) has a limitation of 140 characters, users prefer short words, smilies, emojis, emocations to complete their messages. In addition, tweets may consist of mixed data, available in the form of text, images, videos, and URLs. To deal with such data, we perform rigorous pre-processing to a large volume of tweet stream before deploying the proposed event extraction paradigm. We seek to represent the pre-processed tweets using a

novel word vector based representation which ensures that semantically related tweets reside densely in the semantic space where standard machine learning algorithms can be applied. To cluster semantically similar tweets, we introduce a self-learning based iterative clustering paradigm using a max-margin clustering concept which can efficiently deal with overlapping clusters. Our method is generic and can be adapted to deal with different event tags without loss of generality.

We evaluate the proposed system and compared it with the popular techniques from the literature using 6.5 million streaming tweets, collected in June 2017. In our experiments, self-learning based max-margin clustering outperforms the techniques of literature (k-means clustering, Density-based spatial clustering of applications with noise (DBSCAN) clustering, Web-scale k-means clustering) in terms of precision, Silhouette score, and Calinski-Harabaz score.

## 1.5 Organization of the Thesis

This thesis is divided among six chapters. Each chapter can be read independently. The content of each chapter is described as follows:

**Chapter 1:** This chapter gives the brief introduction of digital media with its characteristics, what is event extraction, various representation of events, the motivation behind selecting the problem statements, and their research gaps. This chapter also comprises of the technical contribution of our proposed work.

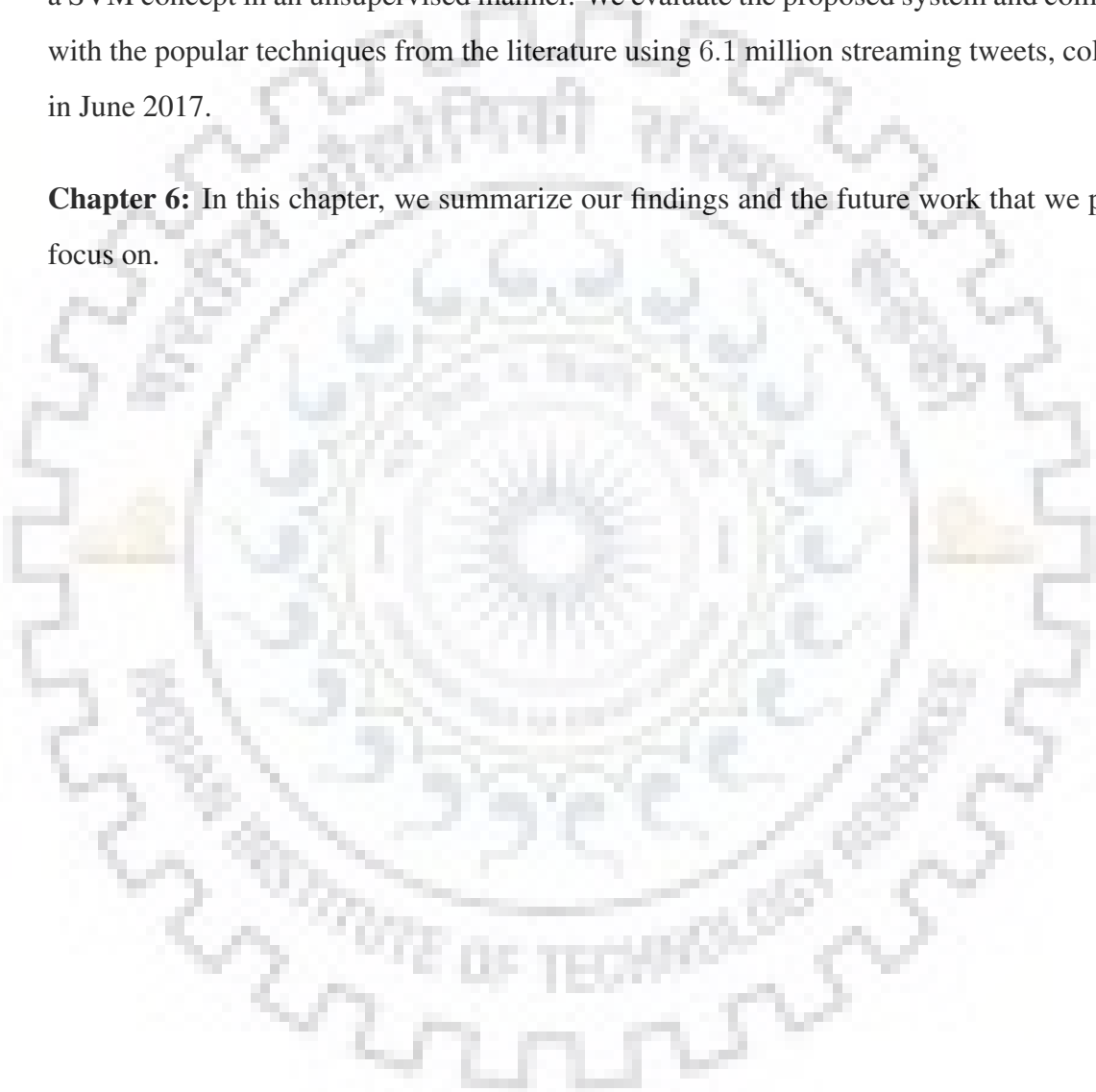
**Chapter 2:** In this chapter, we describe the existing research works related to event extraction, key-phrase extraction, and news exploration systems with their limitations.

**Chapter 3:** This chapter presents a system Named Event Exploration Engine ( $NE^2$ ) that uses the pattern-based method to discover named events using news headlines. Along with the named event, we also discover categories, popular durations, popularity, and its type.

**Chapter 4:** This chapter discusses a system that extracts the key-phrases related to given event. The system generates key-phrases of type person, location, and organization from news media.

**Chapter 5:** In this chapter, we use a self-learning-based max-margin clustering which uses a SVM concept in an unsupervised manner. We evaluate the proposed system and compare it with the popular techniques from the literature using 6.1 million streaming tweets, collected in June 2017.

**Chapter 6:** In this chapter, we summarize our findings and the future work that we plan to focus on.





# Chapter 2

## Literature Review

In this chapter, we concisely study the prior and relevant works that lead to exploring the subjects of this thesis. The literature review starts with key-phrase extraction approaches with their bottlenecks described in Section 2.1. It extends with the existing event extraction approaches and their limitations described in Section 2.2. We continue this chapter with the reviews of existing news exploration engines and their constraints in Section 2.3 and conclude in Section 2.4.

### 2.1 Key-phrase Extraction

Key-phrases are short phrases that lead users to understand and analyze huge amount of data easily and quickly by allowing them to skip trivial content. State-of-the-art techniques for key-phrase extraction are roughly classified into two categories: key-phrase mining [75, 84, 85, 129] and topic learning [37, 43, 73] as shown in Figure 2.1. However, key-phrase mining produces a wide range of informative and important phrases as it covers every single significant point [117].

#### 2.1.1 Key-phrase Mining

Liu *et al.* [75] proposed an unsupervised approach to extract key-phrases from the document based on exemplar terms. Candidate terms are all single terms extracted by removing stop

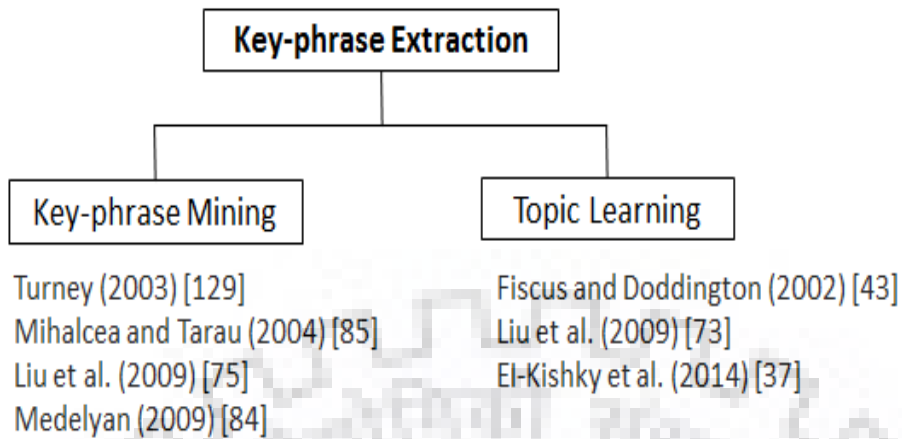


Figure 2.1: Classification of key-phrase extraction approaches.

words from the document. Extracted candidate terms are grouped to form clusters in light of semantic relatedness and the centroid of the clusters are known as exemplar terms. From the generated clusters, the terms close to exemplar terms are selected to extract key-phrases. Mihalcea and Tarau [85] use a semantic graph-based ranking model known as TextRank to extract key-phrases from the natural language documents. KEA [129] applied a supervised Nave Bayes (NB) algorithm to automatically extract the key-phrases from the document. In this, candidate key-phrases are selected by lexical methods and then feature values are calculated for each candidate key-phrase. Afterwards, the machine learning approach is used to anticipate which candidate key-phrase is good key-phrase. The model is built using features like Term Frequency (TF), Inverse Document Frequency (IDF), phrase position and probability of phrase to be a key-phrase in the labeled dataset. This method has been enhanced by multi-purpose automatic topic indexing by using a set of Wikipedia based features and replacing the classifier with bagged decision tree [84].

### 2.1.2 Topic Learning

ToPMine [37] framework extracts arbitrary length topical phrases from textual data corpus by following the frequent pattern mining based approach. It follows two-step procedure: first, phrase mining step transforms each document into a bag of phrases. Afterward, the topic modeling approach is used to combine the semantically similar bag of phrases and la-



tent topics are assigned to phrases. ToPMine focuses on extracting frequent and significant phrases. Liu *et al.* [73] describe a novel data-driven framework known as SegPhrase that extracts quality phrases from text corpus by combining the concept of phrase quality estimation with the Random Forest Classifier (RFC). The quality of phrases is evaluated based on four factors: phrase popularity (frequency in the corpus), informativeness (average IDF), concordance (point-wise mutual information and point-wise KL divergence) and completeness (each phrase should be interpreted as a whole semantic unit in certain context).

### 2.1.3 Limitations

The execution of abovementioned approaches are useful for standard entry content i.e., passage text, however their outcomes do not scale well to extract key-phrases from the news media data as news media follow different writing pattern in contrast to standard data. In this manner, the existing approaches fail to extract interesting and complete information from news media data especially news headlines.

- For instance, an approach mentioned in [75] does not work well for news data as stop word removal may cause to separate the potential key-phrases, such as ‘Make in India’, ‘Fast and Furious 7’. For example, headline “ICC Cricket World Cup 2015: Australian squad” is related to “ICC Cricket World Cup” and stop word removal approach outputs complete headline as a key-phrase.
- Existing POS-tagger are trained for named entity discovery, not for named events discovery. As a result, a large number of false positives key-phrases are generated using existing approach [85]. For example, headline “7 game-changing tech of Cricket World Cup 2015” outputs ‘Cricket World Cup’ as key-phrases and ignore the year. However, year is important for representing a named event.
- Basically, ToPMine focuses on named entities such as person, location, named events and ignores numbers as it uses numbers to detect boundaries. So, we cannot extract number based named events from news headlines dataset.

## 2.2 Event Extraction

Event extraction is an important task of IE and the related field of text mining, data mining, and NLP. The problem of event extraction from unstructured data has been studied extensively under the umbrella of data mining and text mining communities. The sources of unstructured text for event extraction include news articles [46, 89], social media content [10, 70], and search query logs [57] etc. According to recent works, event can be typified as a sentence [44, 70, 97, 120], as a cluster [89] and as a category [103]. The researchers of data mining and text mining community follow clustering, classification, pattern-based methods, and other information extraction approaches for event extraction. So, we divide existing approaches of event extraction among three sub-categories: data-driven, knowledge-driven and hybrid event extraction methods. Figure 2.2 depicts the classification of event extraction approaches based on literature. Data-driven approaches transform data into knowledge with the help of machine learning, linear algebra, statistics, etc. Knowledge-driven approaches extract knowledge by following specific patterns that can be either syntactic or semantic. Whereas, hybrid event extraction approaches fuse data-driven and knowledge-driven approaches.

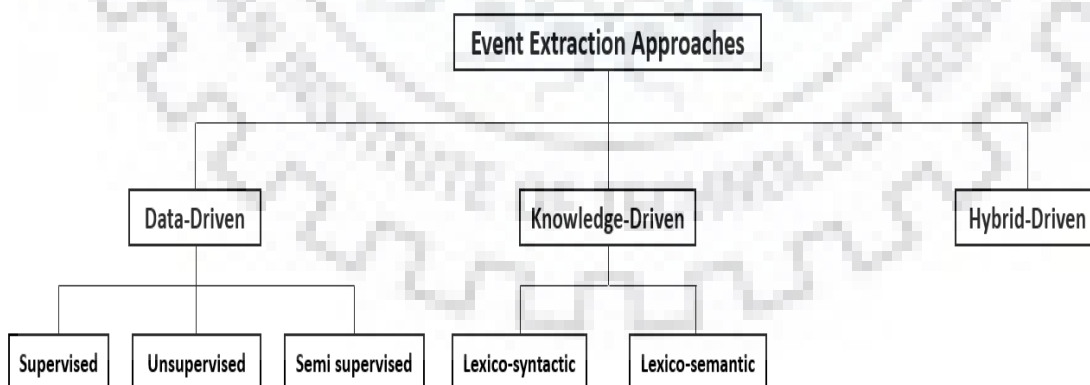


Figure 2.2: Classification of event extraction approaches.

### 2.2.1 Data-driven Event Extraction

Data-driven approaches basically rely on quantitative methods to extract the events. It considers the probability theory methods along with all the quantitative methods such as linear algebra, information theory, and probabilistic modeling. For data-driven event extraction methods, there is a clear difference between the perspectives of supervised and unsupervised learning methods. Supervised learning methods require labeled data or we can say that it requires some expert knowledge. Whereas, unsupervised learning methods are used in absence of expert knowledge or labeled data to find the structure of data and in its exploration. Combination of labeled and unlabeled data can yield extensive improvements in learning accuracy. This combination is known as semi-supervised learning. The semi-supervised learning method is used when a small number of labeled and fairly large number of unlabeled data are available.

There exist a lot of prior work that uses data-driven approaches for event extraction. For example, Liu *et al.* [74] use three-step approach to extract key entities and significant events from web news by combining the concepts of bipartite graph and clustering. Key entities of a particular day are extracted by tracking their records in the specific time window. Bipartite graph is built based on extracted features and news articles related to them with the condition that if the feature term  $t_i$  has relation with article  $d_j$ , then create an edge between  $t_i$  and  $d_j$ . To rank entities and news articles, mutual reinforcement method<sup>1</sup> is employed on the bipartite graph. Finally, clustering of news articles generates significant events. Authors in [94] employ hierarchical clustering method to extract local events with the help of query log entries. EvenTweet [1] detects the localized events by analyzing streaming tweets where the spatiotemporal features of tweets are considered to find the event candidate keywords. Event candidate keywords are subsequently clustered based on spatial similarity to detect localized events. In [138], events are extracted based on a joint distribution of involved named entities, time, location and the keywords of event related tweets. To find out the event related tweets, a lexicon based approach is considered. Kuzey *et al.* [67] build a knowledge base of events extracted from news articles. Similar articles are clustered using

---

<sup>1</sup><http://www.findresources.co.uk/the-syndromes/prader-willi/the-mutual-reinforcement-process>

four features: textual content, publication date of the article, set of entities involved, and semantic type of article. Each cluster is treated as an event, and the headline of a news article that is near to the cluster centroid represents an event. In Event Registry [69], each article is represented as a vector. Articles are clustered based on the entities involved, publication date, title, and body (first paragraph) of articles. Article nearest to the center of the cluster represents an *event*. Category of each event is extracted based on the content of articles and Dmoz taxonomy. Twevent [70] detects the segment based events from tweets. Event segments are identified based on the burstiness of tweet segments where a cluster of segments represents an event. In [131], geospatial events are extracted by analyzing the streaming tweets relating to geospatial locations. A clustering based approach is considered to detect the real-world events based on the burstiness of the geospatial locations of tweets. Authors in [65] proposed a system to extract future events using streaming tweets. The system identifies the time expression and involved entities and then based on these data, tweet clusters are formed to extract future events. Capdevila *et al.* [22] have proposed an event discovery technique named Tweet-SCAN based on the DBSCAN clustering. Tweet-SCAN focuses on four main features of tweets, namely content, time, location and user to group event-related tweets. This method uses a Hierarchical Dirichlet Process (HDP) to model textual content and JensenShannon distance to identify the neighbors in textual dimension. In [96], the authors proposed an unsupervised representation of Adverse Drug Reactions (ADRs) events or drug-disease entity pairs. The ADRs are represented as vectors that link the drug with the disease in their context through a recursive additive model. In [49], authors discuss notable event extraction methods based on several approaches: Incremental-clustering-based approaches, Topic-modeling-based approaches, Term-interestingness-based approaches, etc. D. Sculley in [109] proposed an incremental algorithm that uses mini-batches to cluster the data. The work mentioned in [139] uses a probabilistic model or Bayesian model known as Latent Event Extraction and Visualization (LEEV) for joint event extraction and visualization on Twitter. Inspired by the Latent Event Model (LEM), authors in [138] extracted events in the form of joint distribution of named entities, location, date, and event associated keywords.

Sakaki *et al.* [107] describe a method to detect the real-time events such as “*earthquake*” from tweets by considering twitter users as a sensor. To detect an event, a classifier

is learned based on the word counts, used keywords in tweets and the context of tweets. Authors in [108] describe an ADRs extraction system for Electronic Health Records (EHRs) written in Spanish language. The predictive model, random forest, is trained with manually tagged dataset that uses both semantic and syntactic features of data. The work mentioned in [78] uses stacking approach to extract bio-molecular events. Authors use three-step approach: event trigger extraction, argument extraction with the correct combination of arguments. Support Vector Classification (Linear SVC), Logistic Regression (LR) and Stochastic Gradient Descent (SGD) are used for base-level learning. Whereas, linear SVC and SVM classifiers are used for meta-level learning and trigger extraction respectively.

Ferguson *et al.* [41] present a semi-supervised event extraction method. The labeled training data is automatically generated by taking the advantage of multiple occurrences of mentions related to same event instance in news media. The paraphrases of event mentions are grouped together and simple example of each cluster is selected as a label of the cluster. Kanhabua *et al.* [57] build a two-step classification model to detect event related queries from the query log. First, event candidates are identified using a clustering technique. Further, time series features are used to build classification models with a constraint: queries of a time span generally relate to the same events.

The work mentioned in [91] uses a framework with bidirectional Recurrent Neural Networks (RNN) to extract events in the form of ACM format (event trigger and arguments). The model comprises two RNNs (forward and backward) to find out event triggers and arguments in the single pass and dependencies between them are captured with the help of memory vectors / matrices. Authors in [100] use Abstract Meaning Representations (AMR) to extract event from biomedical data with the hypothesis that event structure is an AMR subgraph. AMR represents a hierarchical structure between entities in the text data. To extract events from AMR automatically, two RNN is used: the first one for labeling the theme and the next one is for labeling the cause of an event. Bjorne [17] introduced a framework, known as Turku Event Extraction System (TEES), which uses a multi-class support vector machine (SVM) to classify the events and dependency parser for collecting features. TEES framework is extended in [15] by replacing SVM classifiers with effective Convolution Neural Network (CNN) method and features are collected based on vector space embedding



concept. Farajidavar *et al.* [40] develop a multi-view learning framework for live annotation and classification of Twitter data, for city event extraction. The proposed framework employs the CNN features with CRF-dictionary-driven Named Entity Recognition (NER) tags to extract city events such as traffic, public transport, weather, social, cultural activities and public safety. Authors in [32] assess three NER methods to extract complex entities (report adverse drug events) exist in the noisy dataset. Two NER methods are based on sequence labeling concepts such as CRF and Bidirectional Long Short Term Memory (BiLSTM) and another one is based on the non-sequence labeling concept. Based on experiments, the authors described that the non-sequence labeling method can best extract continuous multi-word entities. Whereas, CRF using Stanford NER is more successful for discontinuous entities. Chang *et al.* [25] propose a method to extract events from tweets. For each tweet, two Long short-term memory (LSTM) models are used to collect semantic information and to collect rich features from both left to right and right to left direction. The combination of the features is the input to feed forward neural network used for the classification task. In comparison to discrete n-gram features, the neural network can potentially capture non-local dependencies and deep semantic information. On the other hand, [133] propose a method which uses Gaussian Mixture Model (GMM) for extracting bursty words from tweets and time-dependent HDP model to identify the topics from bursty words. Authors in [140] proposed nonparametric Bayesian mixture model to extract events and it uses word embeddings feature to deal with lexical variations of named entities.

Most of the event extraction methods for ACE format rely on supervised learning methods applied on small hand-labeled data. However, it is very expensive to build hand labeled dataset and it is limited in size, which makes supervised methods hard to extract large scale of events for knowledge base population. To deal with the data labeling issue, [26] propose an approach that labels training data for event extraction with the help of world and linguistic knowledge automatically. For world knowledge, Freebase<sup>2</sup> is used as it contains event instance and for linguistic knowledge, Framenet<sup>3</sup> is used as it keeps trigger information. Wikipedia articles are used to label the unstructured text.

---

<sup>2</sup><https://developers.google.com/freebase/>

<sup>3</sup><https://framenet.icsi.berkeley.edu/fndrupal/>

## 2.2.2 Knowledge-driven Event Extraction

In comparison to data-driven approaches, knowledge-driven approaches are based on predefined patterns or templates that define expert domain-specific knowledge. We can roughly divide pattern-based approaches into two categories: lexico-syntactic patterns and lexico-semantic patterns. Lexico-syntactic patterns require lexical representation and syntactic information of data. Whereas, lexico-semantic patterns are more powerful and require lexicon representation with syntactic and semantic information of data.

In the past, numerous works have been done for event extraction using pattern-based methods. Some of them are discussed in this section. REES [5] is a large-scale relation and event extraction system which comprises three pattern-based tagging module, co-reference resolution module, and the template generation module. The system is based on the ontologies of relations and events that cover various domains such as financial, political, military, life related events, business, and their relation. Extracted events comprise of five facts: *who did what to whom when and where?* The work mentioned in [23] focuses on risk assessment with automatically extracting the event type from news articles. The authors applied a chain of processes: after lexical analysis and syntactic parsing of news articles, the output of the process is profiled for semantic role assignment. Further, the concept matching framework is used to find event descriptions from sentences. Authors in [51] present a method to automatically extracting event based commonsense knowledge from web search query by applying lexico-syntactic patterns and semantic role technique. Web search queries are processed to formulate the lexico-syntactic patterns and each extracted sentence is parsed using semantic role labeling concept to extract the commonsense knowledge associated with the event. Nishihara *et al.* [92] propose an approach to obtain personal experience from blogs in terms of event extraction. The extracted events from blogs are defined by three keywords: place, object, and action. For the given query, system downloads related blogs and each blog is processed and separates the sentence containing place keyword. Further, the system extracts object and action keywords from the separated sentences to represent an event.

Jang *et al.* [55] have mentioned a rule-based approach to extract food hazard events from news and social media data. After analyzing official and news articles, authors come up

with event templates consists of 16 fields which are divided among 3 category fields such as company information, product information, and food hazard information. Authors in [114] propose a hybrid method for event extraction in Russian language. The proposed method is based on the combination of manual work such pattern or template construction and automatic corpus-based methods to build a dictionary of vocab required for event extraction. Borsje *et al.* [18] extract financial events from RSS news feeds by applying lexico-semantic patterns and semantic action. By leveraging the concept of existing lexico-syntactic pattern, the authors develop financial ontologies with the consideration that the lexico-semantic patterns can extract much more events in comparison to lexico-syntactic patterns. The patterns used for event extraction are in the triplet form of subject, relation, and object that helps in easy construction and understanding of rules by users. Cohen *et al.* [30] propose an event extraction approach to extract nine basic biological events such as gene expression, protein localization, transcription, binding, regulation, phosphorylation, protein catabolism, positive regulation and negative regulation from a biomedical corpus. By leveraging the semantics and linguistic characteristics of the biomedical domain, authors applied a pattern-based method for concept recognition. BEECON [6] system extracts business events using an ontology-based method from unstructured sources of information and system is able to detect 41 different type of events. This approach uses NLP techniques, manually written rules, and a pattern recognition algorithm to analyze and process business documents. Wouter *et al.* [52] describe a rule-based approach for learning ontology from domain specific (business and politics) data of news articles to extract the events. The method relies on lexico-semantic patterns and compared with the lexico-syntactical pattern-based method to show its superiority and effectiveness. The patterns used in this approach are based on regular expressions that increase the richness of the rules. Dutkiewicz *et al.* [36] describe a rule-based method to extract events from natural language. This method transforms the input text into relation graph using the Stanford dependency parser [33]. Then proposed rules are applied to relation graph to extract events.



### 2.2.3 Hybrid Event Extraction

Aforementioned event extraction approaches have their pros and cons. Data-driven event extraction approaches require manual work to label the data and the amount of manual work is somewhat equivalent to efforts required for knowledge-driven approaches. So, there is a requirement of effective methods that rely on less efforts to extract information. In contempt of disadvantages of data-driven and knowledge-driven approaches of event extraction, the combination of both approaches yields the best results for event extraction task.

Delia Rusu *et al.* [105] describe pattern-based approach to detect events. News articles are parsed using dependency parser to extract events. The same type of extracted events are clustered, with WordNet super-senses [29] and BabelNet senses [90], in such a way that each cluster represents a unique event. Jungermann and Morik [56] mention a hybrid method to extract events from minutes of the plenary session of the German parliament. The method uses a machine learning method Conditional Random Field (CRF) with the lexico-syntactic pattern-based method. Bjrne at el. [16] present an event extraction system that classifies the unlabelled sentences with SVM using a graph-based semantic representation of the sentences. Each sentence is parsed to build a graph that helps in extracting semantic information such as an event trigger and event arguments. The method mentioned in [14] use the combination of machine learning approach and knowledge-based extraction technique to extract security related events from newswire automatically. The crawled news articles are clustered to get rid of duplicacy of data. Further, clustered articles are processed to learn patterns automatically with the help of manually written patterns. Tran *et al.* [128] propose a framework for extracting real-time news events in Vietnamese language. The framework uses the combination of lexico-semantic and maximum entropy machine learning approach to detect events. Maximum Entropy Classifier (MEC) is used to classify the news titles either into event or not-event, based on the training dataset. Then, lexico-semantic patterns are applied to extract events and associated data such as time, place, participants from the detected news titles. The authors in [124] describe a system architecture that extracts the real-time events from newswire data used for global crisis monitoring. News articles are clustered topic wise and each cluster is processed to extract the frame that consists date, location, number of

killed and injured, kidnapped people, actors, and type of event. For pattern learning, a combination of machine learning and knowledge-based methods are used and it requires multiple iterations of machine learning methods followed by manual evaluations. TwiCal [103] extracts calendar of significant events with named entities involved, event phrases, and type of events from streaming tweets. To find events, TwiCal builds an event tagger by manually annotating the tweets with event phrases by exploring the notion of sequence labeling.

### 2.2.4 Limitations

Table 2.1 provides a summary of the state-of-the-art approaches of event extraction discussed in the Sections 2.2.1, 2.2.2, and 2.2.3. Each approach is described with its class, employed methods, event type and their event representation. By analysing the results described in Table 2.1, we come to the fact that the abovementioned data-driven, knowledge-driven and hybrid-driven approaches are suitable to extract events from news media data and represent events in the form of headlines or articles or cluster or set of phrases etc. However, they are not efficient to extract named events (specific name of events) from news headlines, as news headlines follow different writing pattern in comparison to typical textual data. When we talk about event extraction from twitter, the state-of-the-art approaches generally consider keyword burstiness or external thesaurus to identify the event candidates. However, the aforementioned approaches are not sensitive to outliers and overlapping of data.

## 2.3 News Event Exploration

Well known news aggregators such as Yahoo news<sup>4</sup>, Google news<sup>5</sup> mark the data published by online news media and allow users to search keyword-based information. As keyword based document extraction methods retrieve huge amount of data. So, they do not grant speedy insights for news exploration. In order to reduce users efforts required to explore news documents, researchers proposed numerous approaches such as GDEL, EventRegistry [69], EMM, and Searching with Strings, Things and Cats (STICS) [50].

---

<sup>4</sup><https://www.yahoo.com/news/>

<sup>5</sup><https://news.google.com/?hl=en-IN&gl=IN&ceid=IN:en>

Table 2.1: Summary of the related work for event extraction.

Class	Reference	Approach	Event	Event Representation
Data-Driven	Liu et al. (2008) [74]	Bipartite Graph, Clustering	News	Cluster of Articles
Data-Driven	Okamoto and Kikuchi (2009) [94]	Hierarchical Clustering	Local Events (Query Log)	Topic
Data-Driven	Abdelhaq et al. (2013) [1]	Clustering	Local Events (Twitter)	Related Key-words
Data-Driven	Zhou et al. (2015) [138]	Unsupervised Bayesian Model	Twitter	Tweets
Data-Driven	Kuzey et al. (2014) [67]	Clustering	News	News Headline
Data-Driven	Leban et al. (2014) [69]	Clustering	News	Article
Data-Driven	Li et al. (2012) [70]	Clustering	Twitter	Cluster of Segments
Data-Driven	Walther and Kaiser (2013) [131]	Clustering	Twitter	Event Type
Data-Driven	Kunneman and Van den Bosch (2016) [65]	Clustering	Twitter	Calender Date, Named entity, Event phrase, Event type
Data-Driven	Capdevila et al. (2017) [22]	DBSCAN, HDP	Local events (Twitter)	Event Name
Data-Driven	Pérez et al. (2016) [96]	Recursive Additive Model	Health Records	Drug-Disease Entity pairs
Data-Driven	Zhou et al. (2016) [139]	Bayesian Model(LEEV)	Twitter	Named Entities, Date, Location, Related keywords
Data-Driven	Sakaki et al. (2010) [107]	Classifier	Twitter	General Terms as <i>Earthquake</i>
Data-Driven	Santiso et al. (2016) [108]	Predictive model, Random Forest	Health Records	Drug-Disease Entity pairs
Data-Driven	Majumder et al. (2016) [78]	SVC, SVM Classifier	Bio-molecular	Arguments & Event Trigger

Class	Reference	Approach	Event	Event Representation
Data-Driven	Ferguson et al. (2018) [41]	Clustering, JRNN	News	ACE & TAC-KBP
Data-Driven	Kanhabua et al. (2015) [57]	RNN	Biomedical	Arguments & Relation
Data-Driven	Nguyen et al. (2016) [91]	Bi-directional RNN	News	ACE Format
Data-Driven	Rao et al. (2017) [100]	RNN	Biomedical	Event trigger & Arguments
Data-Driven	Björne (2014) [17]	SVM & RLS Classifiers	Biomedical	Entities & Relation
Data-Driven	Farajdavar et al. (2017) [40]	Hierarchical Multi-view Deep Learning	City	Tweets
Data-Driven	Dai et al. (2017) [32]	CRF & BiLSTM	Medical	Drug Diseases pairs
Data-Driven	Chang et al. (2016) [25]	LSTM	Twitter	Tweets
Data-Driven	Wang et al. (2013) [133]	GMM	Twitter	Bursty words as topic
Data-Driven	Zhou et al. (2017) [140]	Bayesian mixture model	Twitter	Cluster of tweets
Knowledge-Driven	Jang et al. (2016) [55]	Lexico-Semantic	Food Hazard	Company, Product & Food hazard Information
Knowledge-Driven	Solovyev and Ivanov (2016) [114]	Lexico-Semantic	Russian Events	ACE format
Knowledge-Driven	Aone and Ramos-Santacruz (2000) [5]	Ontologies based	Financial, Political, Military, Life related, Business & their relation	Who did What to Whom When & Where
Knowledge-Driven	Borsje et al. (2010) [18]	Lexico-Semantic	Financial	Subject, Relation, Object
Knowledge-Driven	Capet et al. (2008) [23]	Lexico-Syntactic & Semantic role labelling	Risk Assessment	Description

<b>Class</b>	<b>Reference</b>	<b>Approach</b>	<b>Event</b>	<b>Event Representation</b>
Knowledge-Driven	Cohen et al. (2009) [30]	Lexico-Semantic	Biological	Basic Biological Events
Knowledge-Driven	Hung et al. (2010) [51]	Lexico-Syntactic & Semantic role labeling	Commonsense Knowledge	Subject, relation, Object
Knowledge-Driven	Nishihara et al. (2009) [92]	Lexico-Syntactic	Personal Experience	Place, Object, Action
Knowledge-Driven	Arendarenko and Kakkonen (2012) [6]	Lexico-Semantic	Business	In the from of Activities
Knowledge-Driven	IJntema et al. (2012) [52]	Lexico-Semantic	Business & Politics	Subject, Relation, Object
Knowledge-Driven	Dutkiewicz et al. (2014) [36]	Lexico-Semantic	Natural Text	Template form
Hybrid	Rusu et al. (2014) [105]	Lexico-Syntactic, Clustering	News	Cluster of Articles
Hybrid	Jungermann and Morik (2008) [56]	Lexico-Syntactic, CRF	Parliamentary Session	Question & answering
Hybrid	Björne et al. (2010) [16]	Lexico-semantic, SVM	Biomedical	ACE format
Hybrid	Best et al. (2008) [14]	Lexico-syntactic, Clustering	Security in News	ACE format
Hybrid	Tran et al. (2012) [128]	Lexico-semantic, MEC	Vietnamese News	News Title
Hybrid	Tanev et al. (2008) [124]	Lexico-syntactic, Clustering	Global Crisis	Event type, Related phrases
Hybrid	Ritter et al. (2012) [103]	Manual annotation with Sequence Labelling	Twitter	Event type, Related phrases

GDELT is another CAMEO-coded dataset consists more than 200 million geolocated events with worldwide scope for 1979 to the present. The information depends on news reports from an assortment of worldwide news sources coded utilizing the Tabari framework for events and incorporates extra software for area and tone. GDELT stores events as triplet of  $Actor_1$ , relation, and  $Actor_2$ . Leban *et al.* [69] build a system Event Registry to extract the events from news articles. A vital usefulness of Event Registry is to cluster the news articles related to an event, regardless of in which language the articles are composed. Event title and a short content are dictated by identifying the article nearest to the centroid of the event cluster and utilizing its title and first passage. Articles are clustered based on the vector space model of the article title, body and named entities each article consists and entities are assigned substantially higher weights than ordinary words. STICS [50] extracts documents based on given keywords, categories and entities. By following the concept of named-element disambiguation, STICS extracts the documents from the web which consists the given query's entities and significant entities.

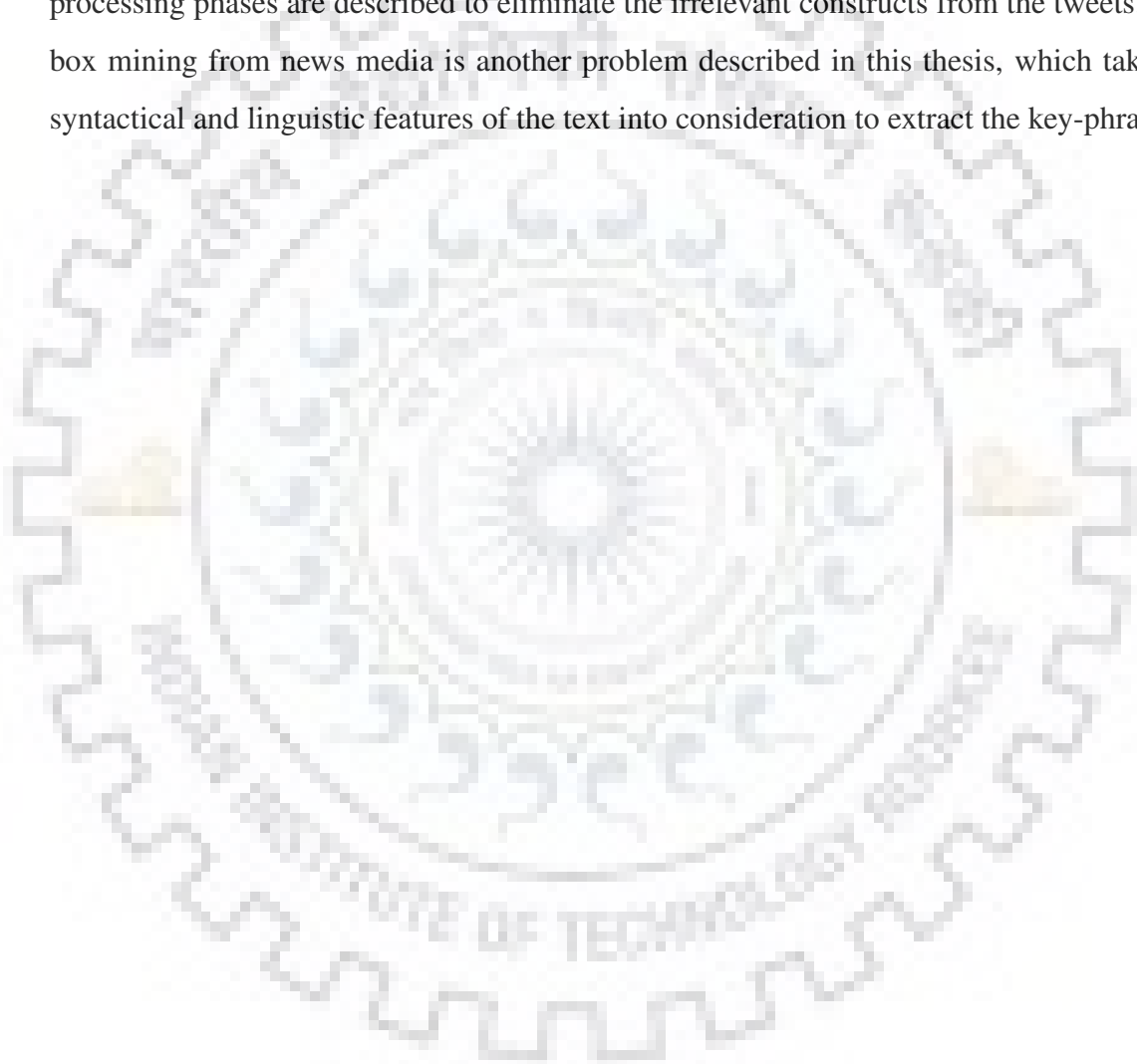
### 2.3.1 Limitations

Inspite of the fact that, the aforementioned news event exploration systems give latest and relevant information immediately, yet they over-burden the user with the substantial amount of results. For example, given input keyword “*FIFA World Cup*” event-driven EventRegistry system come up with more than 10K news headlines, and content-driven STICS provide more than 10 lacs news articles, anyway not each of them might be helpful. Existing systems do not provide an extensive scope of entities connected to any event. For the event-based query, the systems return just popular entities, not concentrating on the connections those entities have with the event.

## 2.4 Summary

In this chapter, we described the state-of-the-art approaches of key-phrase extraction, event extraction, and news exploration engines with their limitations regarding crawled news media

and Twitter data. In order to overcome the aforementioned limitations, we proposed some solutions which are described in the subsequent chapters. Named event extraction is the first problem tackled in this thesis, which leverages the syntactical patterns, URL information, and temporal information of news headlines to extract its associated data. The next problem addressed in this thesis uses the concept of SVM in an unsupervised manner, which helps in dealing with the data overlap issues and the presence of outliers. The extensive tweet processing phases are described to eliminate the irrelevant constructs from the tweets. Info-box mining from news media is another problem described in this thesis, which takes the syntactical and linguistic features of the text into consideration to extract the key-phrases.







# Chapter 3

## Named event Extraction from News

### Headlines

This chapter deals with extracting events with its specific names and associated information from the news content. Section 3.1 describes the problems with the existing event extraction approaches and the brief overview of the proposed solution. In Section 3.2, we discuss the proposed system Named Event Exploration Engine ( $NE^2$ ) with the algorithms used to extract named events, categories, popular durations, type, and support from news headline dataset. Section 3.3 discusses about quantitative and qualitative evaluation of proposed system, and we conclude proposed  $NE^2$  system in Section 3.4.

### 3.1 Introduction

As time passes, the future becomes present and present becomes past. Along with time, many events happen around us and noteworthy events are published through news media and social media. These events are frequently reported by online news media and stored in news archives as unstructured data in the form of news headlines and articles. Usually, news headlines and its associated articles talk about events along with mention of additional

---

The content of this chapter is presented in paper:  
Swati Gupta and Dhaval Patel, " $NE^2$  : named event extraction engine," Knowledge and Information Systems, May 2018.

information such as named entity, location, time, and meta events. According to many news media surveillance systems, such as EMM [116], IMM [82], GDELT [58], large number of distinct news headlines are published daily. In summary, Online news media is a dynamic and emerging source that provides a huge amount of news data for discovering real-time events. Some news events are being updated on Wikipedia by volunteers.

However automatically extracting news events is a challenging problem. Many techniques such as EVIN [66], HEADY [3] have been proposed for extracting events from text documents. Events are represented in many forms by existing event extraction approaches. EVIN [66] represents an event as a cluster of news articles and the name of events is obtained on the basis of its semantic class such as *elections*, *championships*, etc. and HEADY [3] represent an event as a headline. Whereas, Feng *et al.* [11] extracts event in ACE<sup>1</sup> format where an event is represented as a triplet of event mention, event trigger, and event argument.

Incited by the aforementioned discussions, we focus on discovering events with its specific name (named events). As mentioned in Chapter 1, Named events are short length meaningful phrases that represent the specific name of events. In this work, we are extracting specific names of elections, disasters, annually celebrated events, reality shows, live concerts, and events related to movies, politics, society, and business as named events.

In this chapter, we develop a system  $NE^2$  that extracts named events with its *category*, *popular durations*, *support* and *type* from news headlines and its associated information (headline URL, publication date). Popular duration of named events is the time interval or duration in which named event is much popular in news media. Support is an integer value that represents the popularity of the named event. Extracted named events are listed among three types such as recurrent, durative and normal. Recurrent named events recur after a particular time interval and named events which are popular for at-least four months consecutively are known as durative named events. Whereas, named events which are neither recurrent nor durative are listed as normal named events. Table 3.1 shows the sample output of our proposed  $NE^2$  system with three named events such as *Rio Olympics 2016*, *Friendship day*, and *JNU Row*. A normal type named event *Rio Olympics 2016* is extracted using a pattern-based method with candidate-level categories *Sports*, *Football* and high-level cat-

---

<sup>1</sup><http://projects ldc.upenn.edu/ace>

egory *Sports*. This was popular in news media during June to August in 2016 with support 1767. Whereas, the named event *Friendship Day* is a recurrent event which was popular from August to September in 2014, 2015 and 2016.

Table 3.1: Sample of named events.

Named Events	Support	Category-Level		Popular Duration		Type
		Candidate	High	From	To	
<b>Rio Olympics 2016</b>	1767	Sports Football	Sports	2016-June	2016-Aug	Normal
<b>Friendship Day</b>	234	Entertain Celebrity	Arts	2014-Aug 2015-Aug 2016-Aug	2014-Sep 2015-Sep 2016-Sep	Recurrent
<b>JNU Row</b>	1620	Delhi India	Regional	2016-Feb	2016-Mar	Normal

There are many reasons for working with news headlines in spite of news articles. News (articles and headlines) are generated in large quantity, but news headlines can be obtained easily using RSS feed or very simple algorithms [83]. Moreover, news headlines are short in length in comparison to news articles and represent brief introduction or key-idea of its associated article [82]. In summary, it is easy to collect news headlines that contains the important part of the news story. In order to obtain named event category, we leverage the news headline URL information. The motivation behind it is that most of the news sources worldwide use the “*category*” words in their URLs to maintain the web directory structure of news content on their websites. To extract popular duration, we leverage temporal information (publication date) of news headlines. Popular duration of named event helps in listing the type of named events such as recurrent, durative and normal.

A filter-and-refine approach is proposed to extract named events from news headlines. The filter step is also known as key-phrase extraction and refine step is known as named event discovery step. The *filter* step uses prominent features (presence of quotes, colons, and capitalization) of news headlines to extract the key-phrases that are likely to be the results. The *refine* step is to further identify the actual named events from the set of candidate key-

phrases. We use a pattern-based approach to refine named events. Once named events are discovered, we enrich named events with its categories, popular durations, support, and type. English dictionary terms are used to infer candidate-level categories and URLs to infer high-level Dmoz<sup>2</sup> categories. We also develop a technique to infer popular durations of named events using temporal information (i.e., publication date) of news headlines and popular duration helps in finding the type of named event.

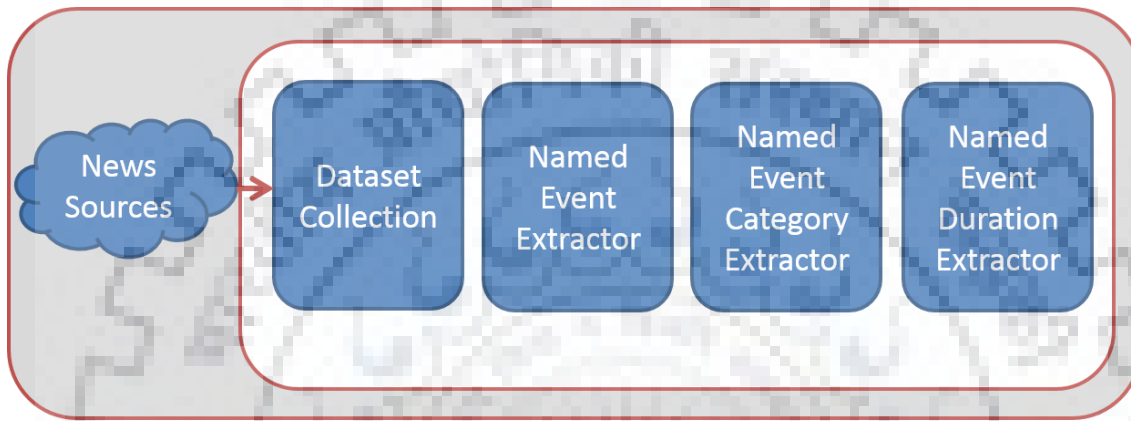


Figure 3.1: Overview of proposed system: Named Event Exploration Engine.

## 3.2 Proposed Approach

Figure 3.1 shows an overview of the proposed approach for named event discovery with its category and temporal information. An input to the  $NE^2$  system is headline dataset. The proposed system consists of three phases: (a) Named Event Extractor - responsible for key-phrase-based named event discovery, (b) Named Event Category Extractor - categorizes discovered named events, and (c) Named Event Duration Extractor - responsible for extracting temporal durations of discovered named events. The output of the system is a named event knowledge base, consists of a set of named events with their categories, and popular durations. The whole procedure is illustrated in Algorithm 1. For a headline  $h$ , key-phrases are extracted (lines 4-5). The details on this extraction are described in Section 3.2.2A. Then, each key-phrase is processed to identify named events (lines 8-9) and the detailed descrip-

<sup>2</sup><https://www.dmoz.org>

tion is explained in Section 3.2.2B. After discovering named events, each named event is enriched with its category (line 12) and popular durations (line 13). Section 3.2.3 and Section 3.2.4 shows the detailed description to extract categories and popular duration of named events respectively.

---

**Algorithm 1:** Named Event Extraction System.
 

---

**Input** :  $\langle H, U, T \rangle$   
 Where  $H$  : Headlines  
 $U$  : URLs  
 $T$  : Headline Date

**Output:**  $\langle E, Ca, Dp \rangle$   
 Where  $E$  : Named Events  
 $Ca$  : Categories  
 $Dp$  : Popular Duration

```

1  $K \leftarrow \phi$ 
2  $E \leftarrow \phi$ 
3 foreach  $h \in H$  do
4    $k_p \leftarrow \text{KeyphraseExtractor}(h)$ 
5    $K \leftarrow K \cup k_p$ 
6 end
7 foreach  $k \in K$  do
8    $e \leftarrow \text{NamedEventExtractor}(k)$ 
9    $E \leftarrow E \cup e$ 
10 end
11 foreach  $e \in E$  do
12    $Ca_e \leftarrow \text{CategoryExtractor}(e, H, U)$ 
13    $Dp_e \leftarrow \text{DurationExtractor}(e, H, T)$ 
14 end

```

---

### 3.2.1 Headline Dataset

The headlines are stored in a headline dataset as a tuple, where each tuple consists of a headline, URL, and temporal information (the time when news headline has appeared in media). The detailed description of headline dataset is given in Section 3.3.1A. Table 3.2 lists six sample headlines and their temporal information. The bold faced text will be discussed in Section 3.2.2.

Table 3.2: Sample headline dataset.

News Headline	Temporal Information
ICC Cricket World Cup 2015: <b>Australian squad</b>	2015-01-13
7 game-changing tech of <b>Cricket World Cup 2015</b>	2015-03-28
<b>Vyapam Scam</b> : twists and turns	2015-07-05
<b>Is Arjun Kapoor</b> hosting ' <b>IIFA Awards 2015</b> ' ?	2015-04-21
<b>Maharashtra Budget</b> : Rs 174 crore for Pune Metro	2015-03-19
<b>Ashish Sharma</b> wins ' <b>Jhalak Dikhla Jaa 7</b> '	2014-09-21

### 3.2.2 Named Event Extractor

In this module, we will explain the process of discovering named events from news headlines. Figure 3.2 depicts the overview of named event extractor. For each input news headline, named event extractor performs the following two steps:

1. **Filter Step.** Extract key-phrases from headline using three prominent features: Colon, Quotes, and Capitalization. (Section: 3.2.2A)
2. **Refine Step.** For each extracted key-phrase, named events are discovered using syntactic patterns. The syntactic patterns are classified into three classes: number-based, seed word-based, and day-based patterns (Section: 3.2.2B).

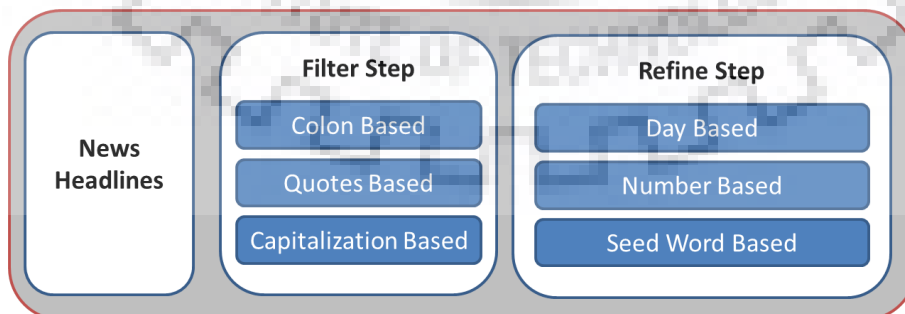


Figure 3.2: Overview of named event extractor.

## A Filter Step: Key-phrase Extraction

Key-phrases are phrases made up of multiple words, obtained from the important part of news headlines which help in discovering the named events. Based on the syntactical structure of news headlines, the important part of the news headline is emphasized by using colons, quotes, and capitalization [136]. We use these observations to extract the key-phrases from news headlines. Given a news headline  $h$ , our key-phrase extraction algorithm (Algorithm 2) works as follows: Colon-based key-phrases are extracted in line (4-10). If headline  $h$  does not contain the colon, capitalization-based key-phrases are extracted (line 12). If headline  $h$  contains quotes, quotes-based key-phrases are extracted (line 14-15). The detailed description of above mentioned key-phrase extraction is described as follows:

---

### Algorithm 2: KeyphraseExtractor( $h$ )

---

**Input** :  $h$  : Headline  
**Output**:  $K$  : Set of key-phrases

- 1  $C_h \leftarrow$  Contiguous Capitalized Words in  $h$
- 2  $Q_h \leftarrow$  Text between Quotes in  $h$
- 3  $K \leftarrow \phi$
- 4 **if** {“:”}  $\in h$  **then**
- 5  $\langle l_1, l_2 \rangle \leftarrow$  Spilt  $h$  by “:”
- 6 **if**  $len(l_1) < len(l_2)$  **then**
- 7  $K \leftarrow K \cup l_1$
- 8 **else**
- 9  $K \leftarrow K \cup l_2$
- 10 **end**
- 11 **else**
- 12  $K \leftarrow K \cup C_h$
- 13 **end**
- 14 **if** {“'”}  $\in h$  **then**
- 15  $K \leftarrow K \cup Q_h$
- 16 **end**
- 17 **return**  $K$

---

- **Colon-based Key-phrase Extraction.** If headline  $h$  contains a single colon ‘:’, then either (left and right) part of colon in news headline  $h$  results in a key-phrase. In this work, we select the part with shorter length (based on the number of tokens) because it reduces the post-processing burden, required to remove stop words from key-phrases.



We have observed that on an average the shorter length part contains 2 to 4 tokens and the other part contains 8 to 9 tokens. So, there is less probability of stop words in a shorter part. Based on collected headline dataset, we have come up with one more observation that the short length part of headline depicts the named event and long part shows associated description or information. Short length part can be either left side or right side of colon.

- **Quotes-based Key-phrase Extraction.** If headline  $h$  contains quotes, then the part of headline  $h$  that is contained between quotes results in a key-phrase.
- **Capitalization-based Key-phrase Extraction.** If headline  $h$  does not contain a colon ‘:’, then the contiguous capitalized words in headline  $h$  results in a key-phrase. Note that, we have ignored the headlines if all of its characters are in upper case.

Highlighted phrases in Table 3.2 represent the key-phrases, generated using the aforementioned key-phrase extraction techniques. For the first news headline in Table 3.2, our technique generates one key-phrase namely “Australian squad” using colon feature. Similarly, we obtain “Cricket World Cup 2015” from the second headline using the capitalization feature.

#### **B Refine Step: Named Event Discovery**

Next phase is to discover named events from extracted key-phrases. As we discussed, the named events are either key-phrases or subpart of key-phrases. Motivated by the recent pattern-based event extraction method [105], in which events are characterized by the presence of the event trigger word, we propose a syntactic pattern-based approach to discover named events from key-phrases. In this work, the event trigger words are subdivided among three categories: (a) Number-based, (b) Seed word-based, and (c) Day-based named event discovery.

The whole procedure of named event discovery from key-phrases is illustrated in Algorithm 3. Formally, let  $P$  is a set of prepositions and  $t_1$  denotes the set of event triggers : “year”, “ number”, “day”, “set of seed words” (line 3). Let  $t_2$  denotes an event trigger “year”



**Algorithm 3:** NamedEventExtractor( $k$ )

---

**Input** :  $k$  : Key-phrase  
**Output:**  $e$  : Named event

```

1  $e \leftarrow \phi$ 
2  $P \leftarrow$  Set of preposition
3  $t_1 \leftarrow \{“year”, “Number”, “Day”, “SeedWords”\}$ 
4  $t_2 \leftarrow \{“year”\}$ 
5 if  $\{P\} \in k$  then
6   | Split  $k$  with  $P$ 
7 end
8  $k_l \leftarrow$  last token of  $k$ 
9  $k_f \leftarrow$  first token of  $k$ 
10 foreach  $t \in t_1$  do
11   | if  $(t == k_l)$  then
12     |  $e \leftarrow k$ 
13   | end
14 end
15 if  $(t_2 == k_f)$  then
16   |  $e \leftarrow k$ 
17 end
18 return  $e$ 

```

---

(line 4). If key-phrase has a preposition, key-phrase is split with the preposition (lines 5-7).  $k_l$  represents last token of key-phrase  $k$  (line 8), and  $k_f$  represents first token of key-phrase  $k$  (line 9). If  $k_l$  represents one of the event trigger mentioned in  $t_1$ , key-phrase is considered as named event (lines 10-13). If  $k_r$  represents event trigger mentioned in  $t_2$ , key-phrase is considered as named event (lines 15-17). The detailed description of pattern-based detection is as follows:

**(a) Number-based Named Event Detection.** In this category, we use year indicator (YYYY) and series indicator (number) as an event trigger. For the series indicator, we consider either one digit or two digit numbers. In particular, key-phrases with following patterns are extracted as the named event. First two patterns are derived using year indicator, and the last one is derived from number indicator:

**⟨Phrase, YYYY⟩**

**⟨YYYY, Phrase⟩**

**⟨Phrase, Number⟩**

By applying aforementioned patterns, the discovered candidate named events are like ICC World Cup 2015, 2014 Commonwealth Games, World Cup 2015, 15 January 2015, etc. We observed that sometimes extracted candidate named events represent the date. Thus, we apply post-processing phase to handle such cases: identify time expressions and remove them.

HeidelTime [119] is a time expression tagger that tags explicit and implicit time expressions. However, HeidelTime does not detect the date with patterns  $\langle YYYY MM DD \rangle$  and  $\langle YYYY DD MM \rangle$ . So, we implemented our own method to detect date expressions by considering all possible combinations to represent date:  $\langle YYYY MM DD \rangle$ ,  $\langle YYYY DD MM \rangle$ ,  $\langle MM DD YYYY \rangle$  and  $\langle DD MM YYYY \rangle$ . A list of prepositions is prepared to remove prepositions from the discovered named event. We select 72 English words as prepositions such as *at, in, on, under, etc.*

**(b) Seed word-based Named Event Detection.** In our context, seed keywords are common words that represent either *event* or *communication* class according to WordNet synset taxonomy. In this category, we use seed keywords as an event trigger. In particular, key-phrases with  $\langle \mathbf{Phrase}, \mathbf{Seed\ keyword} \rangle$  pattern are extracted as named events. Using seed words as the event trigger, the following type of named events are discovered: *Salman Khan Hit-and-Run Case, Ishrat Jahan Case, CNBC Survey, Satyam Scam, Bihar Polls, Piku Review, etc.* Now we explain the process of preparing a seed word list.

S. N. Ghoreishi *et al.* in [46] manually labeled 73 words as the seed word for identifying the recurrent events. However, we are using an automatic approach to identify seed words. To obtain seed words, we tokenized all generated key-phrases and obtained last tokens. Given a token, we find all possible paths between token and its root node (“*entity*”)<sup>3</sup> using direct hypernym synset in WordNet. If at least one path contains *event* or *communication* class, then the given token is included in the seed word list. According to WordNet, *event* is something that happens at a given place and time and *communication* is the activity of conveying information. Figure 3.3 shows the paths for token *Case, Review, Probe, and Treasure*. The *event* and *communication* classes are underlined. Figure 3.3(a)

---

<sup>3</sup><https://wordnet.princeton.edu/>

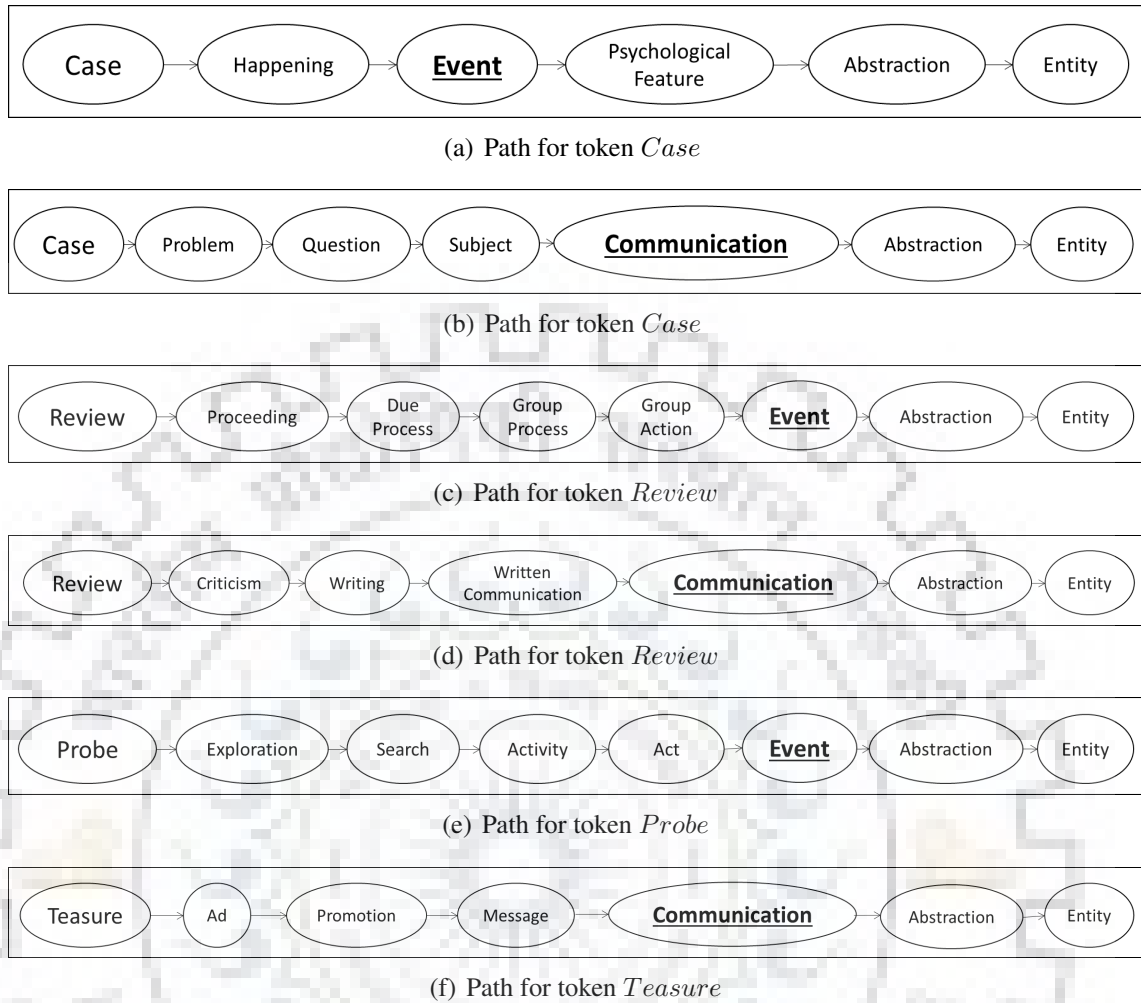


Figure 3.3: Path for tokens and its root node.

and Figure 3.3(b) show the path between token *case* and its root node *entity*. Figure 3.3(c) and Figure 3.3(d) show the path for token *Review*. Figure 3.3(e) and Figure 3.3(f) show paths for tokens *Probe* and *Teasure* respectively. Note that, there are some tokens that belong to *event* as well as *communication* class. In this way, we identify 298 words as seed words to detect named events. Table 3.3 lists 298 seed key-words. These words include review, preview, trailer, scam, issue, etc.

**(c) Day-based Named Event Detection.** Based on collected data, we observe that some recurrent named events are associated with word *Day* such as *Father’s Day*, *Mother’s Day*, *Thanks Giving Day*, *Independence Day*, *Republic Day*, *Valentine’s Day*, etc. In this category, we use a word “Day” as an event trigger. In particular, key-phrases with the following

### 3.2 Proposed Approach

Table 3.3: Seed key words list.

Review	Case	Row	Report	Polls	Preview	Scam	Crash
Issue	List	Probe	Recap	Survey	Blog	Visit	Effect
Study	View	Alert	Debate	Wars	Updates	Update	Killing
Impact	Challenge	Leak	Picks	Remark	Launch	Drive	Arrested
Words	Numbers	Forecast	Scheme	Notes	Cases	Push	Posts
Buzz	Sources	Deals	Schedule	Ties	Matters	Reforms	Tip
Ministers	Lines	Premiere	Chair	Source	Draw	Massacre	Racket
Charges	Dispute	Projects	Clashes	Reports	Features	Forward	Debut
Collapse	Contest	Factor	Ruling	Bash	Comment	Brawl	Account
Address	Stir	Released	Peek	Menace	Links	Works	Clip
Breach	Tussle	Bust	Quiz	Bonds	Parade	Mistake	Comments
Answers	Seconds	Profile	Surge	Link	Heist	Corner	Sight
Plunge	Rout	Haul	Schemes	Spat	Shift	Reviews	Backlash
Sides	Rumors	Mix	Punch	Leaks	Trap	Upgrade	Ambush
Dips	Fate	Pace	Landing	Emails	Rapes	Wounded	Patrol
Pattern	Ventures	Pounds	Reshuffle	Spirits	Crunch	Glance	Quarters
Rampage	Tours	Mistakes	Freeze	Stalemate	Programme	Counsel	Charts
Sells	Revolt	Balls	Spree	Splash	Package	Tag	Scams
Raids	Thrills	Mouth	Contracts	Swap	Wonders	Scoop	Answered
Expose	Replay	Measure	Shower	Burns	Drill	Served	Sought
Retreat	Clue	Spotting	Channels	Clicks	Licences	Fray	Puzzle
Concerts	Crops	Clues	Ruins	Function	Laughs	Farce	Blunder
Accord	Graves	Tumble	Hanged	Prospects	Sacked	Remedies	Certificate
Studies	Exposed	Tango	Busted	Shorts	Booking	Stunt	Pinch
Gamble	Romances	Position	Treats	Visas	Headlines	Pieces	Lying
Crashes	Briefing	Yields	Flap	Controls	Envy	Advances	Bach
Rant	Gesture	Vows	Scrapped	Sacking	Invited	Lapse	Impress
Previews	Vault	Ordered	Essay	Pranks	Cons	Mendes	Aides
Streak	Masses	Quarantine	Encounters	Squeeze	Gazing	Courses	Dissent
Charm	Tools	Harvest	Hunted	Bravo	Landed	Levy	Decree
Shuffle	Pout	Tunes	Offering	Cheated	Query	Contrasts	Positions
Chances	Hijacked	Shadows	Troll	Blockade	Chant	Checks	Styles
Slides	Covered	Transferred	Peril	Careers	Struggles	Recaps	Chronicles
Curves	Escapes	Trace	Deposits	Echo	Boxes	Curve	Dots
Reached	Disputes	Reviewed	Spoofs	Smiling	Interviews	Contrast	Estimates
Buss	Rubbish	Surveys	Flops	Reported	Fashions	Berth	Chatter
Matches	Favor	Outing	Honours	Prowl	Shelters	Taxis	Bursts
Surrenders	Fatigue						

pattern are extracted as named event: **(Phrase, Day)**. Key-phrases containing any generic terms (e.g. some day, every day, etc.) are ignored.

### 3.2.3 Named Event Category Extractor

The proposed system  $NE^2$  categorizes named events into one of the high-level categories of Dmoz taxonomy. Dmoz<sup>4</sup> is the most widely distributed database of Web content classified by humans. We have observed that the distribution of category *Kids and Teens, Health, Home, Shopping and Recreation* is almost negligible and category *World* covers almost all named events for our news headlines dataset. So, in this work, only 10 high-level Dmoz categories are selected: *Arts, Business, Computers, Games, Society, News, References, Regional, Science, and Sports*. Providing a high-level category to named events help users in exploring a large number of named events. However, the named event’s category discovery is challenging as named events are short phrases and do not contain sufficient information that can be used to infer its category. One can argue that news headlines can be used to obtain the category of named events. However, news headlines mostly contain named events and real-world entities, and this information is usually not present in Dmoz taxonomy.

Table 3.4: Sample URLs associated with named event “ICC World Cup 2015”.

<p>...//zeenews.india.com/<b>sports/cricket</b>/icc-world...</p> <p>...//www.financialexpress.com/<b>article/sports</b>/icc-...</p> <p>...//www.thehindu.com/<b>sport/cricket</b>/icc-cricket-...</p> <p>...//timesofindia.indiatimes.com/<b>sports</b>/icc-world-...</p> <p>...//indianexpress.com/<b>article/sports/cricket</b>/...</p>
---

We observe that news URL contains English dictionary words that can be leveraged to infer the categories of named events and using these categories, we can build a recommendation system that recommends category URLs to find related articles (see Appendix A). For example, Table 3.4 shows some sample URLs of news headlines that are used to extract the named event “ICC World Cup 2015”. We have highlighted the dictionary words. These words include sports, cricket, and article. Using these candidate-level categories, we can infer that named event “ICC World Cup 2015” is mapped to high-level Dmoz category “Sports”. Similarly, Table 3.5 shows 5 sample URLs associated with named event “*Piku review*” and categorized under “*Art*” category.

<sup>4</sup><https://www.dmoz.org>

Table 3.5: Sample URLs associated with named event “Piku review”.

---

...//indianexpress.com/ <b>article/entertainment/movie</b> ...
...//wap.business-standard.com/ <b>article/specials/film-revi</b> ...
.../ <b>entertainment/movies/piku</b> ....
...//indianexpress.com/ <b>article/entertainment/movie</b> .....
...//hindustantimes.com/ <b>movie/movie-reviews</b> .....

---

Figure 3.4 shows the Zipf plot (rank-frequency plot) for all such candidate-level categories which are extracted from the named event’s headline URLs. This Zipf plot on log-log scale is plotting the distribution of highly skewed words, where *news*, *business*, *articles* are frequent and frequency of word *hairstyle* is near to one.

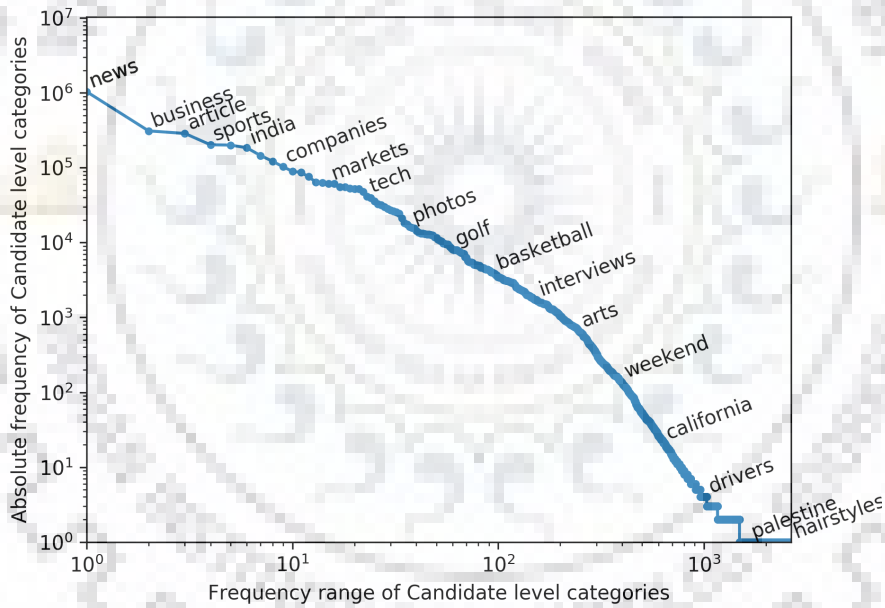


Figure 3.4: Zipf plot for candidate-level categories on log-log scale.

In our work, we leverage URL information to categorize the named events. For each discovered named event, category extractor (Algorithm 4) performs three steps: First, it collects URLs associated with news headlines containing the named event (lines 6-13). Next, URLs are processed and dictionary words are obtained as candidate-level categories (lines 14-20). Finally, candidate-level categories are mapped to high-level categories using Dmoz

**Algorithm 4:** CategoryExtractor( $e, H, U$ )

---

**Input :**  $\langle e, H, U \rangle$   
 Where  $e$  : NamedEvent  
 $H$  : Headlines  
 $U$  : URLs

**Output:**  $\langle HLC_e, CLC_e \rangle$   
 Where  $HLC_e$  : High-Level category  
 $CLC_e$  : Set of candidate-level category

```

1  $H_e \leftarrow \phi$ 
2  $U_e \leftarrow \phi$ 
3  $CLC_e \leftarrow \phi$ 
4  $HLC_e \leftarrow \phi$ 
5  $\{D\} \leftarrow \text{Set of English dictionary words}$ 
6 foreach  $h \in H$  do
7   | if  $e \in h$  then
8   |   |  $H_e \leftarrow H_e \cup h$ 
9   | end
10 end
11 foreach  $h \in H_e$  do
12 |   |  $U_e \leftarrow U_e \cup U_h$ 
13 end
14 foreach  $u \in U_e$  do
15 |   |  $W_u \leftarrow \text{Split } u \text{ by " "}$ 
16 |   | foreach  $w \in W_u$  do
17 |   |   | if  $w \in D$  then
18 |   |   |   |  $CLC_e \leftarrow CLC_e \cup w$ 
19 |   |   | end
20 |   | end
21 end
22  $HLC \leftarrow CLC_e \Leftrightarrow HLC_{DmozTaxonomy}$ 
23  $HLC_e \leftarrow HLC_i : \max(\text{freq}(HLC_i))$ 
24 return  $CLC_e$  and  $HLC_e$ 

```

---

taxonomy (lines 22-23). Sometimes the candidate-level category may be one of the high-level categories. It is possible that more than one candidate level categories are discovered for a named event and each candidate-level category is mapped to different high-level categories. To deal with this issue, first, we collect all possible combinations of candidate-level to high-level mapping and then select high-level category with the highest frequency as the category of named event. In our example, Table 3.6 shows 10 named events along with their candidate-level and high-level categories.



Table 3.6: Named events and their categories.

Named Events	Category	
	Candidate-level	High-level
Commonwealth Games 2014	Sports, News, Tournaments	Sports
Star Wars	Movies, News, Entertainment	Arts
Railway Budget 2015	Business, Budget, India	Business
Cubs Game Day	Sports, Cubs, Baseball	Sports
Independence Day	India, World, Cities	Regional
Bigg Boss 8	Entertainment, Stars, Article	Arts
World Environment Day	Society, India, Punjab	Society
Robot Olympiad 2015	Science, Gadgets, News	Science
2015 Standalone Audited	Companies, Results	Computers
2016 Nuclear Security Summit	World, Breaking, News	News

### 3.2.4 Named Event Duration Extractor

Real world named events are mostly durative i.e., named event persists over a period of time. The proposed system  $NE^2$  extracts popular durations associated with named events. Extracting popular durations of named events help users in exploring the popular named events of a particular time interval. Note that, popular durations of named events do not represent the actual date of named event occurrence, it represents the time when important events of a named event happened. For example, let's consider a named event "Salman Khan Hit-and-Run Case". The popular durations are from 2015-Mar to 2015-May, and from 2015-Dec to 2015-Dec, whereas, "Salman Khan Hit-and-Run Case" happened on 2002-Sep.

We discover the duration of named events using temporal information, associated with news headlines. The procedure of named event duration extractor is illustrated in Algorithm 5. By leveraging temporal information of the news headlines, the popular durations of named events are extracted as follows: First, we obtain temporal information associated with news headlines containing the named event (lines 4-11), where  $T_h$  represents the temporal information of headline  $h$  which contains the named event  $e$ . Next, we prepare a month-wise frequency-based time series for each named event (line 12). The frequency value for a par-



**Algorithm 5:** DurationExtractor( $e, H, T$ )

---

**Input :**  $\langle e, H, T \rangle$   
 Where  $e$  : NamedEvent  
 $H$  : Headlines  
 $T$  : Headline Date

**Output:**  $Dp_e$  : Popular Duration

```

1  $H_e \leftarrow \phi$ 
2  $T_e \leftarrow \phi$ 
3  $Dp_e \leftarrow \phi$ 
4 foreach  $h \in H$  do
5   | if  $e \in h$  then
6   |   |  $H_e \leftarrow H_e \cup h$ 
7   | end
8 end
9 foreach  $h \in H_e$  do
10  |  $T_e \leftarrow T_e \cup T_h$ 
11 end
12  $Dp \leftarrow \text{monthwiseTimeseries}(T_e)$ 
13 foreach  $dp \in Dp$  do
14  | if  $\text{freq}(dp) > \text{mean}(Dp)$  then
15  |   |  $Dp_e \leftarrow Dp_e \cup dp$ 
16  | end
17 end
18 return  $Dp_e$ 

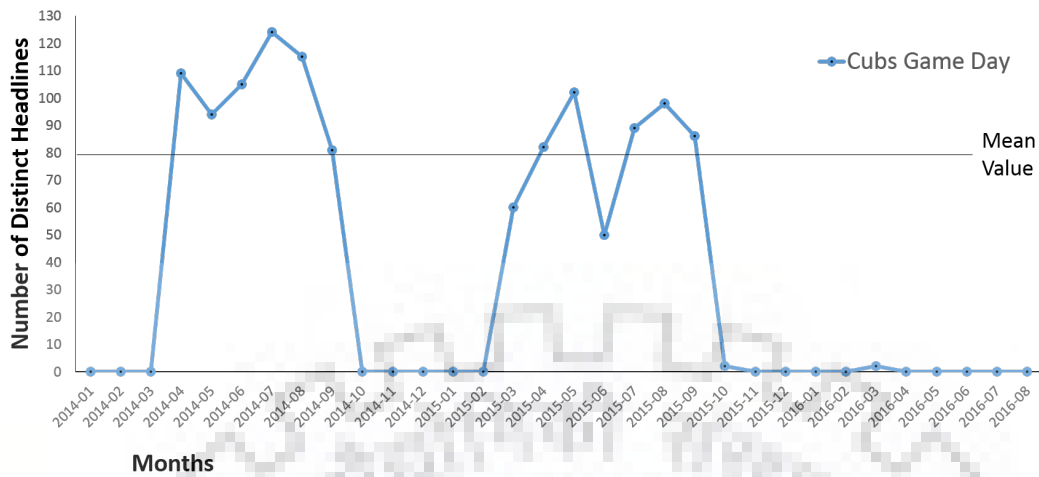
```

---

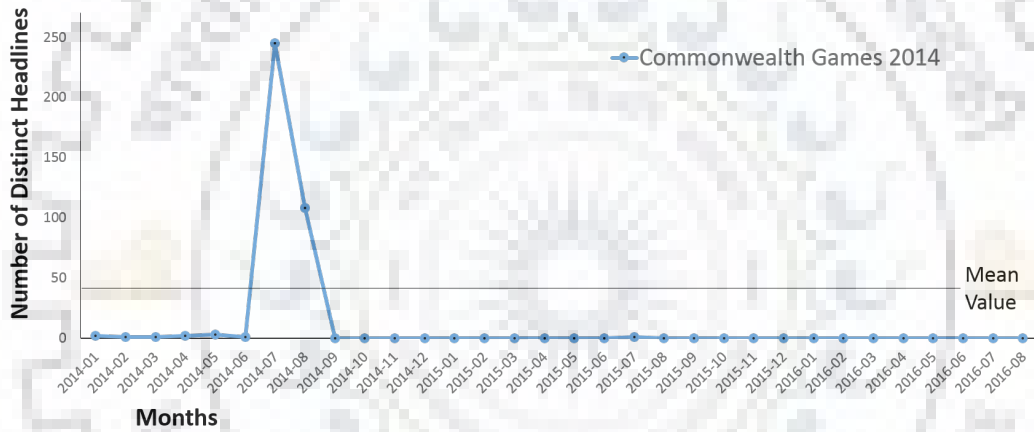
ticular month is the number of unique headlines that contains a named event. Finally, the adjacent months with a frequency greater than the mean value of the time series is output as a duration of the named event (lines 13-17). Choosing the mean value as a threshold is a good choice as mean of value deals with one / two very high or very low values. Figure 3.5 shows the month-wise time series of six named events mentioned in Table 3.6.

Figure 3.5(a) depicts a time series for named event *Cubs Game Day*, horizontal line represents the mean value 79 and peak above the mean value represents the popular durations of named event *Cubs Game Day*. Figure 3.5(b) shows time series for named event *Commonwealth Games 2014* and peak above the mean value 40 represents popular duration for named event *Commonwealth Games 2014*. Similarly, Figure 3.5(c), Figure 3.5(d), Figure 3.5(e), and Figure 3.5(f) represent the time series for named events *Railway Budget 2015*, *Independence Day*, *Bigg Boss 8* and *World Environment Day* respectively where horizontal lines represent the mean values.

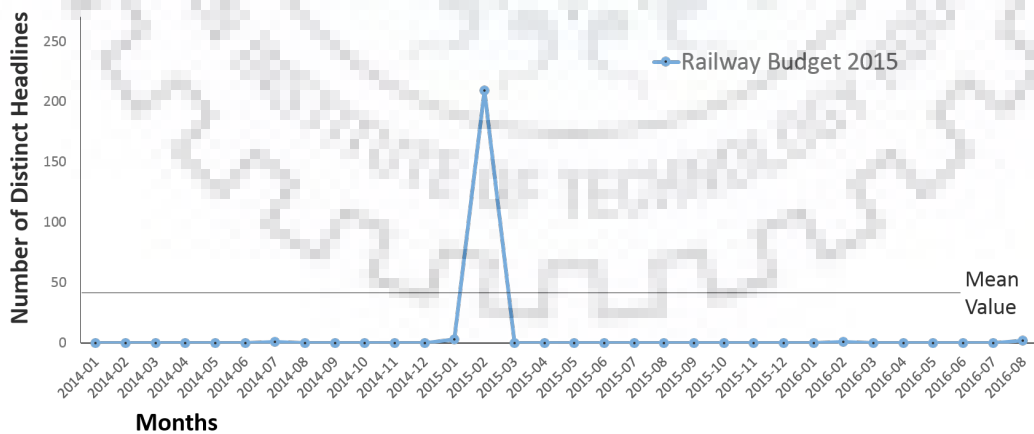
### 3.2 Proposed Approach



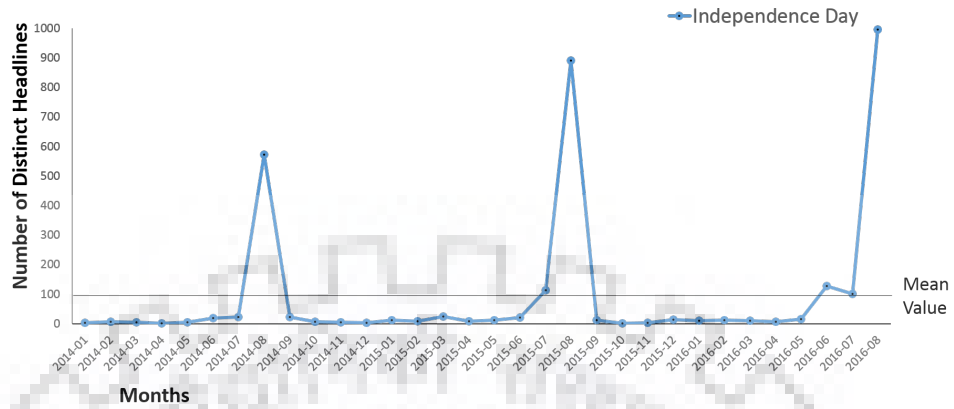
(a) Cubs Game Day



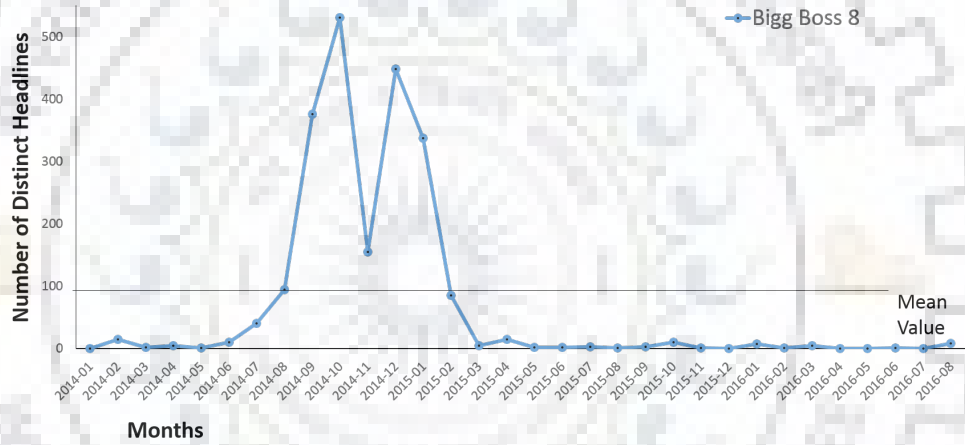
(b) Commonwealth Games 2014



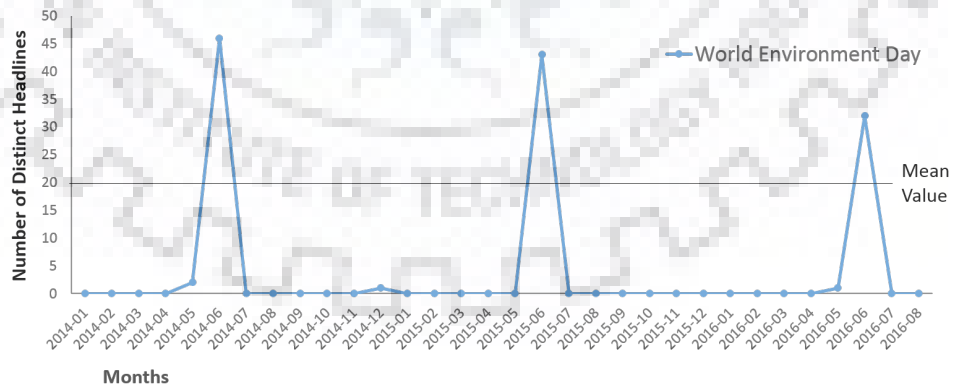
(c) Railway Budget 2015



(d) Independence Day



(e) Bigg Boss 8



(f) World Environment Day

Figure 3.5: Month-wise time series of named events.

## 3.3 Experiments

We have divided experiments into two subsections: quantitative evaluation and qualitative evaluation. Quantitative evaluation section explains the statistics of our system  $NE^2$  on news headlines dataset. Qualitative evaluation section describes a series of experiments to evaluate the performance of our system  $NE^2$ .

### 3.3.1 Quantitative Evaluation

This section shows the statistics and results of the proposed system  $NE^2$  applied to the collected news headline dataset.

#### A Dataset

As named event discovery is performed on news headlines, we have identified 170 English news sources for collecting news data. Some of the news sources are New York Times<sup>5</sup>, Hindustan Times<sup>6</sup>, Time of India<sup>7</sup>, and the majority of the news sources are Indian. Our in-house iMM system [82] crawls each News source at an interval of 30 minutes and collect all the newly published news headlines. The headlines are stored in a headline dataset as a tuple, where each tuple consists of a headline, URL, and temporal information (the time when news headline appeared in media). In summary, we have prepared a dataset of 6.5 million news headlines crawled during January 2014 to August 2016.

#### B Key-phrase Extraction

For key-phrase extraction, we are focusing on the syntactical structure of headlines and observed three presentable patterns of headlines: colon, capitalization, and quotes. Figure 3.6 shows the statistics of prominent feature present in headlines. In the original dataset of 6.5 million news headlines, 26% headlines have colon, 6% headlines have quotes and 86% headlines contain capitalized words. It means approximately 90% headlines contribute to generate the key-phrases.

---

<sup>5</sup><http://www.nytimes.com/>

<sup>6</sup><http://www.hindustantimes.com/>

<sup>7</sup><http://timesofindia.indiatimes.com/>

Table 3.7: Output of key-phrases extraction step.

Colon-based	Quote-based	Capitalized-based
World Cup Support : 2488	Make in India Support : 857	India Support : 210304
White Sox Game Day Support : 257	Good Governance Day Support : 32	Independence Day Support : 1575
Sheena Bora murder case Support : 447	Bigg Boss 8 Support : 217	Union Budget 2015 Support : 594
Bajrangi Bhaijaan Support : 175	2015 BRIT Awards Support : 1	2014 Winter Olympics Support : 37
ICC Cricket World Cup Support : 131	Paranormal Activity 5 Support : 1	LIVE Oscar Awards 2015 Support : 2
Panjab university elections Support : 2	Japan Film Festival 2014 Support : 1	For 2015 World Cup Support : 1

By applying our key-phrase extraction algorithm on all the headlines in the dataset, we obtain 3,96,289 key-phrases using Colon-based, 1,02,788 key-phrases using quotes-based, and 17,81,549 key-phrases using Capitalized-based feature. Table 3.7 shows the sample of key-phrases obtained on the complete dataset along with their support. Note that, key-phrases with support 1 in Table 3.7 also represents a meaningful named event.

### C Named Events

By applying our named event discovery algorithm on 6.5 million input headlines, our system  $NE^2$  discovered 75,689 number of distinct named events. Table 3.8 shows the 20 named events with their support. First eight named events<sup>8</sup> are very popular compared to next six named events<sup>9</sup>, and last six named events<sup>10</sup> are least popular in our news headline dataset.

<sup>8</sup>Star Wars, World Cup 2015, Bigg Boss 8, Bigg Boss 9, Cubs Game Day, World Cup 2014, Independence Day, Rio Olympics 2016

<sup>9</sup>Bihar Assembly Polls, Coal Scam Case, Delhi Election Results 2015, Conversion Row, Kerala Polls, Fathers Day

<sup>10</sup>Saif Ali Khan Assault Case, Tapas Pal Case, Selfie Row, Bangkok Blast Case, Lok Saba Polls

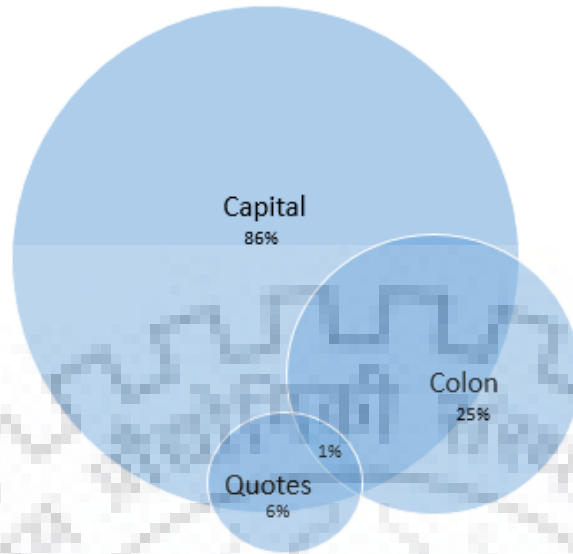


Figure 3.6: Distribution of headlines containing the prominent features: colon, quotes, capitalized words.

Table 3.8: Discovered named events with their support.

Named Event	Support	Named Event	Support
Star Wars	4555	Delhi Election Results 2015	81
World Cup 2015	4196	Conversion Row	81
Bigg Boss 8	3229	Kerala Polls	80
Bigg Boss 9	2421	Fathers Day	79
Cubs Game Day	2252	Saif Ali Khan Assault Case	4
World Cup 2014	2091	Tapas Pal Case	4
Independence Day	1836	Selfie Row	4
Rio Olympics 2016	1767	2008 Mumbai Attack Case	4
Bihar Assembly Polls	83	Bangkok Blast Case	4
Coal Scam Case	82	Lok Saba Polls	4

Figure 3.7 shows the distribution of support for named events in our dataset. The Zipf plot on log-log scale shows that we discovered many events having low support or less popular named events. Thus, we can say that our algorithm is capable to extract the named events

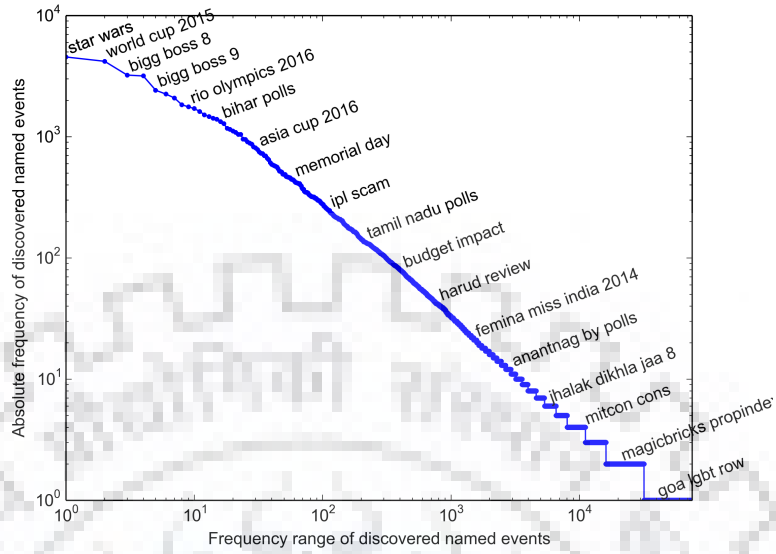


Figure 3.7: Zipf plot for discovered named events on log-log scale.

which are highlighted once or very few times in news media. As shown in Figure 3.7, named event *Star Wars* is very much popular with high support while *Goa Lgbt Row* is a named event with support 1.

Table 3.9: Distribution of discovered named events.

Technique	Colon	Capitalization	Quotes
Number-based	5190	<b>16704</b>	168
Seed word-based	25944	<b>35347</b>	986
Day-based	842	<b>2922</b>	150

Table 3.9 shows the distribution of the number of discovered named events using all three features: colon, capitalization, and quotes. We observed that capitalized key-phrases generate many named events as compared to other key-phrases (values are in bold) because capitalized key-phrases consider all possible noun phrases in this news headlines dataset. Table 3.10 shows the sample of high support named events extracted by all three techniques using all three types of patterns. It may be possible that the named event discovered by one type of key-phrase may be generated through another type of key-phrase. For example, as

### 3.3 Experiments

shown in Table 3.10 named event *Big Boss 9* is extracted from two type of key-phrases: colon-based key-phrase and quotes-based key-phrase.

Table 3.10: Sample of discovered named events.

Technique	Colon	Capitalization	Quotes
Number-based	ICC World Cup 2015	World Cup 2015	Bigg Boss 9
	Rio Olympics 2016	Bigg Boss 8	Kung Fu Panda 3
	Bigg Boss 9	World Cup 2014	Kyaa Kool Hain Hum 3
Seed word-based	JNU Row	Star Wars	2G Case
	Coal Scam	Vyapam Scam	LS Polls
	Bihar Polls	Saradha Scam	Delhi Polls
Day-based	Blackhawks Game Day	Independence Day	Independence Day
	White Sox Game Day	Republic Day	Mother's Day
	Obama's Day	Cubs Game Day	Black Day

#### D Named events Category

Out of 75,689 discovered named events, we have identified categories for 62,950 (82%) named events using URL based method. Note that, we are not able to discover categories of around 18% of named events. This is due to the fact that either URL does not contain categories (6% named events) or Dmoz taxonomy does not provide the mapping of a high-level category to candidate-level category (11.7% named events). Table 3.11 shows sample of named events for which Dmoz taxonomy does not provide a high-level category.

In Figure 3.8, we show the distribution of high-level categories for which extraction is done. The proposed method maps most of the named events to *Regional, Sports, News, and Business* categories. According to [62], articles published in news media mostly fall into the category: *Sports, Business, Politics (Regional), and Arts*. Indirectly, our method confirms the technique that is proposed to classify the articles into categories. Note that, our method only makes use of URL information, whereas, the method proposed by [62] uses the article



Table 3.11: Sample of named events for which Dmoz taxonomy does not provide a high-level category.

Named Events	Candidate level Categories
Syedna Issue	video
Dakar Rally 2017	auto , article
2012 Republican National Convention	live, video
Walter Scott Murder Case	breaking, article
Wolf Movie Review	blog, article

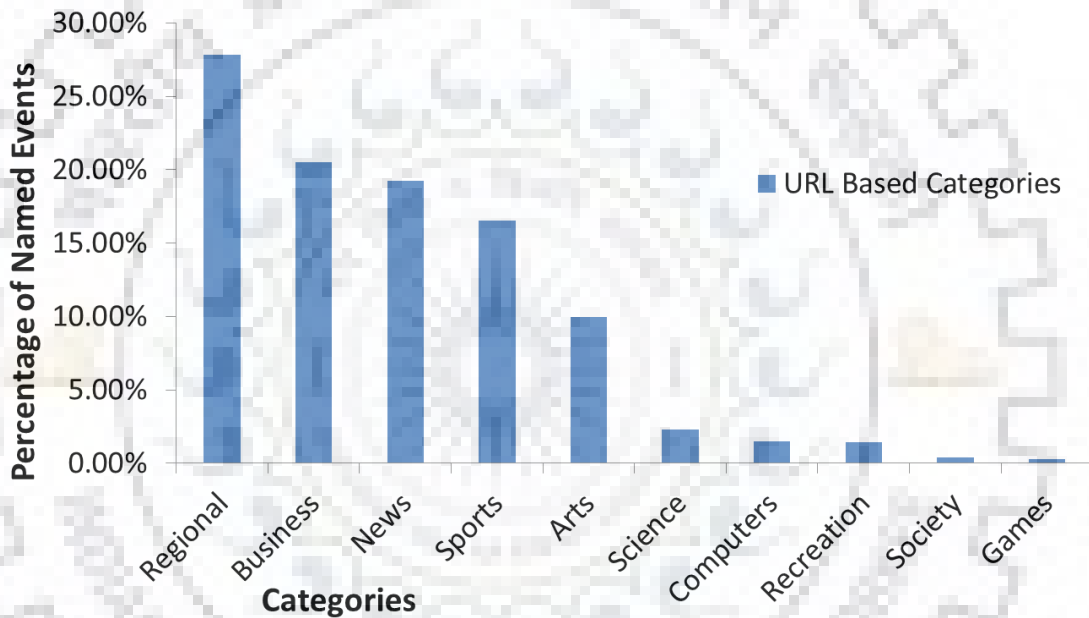


Figure 3.8: Distribution of named event's categories.

contents for obtaining the category of articles. However, extracting news articles for so many URLs are not possible as many articles are not even available online. Table 3.12 shows the top 5 named events for four categories *Sports*, *Business*, *Regional*, and *Arts*.

### E Named events Popular Durations

Our algorithm extracts popular durations for 73,288 (96.8%) named events. Table 3.13 shows the top five named events that have occurred during six months of the period (January 2016 to June 2016). Some named events are durative, some are recurrent and rest named events are popular for short period of times.

Table 3.12: Category wise named events.

<b>Sports</b>	<b>Business</b>	<b>Regional</b>	<b>Arts</b>
World Cup 2015	Coal Scam	Jnu Row	Star Wars
Cubs Game Day	Economic Survey	Bihar Polls	Bigg Boss 9
Rio Olympics 2016	Rail Budget 2016	Saradha Scam	Bigg Boss 8
Sox Game Day	Union Budget 2016	Republic Day	Gunday Review
Asia Cup 2016	Auto Expo 2016	Assembly Elections 2016	Nach Baliye 7

Table 3.13: Sample of discovered named events.

<b>January</b>	<b>February</b>	<b>March</b>
Star Wars	Jnu Row	Jnu Row
Bigg Boss 9	Asia Cup 2016	Coal Scam
Republic Day	Rail Budget 2016	Opening Day
Auto Expo 2016	Union Budget 2016	Jason Day
Salman Khan Hit-and-Run Case	Valentine's Day	Asia Cup 2016
<b>April</b>	<b>May</b>	<b>June</b>
Mother's Day	Yoga Day	Rio Olympics 2016
Earth Day	Memorial Day	Yoga Day
Assembly Elections 2016	Exit Polls	NSEL Scam
VVIP Chopper Scam	Indian Premier League 2016	French Open 2016
West Bengal Polls	Mother's Day	Ishrat Jahan Case

Our temporal information-based algorithm extracts (17%) recurrent named events, which reoccurs after a particular time interval. Figure 3.9 shows the month-wise time series for five recurrent named events, all five named events reoccurs yearly. As shown in Figure 3.9, recurrent named event *Republic Day* occurs in January month each year and recurrent named event *Independence Day* occurs in July with less popularity and in August month with high popularity each year. While named event *Earth Day* is popular in April month each year, *Opening Day* is popular during February month to March month each year, and named event

*International Yoga Day* is popular in July month in 2015 and 2016.

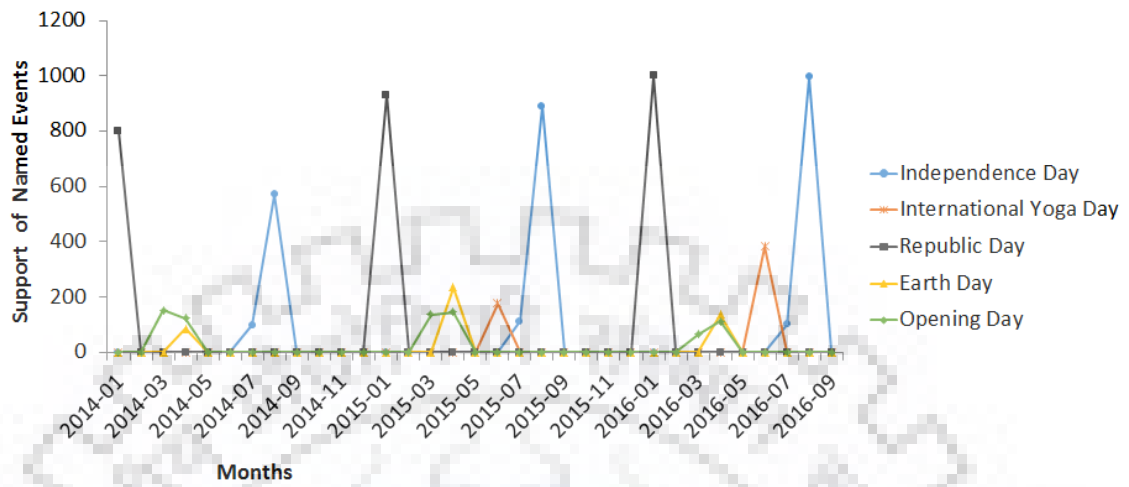


Figure 3.9: Month-wise time series for recurrent named events.

Our algorithm extracts (3.6%) durative named events, which are popular for at least four months consecutively. Figure 3.10 shows the time series for the top five durative named events. As shown in Figure 3.10, named event *Star Wars* was popular for nine months during August 2015 to April 2016, named event *Bigg Boss 8* was popular for six months during August 2014 to January 2015 and named event *Bigg Boss 9* was popular for five months during September 2015 to January 2016.

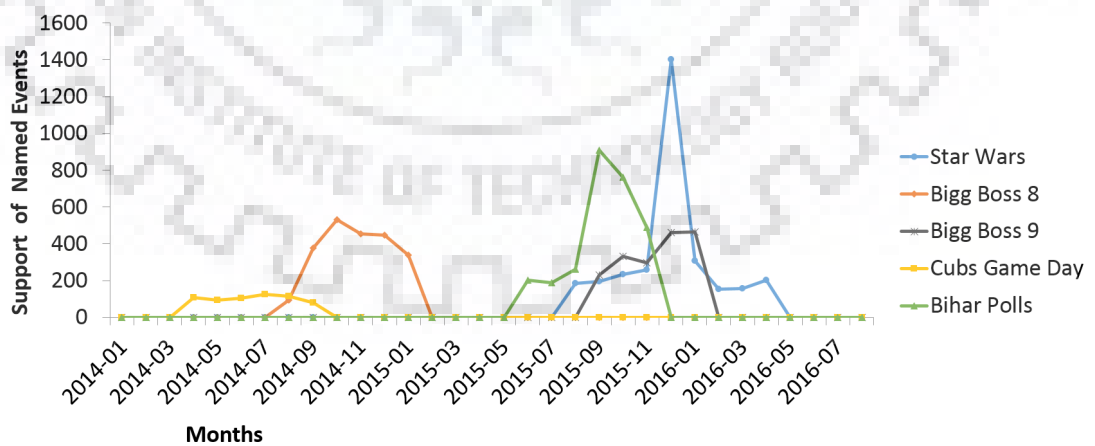


Figure 3.10: Month-wise time series for durative named events.

#### 3.3.2 Qualitative Evaluation

We prepare two datasets to assess the quality of the results of our system: the first dataset is prepared using news media keywords to obtain the accuracy of discovered named events. The second dataset is prepared using Google Trend to assess the correctness of the popular durations of named events. The details of both datasets are given as follows:

- **Meta keyword Dataset.** Each news article is now accommodated with an additional piece of information, in a form of meta keywords. Meta keywords are also important key-phrases that help to the different news articles. For example, meta keywords are BJP, Bihar elections, Bihar rally, Independence Day, etc. Using IMM [82], we obtain 8.2 million meta-keywords collected from October 2015 to August 2016. Note that, meta keyword extraction is a time-consuming task as it requires downloading and parsing of an article.
- **Google Trend Dataset.** Google trend<sup>11</sup> maintains relevant information about the popular search query. Since Google does not provide a corpus on search query log, we use it's online facility to verify the popular durations of named events. For preparing dataset, we randomly pick 250 named events having the frequency higher than 6 as well as 250 named events having the frequency lower than 6 and search trends into Google trends. Google Trend dataset contains the name of the named events and month-wise distribution of query traffic. Named event may contain more than one duration.

#### A Key-phrase Extraction Experiment

We use two sets of headlines for evaluating key-phrases by comparing three state-of-the-art methods (explained in Chapter 2) of key-phrase extraction.

- **Set-1.** Randomly selected 20k headlines from news headline dataset collected during January 2014 to April 2015 (16 months).

---

<sup>11</sup><https://www.google.com/trends>

- **Set-2.** Randomly selected 20k headlines from news headline dataset collected during May 2015 to August 2016 (16 months).

Set-1 and Set-2 are processed to extract key-phrases using our key-phrase extraction method and three state-of-the-art methods. After extracting key-phrases, we apply our named event discovery algorithm to discover named events. For each method, we select top 10 named events and asked an expert to evaluate its correctness. Figure 3.11 describes the number of named events which are correct.

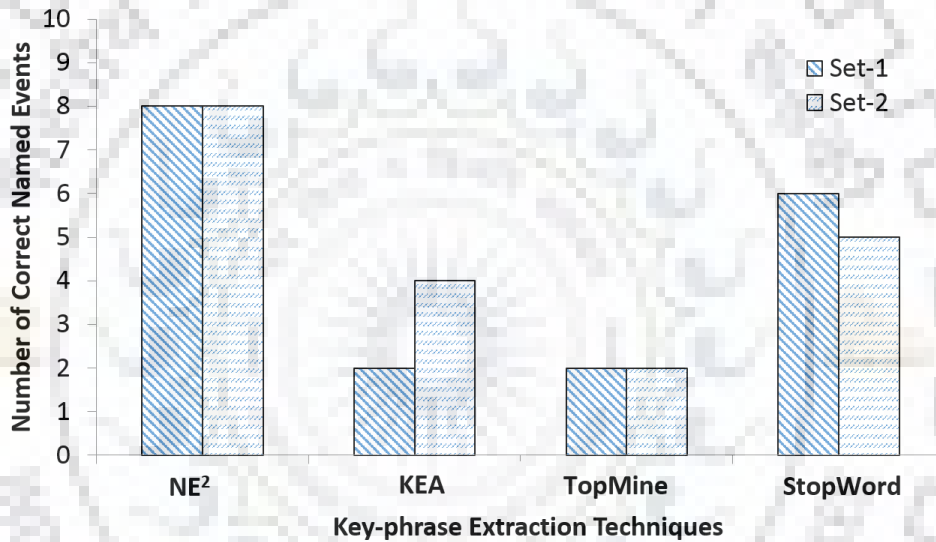


Figure 3.11: Number of correct named events out of selected top 10 named events of each method.

## B Named events Discovery Experiment

To evaluate discovered named events, we perform four sets of experiments: user study, manual tagging, comparison with Wikipedia pages, and meta keyword dataset. For user study, we have selected 50 users and asked each user to verify the named events. For manual tagging, we have prepared a dataset consists of 6000 headlines and these headlines are manually labeled to identify the named events. Wikipedia page availability and meta keywords availability in meta keyword dataset shows that the discovered named event is correct and in a proper format.

1. **User Study.** To evaluate the accuracy of discovered named events, we perform a user study with 50 users. We randomly assign 10 discovered named events to each user from our knowledge base of 75,689 named events. Each user is asked to verify the correctness of the discovered named event. Named events for users are generated by random weighted sampling, where the weight of a named event is directly proportional to their support. A sample of distinct 387 discovered named events are verified by users, out of which 264 (68%) named events are estimated correctly.
2. **Manual Tagging.** In this section, we evaluate discovered named events against the manually annotated dataset. There is no ground truth dataset associated with news headlines and named events. So, we rely on manually annotating a subset of headlines. We randomly select 6000 headlines from six months (January 2016 to June 2016) headlines<sup>12</sup> and manually annotate the named events. Figure 3.12 depicts the month-wise precision and recall of discovered named events against the manually annotated dataset.

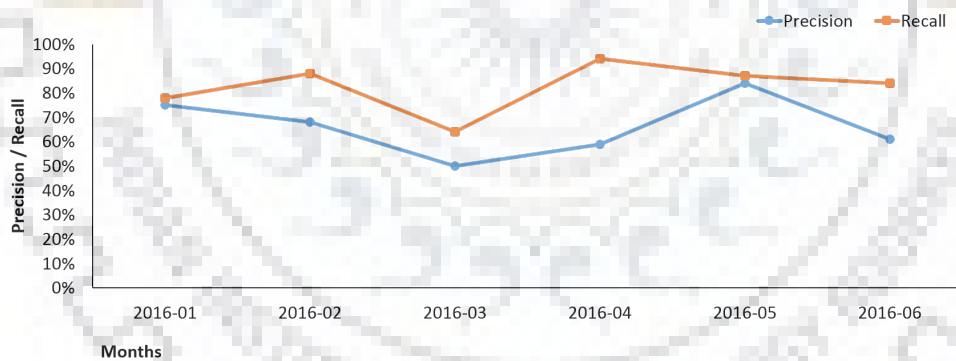


Figure 3.12: Month-wise precision and recall of discovered named events against the manually annotated dataset.

3. **Comparison with Wikipedia Pages.** In this subsection, we identify whether discovered named events have Wikipedia pages or not. We are doing this experiment with the assumption that if a named event has a Wikipedia page, it is correct. For extract-

<sup>12</sup>1000 headlines from each month

ing Wikipedia pages, a python *Wikipedia* package is used and get Wikipedia pages for 62.3% named events.

4. **Comparison with Meta keyword Dataset.** In this section, we compare discovered named events with meta keyword Dataset. We have prepared meta-keyword dataset with the intuition that a named event is correct if it is being used as a meta keyword of any news article.

The proposed approach discovered 75,689 named events using news headlines crawled during January 2014 to August 2016. Now we verify whether named events discovered by our method are present in the meta keywords dataset or not. To verifying named event accuracy with the meta keyword dataset, we have extracted named events from headlines crawled during October 2015 to August 2016 as the meta keyword dataset is prepared only for this particular interval. For day-based and number-based named events, we perform exact matching. Whereas, to verify seed word-based named events, we use a two-phase process: in the first phase, we do the exact match, and the second phase is performed only if the first phase fails. In the second phase, we match the named event by removing its seed word.

**Results and Discussion.** All together 47% named events are present in the meta keyword dataset. We believe there will be more overlap when we disambiguate named events. But this is another research problem, on which we are not focusing in this thesis. Table 3.14 shows sample named events present in meta keyword dataset and named events which are not present in meta keyword dataset. Note that, the named events which are not present in meta keyword dataset are frequent and also correct.

### C Category Extraction Experiments

To evaluate discovered named event's category, we have performed two experiments: crowdsourcing and manual tagging. For crowdsourcing, we have selected 50 users and on the basis of their opinion, we have evaluated the named events category accuracy. For manual tagging, we have prepared two lists, each contains 250 named events, and manual identified their categories. This manually prepared dataset is used as a baseline dataset.



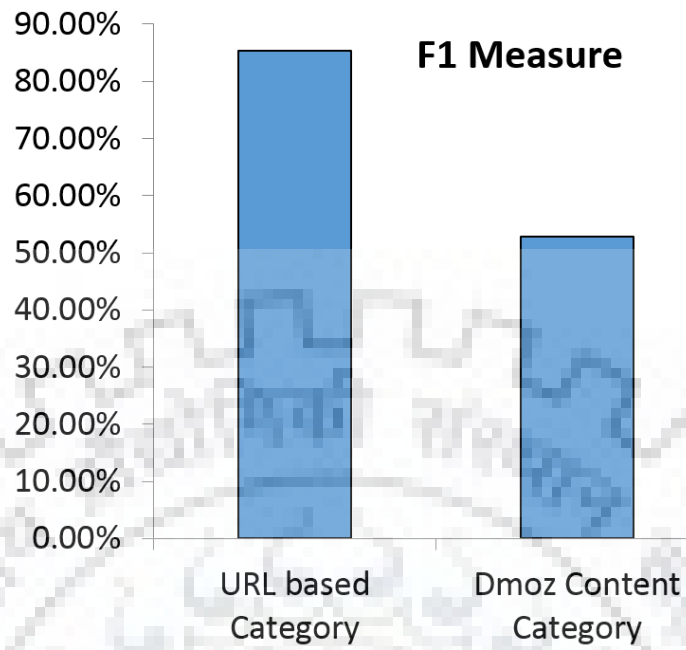
Table 3.14: Named Events.

Named Events	
Present in meta keyword dataset	Absent in meta keyword dataset
Star Wars	World Cup 2015
Independence Day	2017 Uttar Pradesh Polls
ICC World Cup 2015	Bulls Game Day
2016 Rio Olympics	VVIP Chopper Scam
Bigg Boss 8	Maggi Row

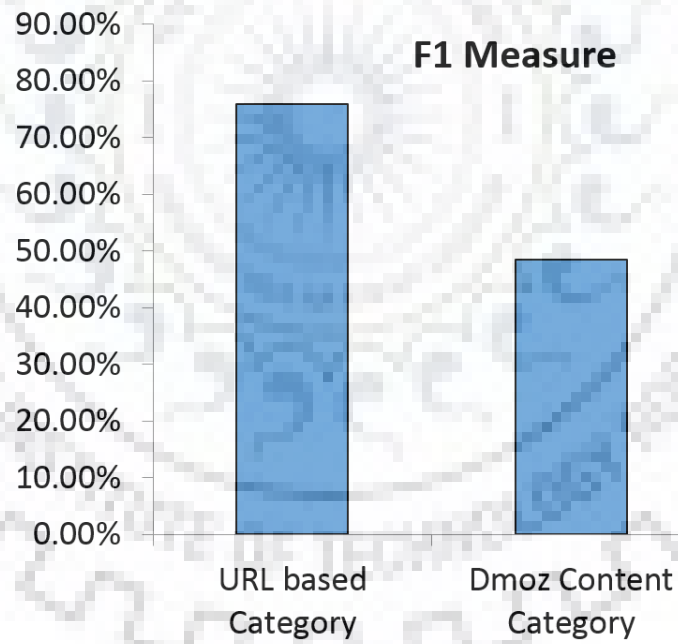
1. **User Study.** To evaluate the accuracy of the discovered named event’s category, we perform a user study with 50 users. We randomly assign 10 discovered named event and category pair to each user. Each user is asked to verify the correctness of the discovered named event’s category. Named event pair for users are generated by random weighted sampling, where the weight of a named event is directly proportional to their support. A sample of distinct 419 named event pairs was verified by users, out of which 300 (71.6%) named event pairs are estimated to be correct.
2. **Manual Tagging.** In this section, we evaluate the categories of the discovered named events and categories suggested by Dmoz content dataset for the same named events against the baseline method.

**Baseline Datasets:** We have prepared two lists of 250 named events each: one of high support of named events and other of low support of named events. We have manually identified the categories of each named event and use it as a baseline to evaluate our URL-based method and Dmoz content-based method for the same set of named events. Recall, that our named event categorization method extracts category for each named event using URLs. We apply the URL-based method and Dmoz content-based categorization on baseline named event lists and evaluate them against their corresponding baseline categories. Results are shown in Figure 3.13. We can see that F1 measure of URL-based method is better than the Dmoz content-based categorization for both types of named events. We are getting higher F1 measure for named events having high support (Figure 3.13(a)) than low support named events (Figure 3.13(b)).





(a) F1 measure of high support named events



(b) F1 measure of low support named events

Figure 3.13: F1 Measure of named event's categories.

## D Named Event's Duration Experiment

This section evaluates whether our technique extracts correct popular durations for named events or not. We use Google Trend Dataset with the intuition that discovered named event's durations are correct if the named event has high search traffic on Google search engine during the same time interval.

Now, we verify whether durations of named events match with the Google trend dataset. Our named event duration extractor extracts at least one duration for 77,720 named events. We evaluate our method with Google trend dataset only for 500 named events. We are verifying whether the duration of a named event generated by our method matches with at least one duration of Google Trend Dataset.

**Result and Discussion.** At least one duration of 392 (78.4%) named events match with durations of randomly picked 500 named events. This result suggests that using only headline dataset, we are able to identify the time interval during which many users have performed a Google search.

### 3.4 Summary

In this chapter, we present a system  $NE^2$  for discovering named events and its related information from news headlines. The proposed technique analyzes the news headlines in a single pass and generates the set of named events by following the filter-and-refine method. Filter method extracts key-phrases from news headlines, whereas, refine method focuses on tagging the named events from extracted key-phrases. Each named event is annotated with the categories, type, support, and popular durations. The system enables users to explore named events using temporal and category information.

Table 3.15: Named events with support, categories, popular durations and its type.

Named Events	Support	Category		Durations	Type
		Candidate-Level	High-Level		
Cubs Game Day	1657	Sports, Cubs, Baseball	Sports	14-Apr to 14-Sep, 15-May to 15-May	Recurrent
Independence Day	864	Entertainment, India, World	Regional	14-Aug to 14-Aug, 15-Aug to 15-Sep	Recurrent
Republic Day	583	National, Cities, India, Article	Regional	15-Jan to 15-Jan, 16-Jan to 16-Jan	Recurrent
White Sox Game Day	571	Baseball, Sports	Sports	15-Apr to 15-May, 15-Jul to 15-Sep	Recurrent
Valentine's Day	548	Entertainment, Story, Celebs	Arts	15-Feb to 15-Feb, 16-Feb to 16-Feb	Recurrent
Sox Game Day	440	Sports, Baseball	Sports	14-Apr to 14-Aug, 15-May to 15-May	Recurrent
Blackhawks Game Day	406	Sports, Hockey, Site	Sports	14-Mar to 14-Apr, 14-Oct to 14-Oct	Recurrent
Bulls Game Day	343	Bulls, Sports, Basketball	Sports	14-Feb to 14-Apr, 14-Dec to 15-Jan	Recurrent
Obama's Day	331	Story	Regional	16-Jan to 16-Jan	Normal
Teachers Day	294	Cities, Entertainment, Nation	Regional	14-Sep to 14-Sep, 15-Sep to 15-Sep	Recurrent
Bigg Boss 8	2668	Entertainment, Gossip, Television	Arts	14-Sep to 15-Jan	Durative
Bigg Boss 9	523	Entertainment, Article, Television	Arts	15-Sep to 16-Jan	Durative
Nach Baliye 7	298	Entertainment, News, Television	Arts	15-Mar to 15-Apr, 15-Jul to 15-Jul	Normal
Ragini MMS 2	276	Entertainment, Play, Movies	Arts	14-Feb to 14-Mar	Normal
Pitch Perfect 2	172	Movies, Entertainment, Hollywood	Arts	14-Feb to 14-Feb, 14-Nov to 14-Nov	Normal
Xiaomi Mi 4	137	Gadgets, Mobiles, Tech	Business	14-Jul to 14-Jul 15-Jan to 15-Apr	Durative
Pyaar Ka Punchnama 2	126	Entertainment, Movies, Hindi	Arts	15-Aug to 15-Oct	Normal

Named Events	Support	Category		Durations	Type
		Candidate-Level	High-Level		
Asus Zenfone 2	122	Gadgets, Mobiles, Reviews	Business	15-Mar to 15-May, 15-Aug to 15-Aug	Normal
Samsung Galaxy Note 4	117	Tech, Gadgets, Mobiles	Business	14-Aug to 14-Nov	Durative
Big Hero 6	107	Movies, Life, Entertainment	Arts	14-Jul to 14-Jul, 14-Oct to 14-Nov	Normal
Star Wars	60	Movies, Life, Entertainment	Arts	14-Feb to 14-Feb, 15-Apr to 15-May	Recurrent
LS Polls	15	Punjab, Nation, Newdelhi	Regional	14-Feb to 14-May	Durative
Vyapam Scam	15	India, National, Article	Regional	15-Jul to 15-Aug	Normal
FTII Row	12	Entertainment, India, National	Regional	15-Jul to 15-Sep	Normal
Thrones Recap	12	News, Story, Entertainment	News	14-Apr to 14-Apr, 15-Apr to 15-Jun	Recurrent
Winners List	11	Entertainment, Hollywood, Movies	Arts	14-Mar to 14-Apr, 14-Aug to 14-Aug	Normal
Piku Review	11	Entertainment, Article, Movies	Arts	15-May to 15-May	Normal
2G Scam	10	India, Article, Business	Business	14-Apr to 14-May, 14-Aug to 14-Sep	Normal
World Cup 2015	3930	Sports, Article, Cricket	Sports	15-Jan to 15-Mar	Normal
ICC World Cup 2015	1019	Sports, Cricket, Article	Sports	15-Jan to 15-Mar	Normal
Union Budget 2015	854	Article, Business, Economy	Business	15-Feb to 15-Mar	Normal
Asian Games 2014	835	Sports, News, Article	Sports	14-Sep to 14-Oct	Normal

Named Events	Support	Category		Durations	Type
		Candidate-Level	High-Level		
World Cup 2014	600	Sports, Football, Soccer	Sports	14-Jun to 14-Jul	Normal
Rail Budget 2015	462	Business, News, Economy	Business	15-Feb to 15-Feb	Normal
Cricket World Cup 2015	435	Sports, Cricket, Article	Sports	15-Feb to 15-Mar	Normal
Auto Expo 2014	427	Business, Story, Industries	Business	14-Feb to 14-Feb	Normal
Union Budget 2014	335	News, Business, Budget	Business	14-Jul to 14-Jul	Normal
US Open 2015	300	Tennis, Sports, Article	Sports	15-Sep to 15-Sep	Normal
2015 World Cup	248	Sports, Article, Cricket	Sports	15-Jan to 15-Mar	Normal
2022 World Cup	147	Football, Sports, News	Sports	14-Jun to 14-Jun, 14-Sep to 14-Sep	Normal
2018 World Cup	131	Football, Sports, Soccer	Sports	14-Jul to 14-Jul, 15-Mar to 15-Mar	Normal
2016 Rio Olympics	103	Sports, News:, Shooting	Sports	15-Mar to 15-May, 15-Jul to 16-Jan	Durative
2014 Lok Sabha	94	News, Article, India	Regional	14-Feb to 14-May	Durative
2002 Hit-and-Run C Ase	91	Entertainment, India, Celebrity Arts	Arts	15-Mar to 15-May, 15-Dec to 15-Dec	Normal
2014 Commonwealth Games	88	Sports, News, Tournaments	Sports	14-Jul to 14-Aug	Normal
2014 FIFA World Cup	76	Football, Sports, Soccer	Sports	14-Jun to 14-Jul	Normal
2014 World Cup	54	Sports, Football, Soccer	Sports	14-Jun to 14-Jul	Normal
2022 Winter Olympics	49	Sports, News, Olympics	Sports	14-Jul to 14-Jul, 15-Jul to 15-Jul	Recurrent



# Chapter 4

## Infobox Mining of Events

This chapter deals with infobox mining of the given events from the newswire dataset. Section 4.1 describes the brief overview of the issues with the existing approaches for extracting key-phrases to the given query. The detailed description of the proposed approach with the algorithms used for data processing phase to key-phrase extraction are mentioned in Section 4.2. Experimental results and performance evaluation of the proposed approach with the state-of-the-art approaches are described in Section 4.3. In Section 4.4, we conclude the proposed approach.

### 4.1 Introduction

News media are publishing ideas, events, and opinions in an increasingly wide range of data formats such as news articles, headlines, videos, tweets, hashtags, etc. The explosion of big news data has sparked the text and data mining research communities to focus on developing systems for news data exploration and analysis. Broadly, two types of news data exploration systems are developed till date: Event centric (GDELT [58], EventRegistry [69]) and Content-centric (STICS [50], EMM [116]). In Event centric system, input query maps to real-world events, whereas, the content-centric system outputs related news articles of a

---

The content of this chapter is presented in the paper:  
Nikita Jain, Swati Gupta, and Dhaval Patel. 2016. *E<sup>3</sup>: Keyphrase based News Event Exploration Engine*. In Proceedings of the 27th ACM Conference on Hypertext and Social Media (HT'16). ACM, New York, NY, USA, 327–329.

given query. Although, both types of systems provide up-to-date news information in real time, but they overload the user with the large amounts of results. Moreover, there is need for a system that enables readers to get a broad overview of the news data generated in response to a user query.

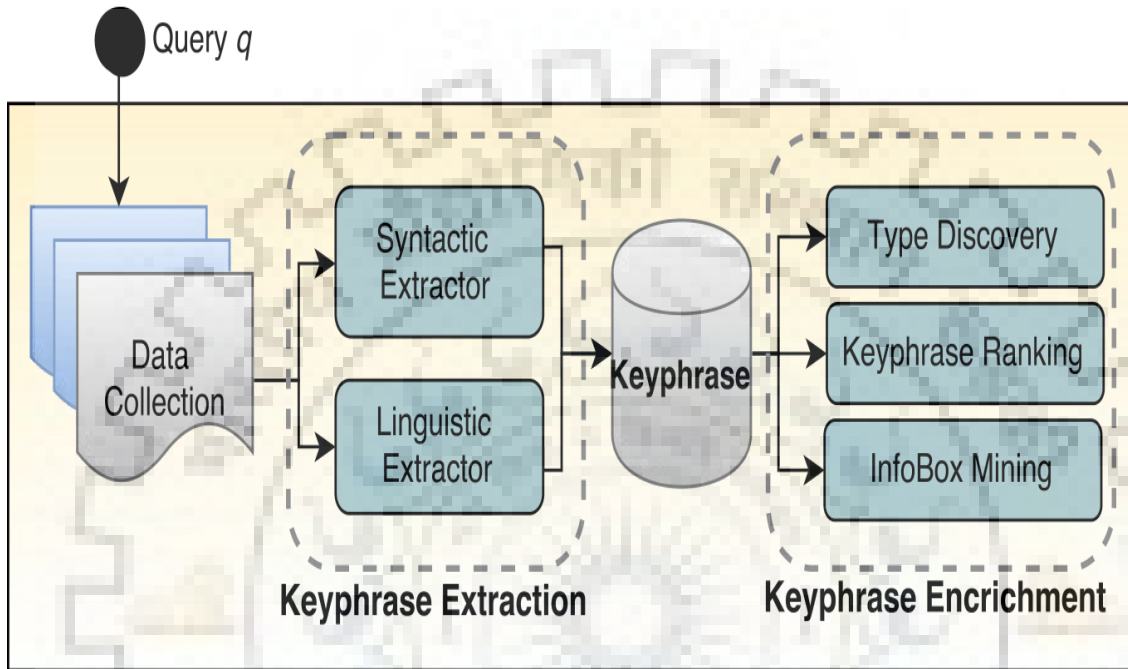


Figure 4.1: Overview of the proposed  $E^3$  System.

## 4.2 Proposed Approach

In this section, we explain our proposed solution for key-phrase extraction from newswire dataset. Figure 4.1 depicts the overview of the proposed approach to extract key-phrases of type: person, location, and organization associated with the given query. The input to the Event Exploration Engine ( $E^3$ ) system is query (event name) and news media.  $E^3$  system follows two-step procedure: key-phrase extraction and key-phrase enrichment. Key-phrase extraction phase is responsible for extracting the key-phrases using syntactical and linguistic features of the text. Key-phrase enrichment phase is responsible for tagging or identifying the information from the extracted key-phrases. The output of the proposed system is infobox which consists of key-phrases, type, rank, and description. The whole procedure of the



proposed system is mentioned in Algorithm 6. For a given query, first, we find out the news sources using our in-build system iMM (line 2). Then, each news source is crawled to collect the dataset consists of headlines, publication time, meta keywords and articles (lines 3-6). After collecting news data, it is processed to extract the key-phrases (lines 7-11) and the detailed description is mentioned in Section 4.2.2. Further, each key-phrase is processed to extract the description related to a given query (lines 12-14). The details on this extraction are mentioned in Section 4.2.3.

### 4.2.1 Data Collection

In this proposed work, we extract the key-phrases related to given query (name of event) from news media. For the input query  $q$ , our data collection module collects data from news media. For data collection, our in-house iMM system is used which periodically extracts news headlines along with their URL, publication date and meta keywords. First, we figure out the headlines and meta keywords containing the input query. Subsequently, URLs of such headlines and meta keywords are processed to crawl the associated news articles and meta description. Thus, for a given query  $q$ , the data collection module prepared a set of news records  $D_r$ , where each entry is described as quadruplet Headline, Publication Time, Keywords, Articles.

### 4.2.2 Key-phrase Extraction

In this section, the collected data is processed to extract the key-phrases. The collected data is of two types: short length and long length data. Short length data consist of meta keywords and news headlines. Whereas, long length data consist of news articles and meta descriptions. We observe that news headlines are short in length and contain special tokens such as colon (“ : ”), quotes (“ ` ”), hashtags (“ # ”) to highlight the important facts. On the other hand, meta description and news articles are long passage texts and follow grammatical rules. Thus, we extract the key-phrases by considering two different types of extractors: syntactic extractor and linguistic extractor. All the news records related to given query are processed to obtain a set of key-phrases along with the number of times they are generated.

---

**Algorithm 6:** Key-phrase based Event Exploration Engine.

---

**Input :**  $q$ : Query  
**Output:**  $Infobox : \langle K, Type, Rank, Description \rangle$

- 1  $D_r \leftarrow \phi$
- 2  $K \leftarrow \phi$
- 3  $URL_{set} \leftarrow \text{Collect}(\text{iMM}, q)$
- 4 **foreach**  $u \in URL_{set}$  **do**
- 5      $D \leftarrow \langle \text{Headline}, \text{Publication Time}, \text{Keywords}, \text{Article} \rangle$
- 6      $D_r \leftarrow D_r \cup D$
- 7 **end**
- 8 **foreach**  $d_r \in D_r$  **do**
- 9      $K \leftarrow K \cup d_r.\text{Keywords}$
- 10     $K \leftarrow K \cup \text{SyntacticExtractor}(d_r.\text{Headlines})$
- 11     $K \leftarrow K \cup \text{LinguisticExtractor}(d_r.\text{Article})$
- 12 **end**
- 13 **foreach**  $k \in K$  **do**
- 14     $Infobox \leftarrow \text{KeyphraseEnrichment}(k, q, \text{Publication Time})$
- 15 **end**

---

### A Syntactic Extractor

Syntactic Extractor utilizes special characters such as colon, quotes, and hashtags for key-phrase extraction. For each headline collected for a given query, syntactic extractor (Algorithm 7) works as follows: Hashtag based key-phrases are extracted in lines(4-6), colon-based and quotes-based key-phrases are extracted in lines (7-14) and lines (15-17) respectively. The details on these features are described as follows:

- For hashtag processing, capitalized letters and numeric values help in separating the phrases as hashtag phrases are not space separated. Processed hashtag phrases are extracted as key-phrases.
- For colon-based key-phrases, colon is used to split the headline into two sub-parts. The part with the short length is considered as key-phrase.
- For quotes-based key-phrase, the text enclosed between the quotes is extracted as key-phrase.

**Algorithm 7: SyntacticExtractor(H)**


---

```

Input :  $H$  : headline
Output:  $K$  : Set of key-phrases
1  $Q_h \leftarrow$  Text between Quotes in  $h$ 
2  $K \leftarrow \phi$ 
3 foreach  $h \in H$  do
4   if  $\{ \# \} \in h$  then
5      $K \leftarrow K \cup \text{ProcessHashTagPhrase}$ 
6   end
7   if  $\{ : \} \in h$  then
8      $\langle l_1, l_2 \rangle \leftarrow \text{Spilt } h \text{ by } \{ : \}$ 
9     if  $\text{len}(l_1) < \text{len}(l_2)$  then
10       $K \leftarrow K \cup l_1$ 
11    else
12       $K \leftarrow K \cup l_2$ 
13    end
14  end
15  if  $\{ " \} \in h$  then
16     $K \leftarrow K \cup Q_h$ 
17  end
18 end
19 return  $K$ 

```

---

**B Linguistic Extractor**

Linguistic extractor applies language specific part of speech (POS) tagging on meta description and article. Linguistic extractor (Algorithm 8) process each line from the input article or meta description to annotate nouns, adjectives, noun apostrophe connector and numbers to noun family (lines 2-5). Then the accumulated noun and its family are extracted as key-phrases (line 6).

**4.2.3 Key-phrase Enrichment**

As discussed previously, the size of generated key-phrases may be large and noisy. To resolve this problem, the proposed key-phrase enrichment module helps in extracting valuable information by filtering extracted key-phrases. The key-phrases are filtered using news media specific stop words such as update, video, photo, pti, and others. Next, we apply case normalization and remove duplicate key-phrases. At this point, noisy keyphrases are removed.

---

**Algorithm 8:** LinguisticExtractor(A)

---

**Input** :  $A$  : Article,  $M$  : Meta description

**Output:**  $K$  : Set of key-phrases

```

1  $K \leftarrow \phi$ 
2 foreach  $a \in A, M$  do
3   Replace Adjectives and its family (JJ, JJR, JJS) with Noun family
4   Replace cardinal number (CD) with Noun family
5   Replace coordinating conjunction (CC) with Noun family
6    $K \leftarrow$  select accumulated noun and its family
7 end
8 return  $K$ 

```

---

The remaining key-phrases are passed through the type discovery, key-phrase ranking, and infobox mining modules. In the type discovery module, NER tagger is used to classify key-phrase among three types: *person*, *location*, and *organization*. Using Key-phrase ranking module, key-phrases are organized according to the value of frequency, novelty, and activeness. Key-phrases with type person and organization are preferred for infobox mining.

---

**Algorithm 9:** KeyphraseEnrichment( $K, Q, T$ )

---

**Input** :  $\langle K, Q, T \rangle$

Where  $K$  : Key-phrase

$Q$  : Query

$T$  : Temporal Information

**Output:**  $Infobox : \langle Type, Rank, Class, Description \rangle$

```

1  $Type \leftarrow TypeDiscovery(K)$ 
2  $Rank \leftarrow KeyphraseRanking(K, T)$ 
3  $time \leftarrow TimeInterval(k, q, T)$ 
4  $Class \leftarrow ClassFinder(k, time)$ 
5  $Description \leftarrow TypeDiscovery(k, q, T)$ 
6 return  $Infobox$ 

```

---

## A Type Discovery

Type discovery phase classifies the extracted key-phrases among three classes: *person entity*, *location entity*, and *organization entity*. Key-phrases do not belong to the abovementioned

Table 4.1: Results of type discovery phase for given query “*Bihar Election*”.

Person	Location	Organization
Narendra Modi	New Delhi	Bihar Assembly
Nitish Kumar	Lok Sabha	Shiv Sena
Jitan Ram	Red Fort	Election Commission
Amit Shah	West Bengal	Lok Janshakti Party
Ram Vilas Paswan		Bihar BJP
Sonia Gandhi		Rashtria Janata Dal
Shahnawaz		Hindustan Awam Morcha

classes known as *News Concepts*. Language-specific NER is used to classify the key-phrases. As existing NER tagger is not suitable for Indian named entities, so we also used a separate list<sup>1</sup> of 90,000 Indian names, which contain all three types of named entities (Person, location and organization) [63]. Table 4.1 shows sample key-phrases of three classes extracted for given query “*Bihar Election*”.

## B Key-phrase Ranking

Key-phrase ranking phase sorts the extracted key-phrases based on their support count, activeness and novelty. Activeness and novelty of key-phrase are calculated with respect to the publication time of their corresponding articles and headlines. To find out whether the extracted key-phrase is novel or active: first, time series for given query is built at which support count is very high. Extracted key-phrase is novel, if it is popular only during the time duration of the given query. Whereas, key-phrase is active if it is popular around the time duration of the query. Table 4.2 shows novel and active key-phrases extracted for query “*Bihar Election*” and key-phrases are arranged based on their support count from top to bottom and left to right. The key-phrase “*Bihar Election*” is much popular in comparison to the key-phrase “*Election Results*”. In the table, novel and active key-phrases are mentioned with the symbols (#) and (+) respectively.

<sup>1</sup><https://github.com/NikkiJain09/Transliteration>

Table 4.2: Novel and Active Key-phrases associated with query “*Bihar Election*”.

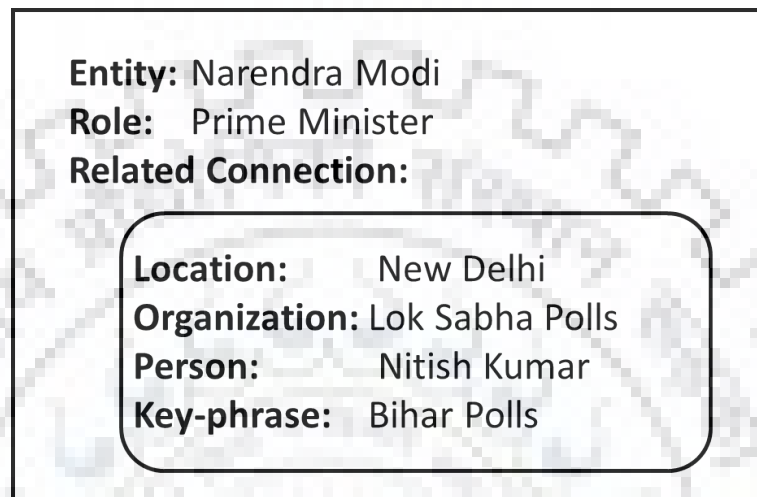
Bihar Election (+)	RJD (+)
BJP	NDA (+)
Bihar (+)	Lalu Prasad (+)
Nitish Kumar (+)	Congress
Bihar Polls	Bihar Assembly
Grand Alliance (#, +)	Elections
Narendra Modi (+)	Election Results (#, +)

### C Infobox Mining

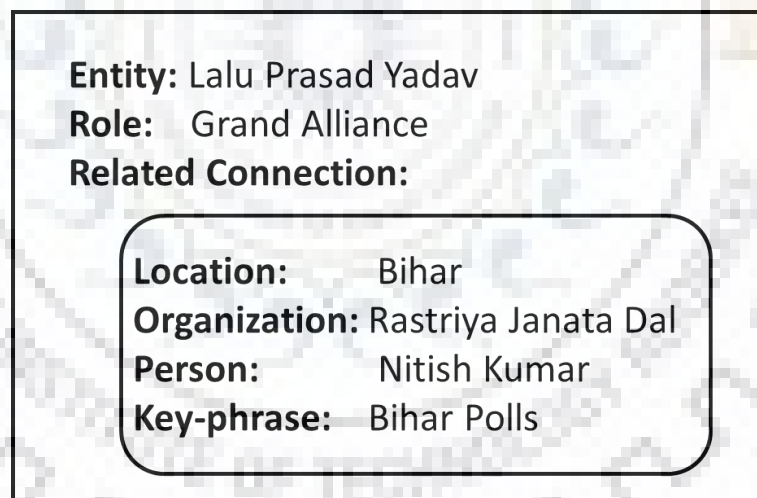
Infobox mining phase extracts the description for the selected key-phrase in context of the given query. Description consists involved entity, role and the topmost related connection for selected key-phrase with respect to the given query. Extracted key-phrases with person entity, location entity, and organization entity classes are selected for the topmost related connection based on high support count and co-occurrence value with the selected key-phrase. A phrase which frequently co-occurs with the selected key-phrase in the collected news corpus is preferred for the *Role* of selected key-phrase. Figure 4.2 depicts infobox for key-phrases: Figure 4.2(a) for entity “*Narendra Modi*” and Figure 4.2(b) for entity “*Lalu Prasad Yadav*” in context of query “*Bihar Election*”. For key-phrases “*Narendra Modi*” and “*Lalu Prasad Yadav*”, Google search engine infobox shows role as *Indian politician* that is very generic. Whereas our proposed system finds out the specific role as *Prime Minister* and *Grand Alliance* for key-phrases *Narendra Modi* and *Lalu Prasad Yadav*, respectively.

## 4.3 Experiments

In this section, we perform a series of experiments to evaluate the performance of our proposed approach for key-phrase extraction against the state-of-the-art approaches. To the best of our knowledge, there is no labeled data available for key-phrase extraction from news media. So, we evaluate our proposed approach based on three aspects: richness of key-phrases, meaningfulness of key-phrases, and correctness of key-phrases.



(a) Infobox of 'Narendra Modi' for query 'Bihar Election'



(b) Infobox of 'Lalu Prasad Yadav' for query 'Bihar Election'

Figure 4.2: Infobox of key-phrases with respect to query "Bihar Election".



The richness of the proposed system is evaluated on the basis of the number of key-phrases (active, novel, person, location, organization and news concepts) extracted for the given query. To identify the meaningfulness of extracted key-phrases, we compare the results of our method with the query log of Google Trend. For the correctness of key-phrases, we use a manually annotated dataset as a baseline.

### 4.3.1 Richness of Key-phrases

This section evaluates our key-phrase extraction approach against the state-of-the-art approaches: KEA and ToPMine in terms of quantity, interesting and effective understanding. KEA system is trained with the passage text, which follows different writing patterns as compared to news data. So, we trained KEA model for six queries<sup>2</sup> including general and specific with the help of our data collection module (Section 4.2.1). The training is done with 1.2 million tokens.

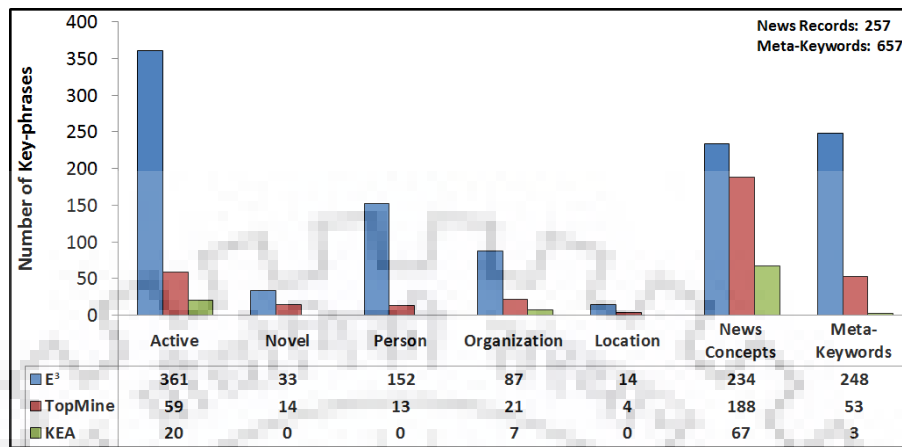
Figure 4.3 depicts the statistics for three approaches: our proposed key-phrase based  $E^3$ , TopMine, and KEA. Figure 4.3(a), Figure 4.3(b), Figure 4.3(c), Figure 4.3(d), Figure 4.3(e), and Figure 4.3(f) shows the distribution of extracted key-phrases with type *Active*, *Novel*, *Person*, *Location*, *News Concepts*, and *Meta keywords* for queries *Bihar Election*, *Paris Attack*, *Vyapam Scam*, *Chennai flood*, *Election*, and *ISIS* respectively. From Figures, we observed that our proposed method outperforms against the state-of-the-art methods and has notable contribution to the work of key-phrase extraction. Some significant considerations are as follows:

- In comparison to KEA and ToPMine key-phrase extraction method,  $E^3$  extracts many related entities (person, location, and organization type). Consequently, encouraging to retrieve the entities or other descriptions using this infobox mining approach related to given entity and query.
- Our system provides an interesting and effective understanding of the results as compared to the other two state-of-the-art approaches, as  $E^3$  extracts the large number of

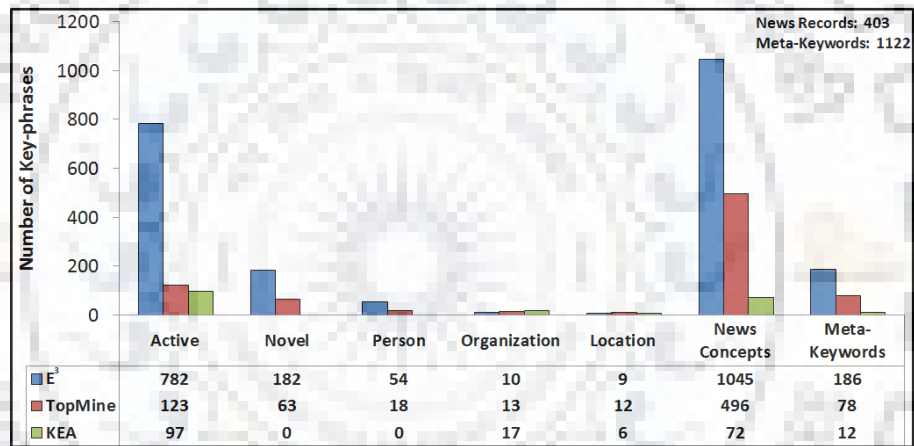
---

<sup>2</sup>Bihar Election, Paris Attack, Vyapam Scam, Chennai flood, Election, and ISIS

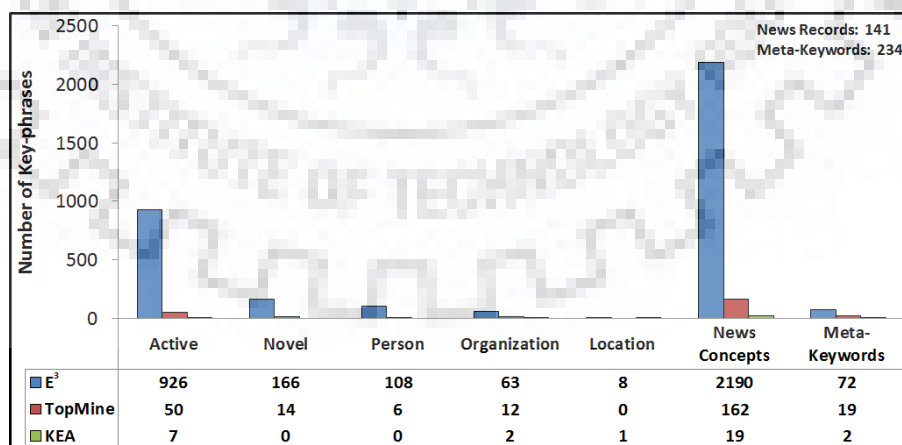




(a) Distribution of extracted key-phrases for query 'Bihar Election'

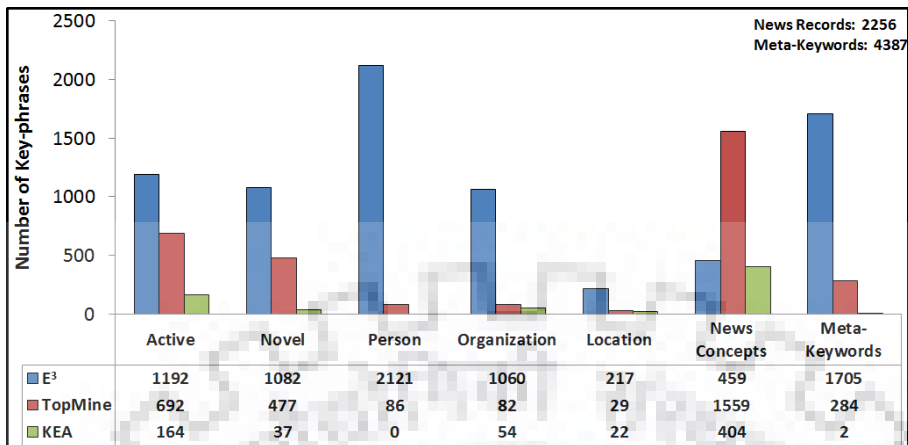


(b) Distribution of extracted key-phrases for query 'Paris Attack'

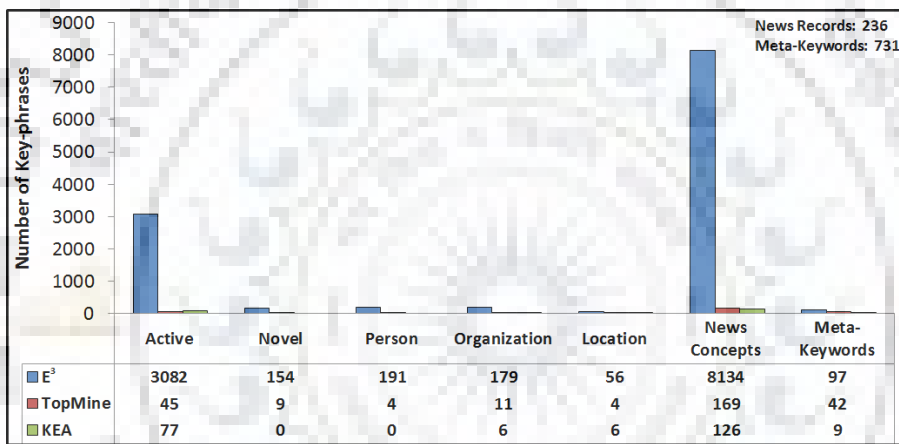


(c) Distribution of extracted key-phrases for query 'Vyapam Scam'

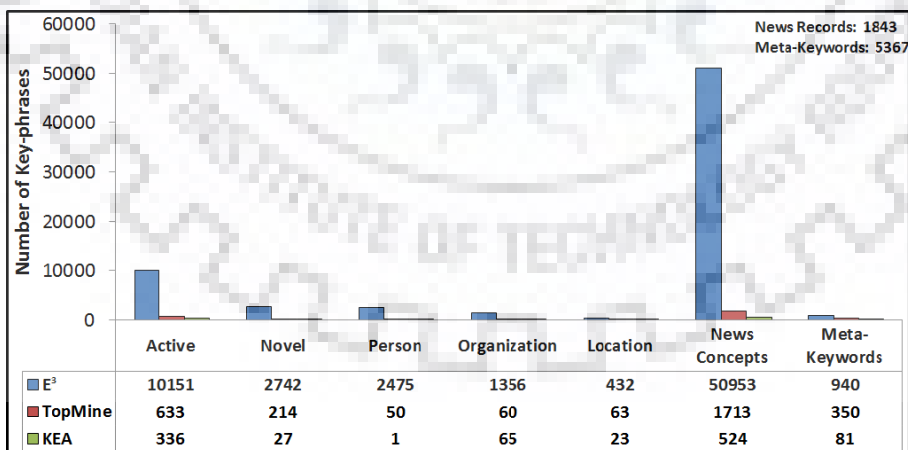
### 4.3 Experiments



(d) Distribution of extracted key-phrases for query 'Election'



(e) Distribution of extracted key-phrases for query 'Chennai Flood'



(f) Distribution of extracted key-phrases for query 'ISIS'

Figure 4.3: Distribution of extracted key-phrases for  $E^3$ , KEA and ToPMine key-phrase extraction techniques.

active and novel key-phrases from others. We define interestingness and effectiveness of key-phrases on the basis of their novelty and activeness.

### 4.3.2 Meaningfulness of Key-phrases

In this section, we compare the result of our method with Google search query log. Since Google does not provide such dataset, we use Google’s public web facility Google Trends<sup>3</sup> which shows the Google search query volume and related queries with respect to time. We evaluate the extracted key-phrases from  $E^3$  with the related search queries of Google Trends for the given query and results are shown in Table 4.3. First, we find out the number of exact matches. Subsequently, we divide the rest queries into three sub-types: related queries, spelling mistakes, and matches including subsets. Examples for spelling mistakes are like “*Bihar Vidhan Sabha Election*” and “*Bihr Vidhan Sabha Election*”. For Matches including subsets, one example is “*Bihar Election*” and “*Exit Poll Bihar Election*”.

Table 4.3: Comparison of extracted key-phrases from  $E^3$  against Google Trend queries.

Number Of	Bihar Election	Election
Complete matches	7	25
Related queries	23	50
Spelling mistakes	3	3
Matches including subsets	13	35

### 4.3.3 Correctness of Key-phrases

We evaluate extracted key-phrases against the manually annotated dataset in this section. In order to create baseline dataset for two queries ‘*Bihar Election*’ and ‘*Vyapam Scam*’, we randomly selected 50 headlines and their associated data for each query with the help of data collection module (Section 4.2.1). Further, we manually extracted the key-phrases out of the collected dataset. Figure 4.4 depicts the precision and recall for the extracted key-phrases from our  $E^3$  system against the baseline dataset.

<sup>3</sup><https://trends.google.com/trends>

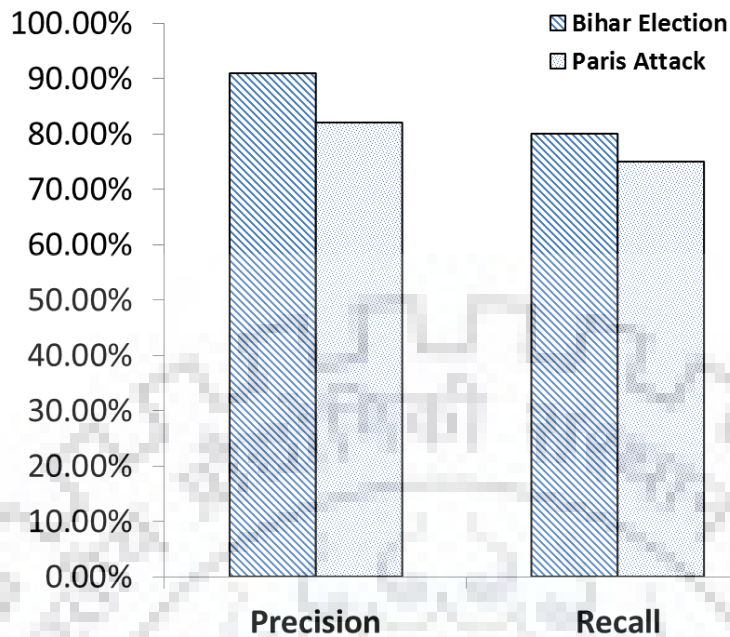


Figure 4.4: Precision and Recall of extracted key-phrases.

## 4.4 Summary

In this chapter, we proposed a method that extracts a brief overview of news data in terms of key-phrases for the given query. This approach follows two-stages: key-phrase extraction and key-phrase enrichment. To deal with the heterogeneity of news data (news headlines, meta keywords, and news articles), we use two types of extractor for key-phrase extraction: syntactical extractor and linguistic extractor. Further extracted key-phrases are enriched by labeling them as a novel, active, and news-concept. We also classify the key-phrases as person, location, and organization class. Based on the performed experiments, we observed that the proposed system  $E^3$  outperforms to the state-of-the-art approaches of key-phrase extraction.

## Chapter 5

# Event Extraction from Streaming Tweets

This chapter discusses an unsupervised methodology that uses self-learning-based max-margin clustering approach to detect events from tweets. In Section 5.1, we discuss the characteristics, challenges with Twitter data, and issues with the existing work of event detection. Section 5.2 details the proposed methodology from pre-processing to event detection techniques. Our method's performance results are mentioned in Section 5.3 and we conclude the proposed methodology in Section 5.4.

### 5.1 Introduction

Nowadays Twitter is considered as one of the most widely used social media platform globally for spreading information, sharing opinions, and expressing personal views. Till December 2017, Twitter has on an average 328 million monthly active users<sup>1</sup>, which clearly depicts its versatility. Twitter has a global ranking of 12 among all the websites available on the Internet on the basis of the user traffic<sup>2</sup>. Since a typical post on Twitter (also known as tweet) has a limitation of 140 characters, users prefer short words, smilies, emojis, emocations to complete their messages. In addition, tweets may consist of mixed data, available

---

The content of this chapter is presented in the paper:  
Swati Gupta and Biplab Banerjee “*Unsupervised Event Detection using Self-learning-based Max-margin Clustering: Analysis on Streaming Tweets*,” IETE Journal of Research, August 2018.

<sup>1</sup><https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>

<sup>2</sup><https://www.alexa.com/siteinfo/twitter.com>

in the form of text, images, videos, URLs. However, considering the near real-time information conveyed by the tweets, it is of great importance to extract semantic information from them, which is nevertheless a challenging task given the heterogeneous contents of the tweets. Tweets have recently been used for various significant purposes like event extraction [65, 70, 103, 140], opinion mining [61, 72], sentiment analysis [88, 141], disaster management [7, 53], and stock market prediction [122].

Among others, the problem of event extraction from Twitter data streams poses sufficient challenges due to the aforementioned issues. An event can loosely be defined as the depictions of surrounding states of the users at the temporal scale. In short, everything posted by the users on Twitter is largely focused on events taking place around them. In recent past, there are various methods suggested to extract events from Twitter data: some of them consider keyword burstiness [1, 107] while others depend on some external thesaurus [70] to predict the event candidates. However, given the noisy contents of the tweets along with the possible heterogeneity in the contents, it is potentially difficult to predict the events from tweet streams without concrete supervision.

Inspired by the aforementioned discussions, we focus on extracting events from tweets (text data) using a novel unsupervised learning approach. Our proposed method is quite different from the existing techniques in the following aspects: First, we seek to understand the semantic of each tweet using a pooling-based distributed word representations. We perform extensive pre-processing in order to eliminate irrelevant constructs from the tweets and assign higher weights to event-specific words automatically. The obtained representation ensures high intra-class and low inter-class similarity measures. Further, in order to group semantically similar tweets automatically, we propose a novel two-level clustering framework considering DBSCAN [38] and a self-learning-based max-margin clustering. The proposed self-learning-based max-margin clustering is initialized based on the results of the DBSCAN stage and iteratively updates the clusters until no further change is observed in the clustering outcomes.

Broadly, the existing models extensively use supervised strategies in highlighting the events, apart from manually fixing informative keywords for different events. Instead, we seek to obtain vector representation of the tweets in a semantic space where the standard

machine learning algorithms can be applied. In addition, we consider some of the intricate problems in data clustering due to data overlap and presence of outliers and propose an iterative self-learning algorithm based max-margin clustering. Our method is generic and can be adapted to deal with different event tags without loss of generality.

## 5.2 Proposed System Overview

This section details the proposed system's architecture and flow graph as shown in Figure 5.1. In particular, the framework comprises four broad components: (i) Tweet repository is constructed initially by collecting streaming tweets on a daily basis using the publicly attainable twitter streaming API. (ii) These tweets are subsequently pre-processed in order to reduce the effects of noisy and irrelevant contents. (iii) A distributed word embedding-based vector-space representation of the refined tweet is further obtained, and (iv) Finally, a novel self-learning-based clustering is introduced in order to group semantically similar tweets.

### 5.2.1 Tweet Pre-processing

Considering that the tweets are generically user-defined texts with a limit of 140 characters, it is expected to contain noisy or apparently meaningless data in the form of short words, hashTags, and other special symbols. To cope with noisy data, we develop a tweet processing framework consisting of a series of meaningfully defined refinement stages (Figure 5.2). Note that we select tweets in US English in terms of JSON objects for the experiments in this chapter. Let  $T = \{t_1, t_2 \dots, t_z\}$  be the  $z$  raw tweets collected initially. The pre-processing stages carried out subsequently are mentioned in the following:

#### A Remove URLs

URLs in tweet referred to a linked web-page. Since, we are focusing on extracting events from the tweet contents, it is expansive and not much beneficial to consider URLs for further processing. Hence, we choose to remove URLs from tweets.

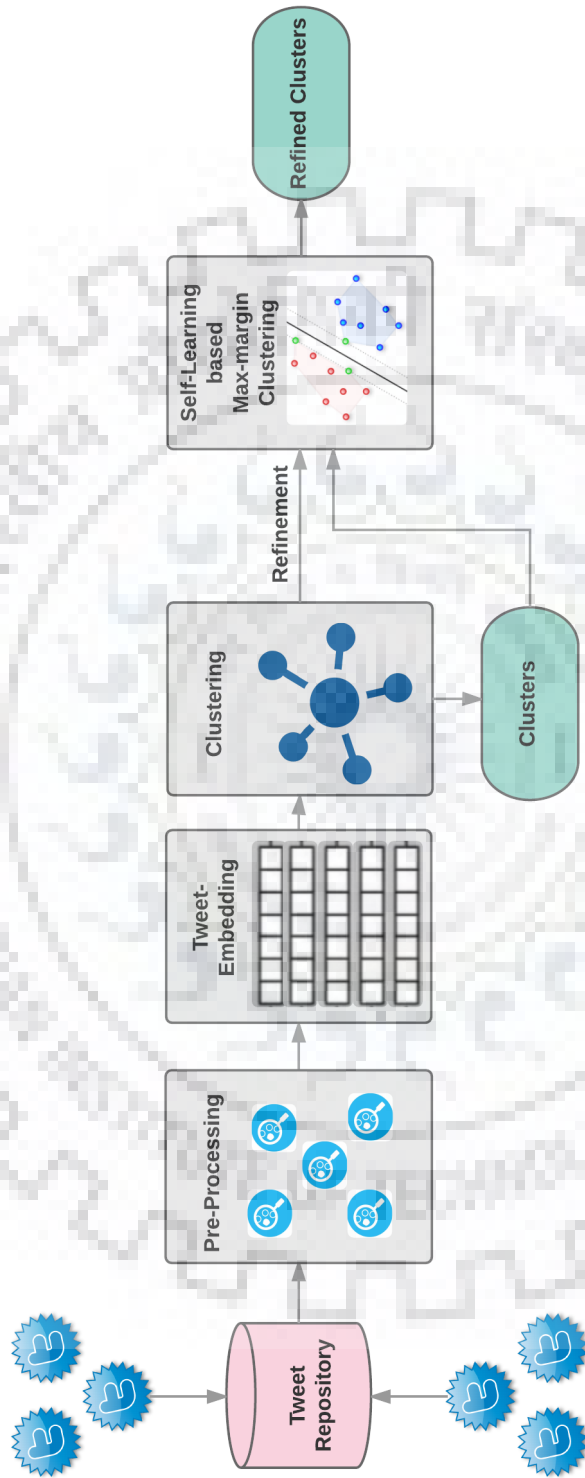


Figure 5.1: Overview of the proposed method.



## B Remove Retweeting and @ Tags

Retweeting is an action of re-posting an already posted tweet. On the other hand, @ Tags refer to user's names to which post is related. Given the irrelevance of such tags in ascertaining the event details, we removed such tags.

## C HashTag Processing

Users put HashTag (#) prior to interesting phrases at any place: beginning, middle, or end of the tweet. HashTags help in classifying and searching the relevant tweets. So, HashTag processing helps us to find the user interest. Phrases attached with HashTags are not space separated. To obtain information from HashTag phrases, we separate HashTags phrases into capitalized letters and numeric values.

## D Remove Special Symbols

Combination of special symbols in tweet creates smiley, emoji and emoticons. These artistic features help in sentiment analysis of tweet, but are less important for event extraction. Such special characters are removed from further processing.

## E Spelling Correction

To cope with wrong spelling and short forms of words, used in user generated texts, the following correction methods are adopted:

- **Process Repeated Letters:** If a given character or a sequence of characters are repeated more than two times, they are replaced with the twice text constructs e.g. “*hahaha*” is replaced with “*haha*” and “*helllo*” is replaced with “*hello*”.
- **Lingo2Word:** Lingo2Word<sup>3</sup> is dedicated to the deactivation of text messaging, Internet shorthand language, and translate messages in ordinary English. For example, the word “*bz*” is replaced with “*because*” whereas “*hv*” is replaced with “*have*”.

---

<sup>3</sup><http://www.lingo2word.com/>

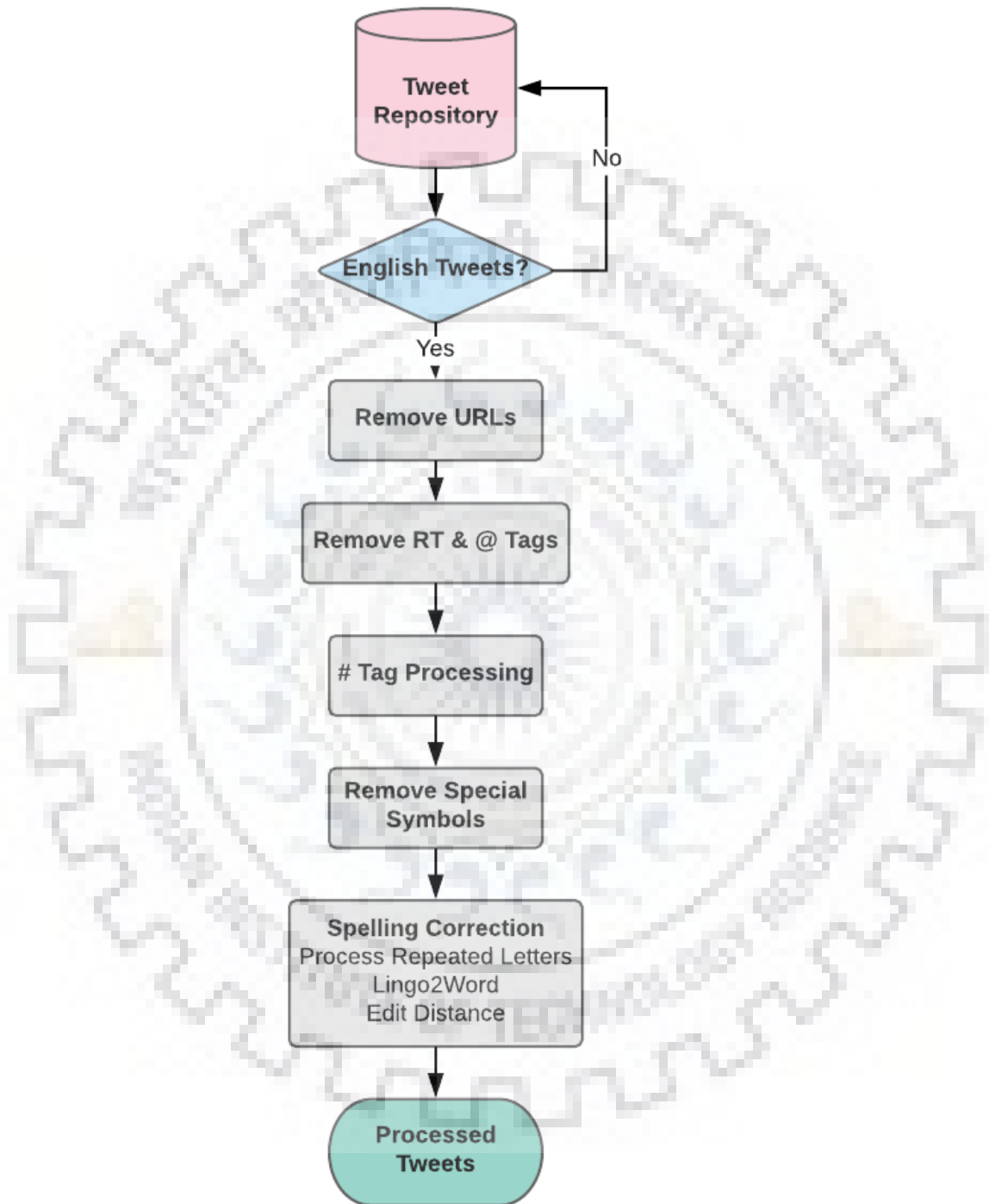


Figure 5.2: Steps to process raw tweets.

- **Edit Distance:** WordNet Dictionary<sup>4</sup> and frequency of words in tweet dataset are considered as a repository to find a correct word using edit distance method [102]. In our experiments, in order to obtain correct words, we perform maximum two edit operations among three types of edit operation: insertion of character, deletion of character, and replacement of a character. For example, the word “*rast*” is replaced with “*rest*” with edit distance one and “*replacement of a character*” edit operation.

## 5.2.2 Tweet Embedding

In this section, we describe the procedure for obtaining the vectors of processed tweets. These vectors are the combination of word representations (Word2Vec) and distributed representations (Doc2Vec) of tweets. Word2Vec [86] and Doc2Vec [68] are trained to reproduce the semantic context of words and phrases respectively. An input to tweet embedding is a large corpus of processed tweets and it produces a vector space with each unique tweet assigned an identical vector in the vector space. Tweet vectors are positioned in the vector space such that tweets that share common semantic contexts are positioned close to one another.

Let  $S$  be a set of stop words, comprises  $m$  different stop words  $s_1, s_2, \dots, s_m$ . We select the tweets satisfying the following condition for further processing:

$$\{ |w| \geq 4 : w \in t_i \ \& \ w \notin S \} \quad (5.1)$$

Where  $w$  represents the tokens in tweet  $i$  without stop words. Suppose, we have processed a tweet repository  $T_p$  containing  $n$  tweets  $t_{p1}, t_{p2}, \dots, t_{pn}$ , where  $n < z$ . We form  $d$ -dimensional vectors for each word representations  $\vec{t}_w$  and distributed representation  $\vec{t}_d$ , where each dimension  $l$  represents the co-ordinates in vector space. Word vector of tweet  $i$  is calculated as summation of vectors of each token in  $i^{th}$  tweet (Equation 5.2 and Equation 5.3):

$$card_i = |t_i| \quad (5.2)$$

<sup>4</sup><http://wordnetweb.princeton.edu/perl/webwn>

$$\vec{t}_{wi} = \sum_j^{card_i} \vec{t}_{wij} \tag{5.3}$$

Where,  $card_i$  represents the cardinality of  $i^{th}$  tweet and  $\vec{t}_{wij}$  shows the word vector of  $j^{th}$  token of  $i^{th}$  tweet. Distributed representation vector of tweet  $i$  is calculated as  $\vec{t}_{di}$ . Each tweet vector  $\vec{t}_i$  of tweet  $i$  is the combination of word vector  $\vec{t}_{wi}$  and distributed vector  $\vec{t}_{di}$ .

$$\vec{t}_i = \vec{t}_{wi} \parallel \times \parallel \vec{t}_{di} \tag{5.4}$$

Where,  $\parallel \times \parallel$  represents the concatenation operator. Thus, the dimension of each tweet vector  $t_i$  is  $2d$ . The concept of t-Distributed Stochastic Neighbor Embedding (t-SNE) [76] is used to visualize the  $2d$ -dimensional tweet vectors. Figure 5.3 depicts vector space for 8 tweets ( $t_1$  to  $t_8$ ) listed in Table 5.1. From figure, we observe that semantically similar tweets have close proximity as compared to semantically different tweets.

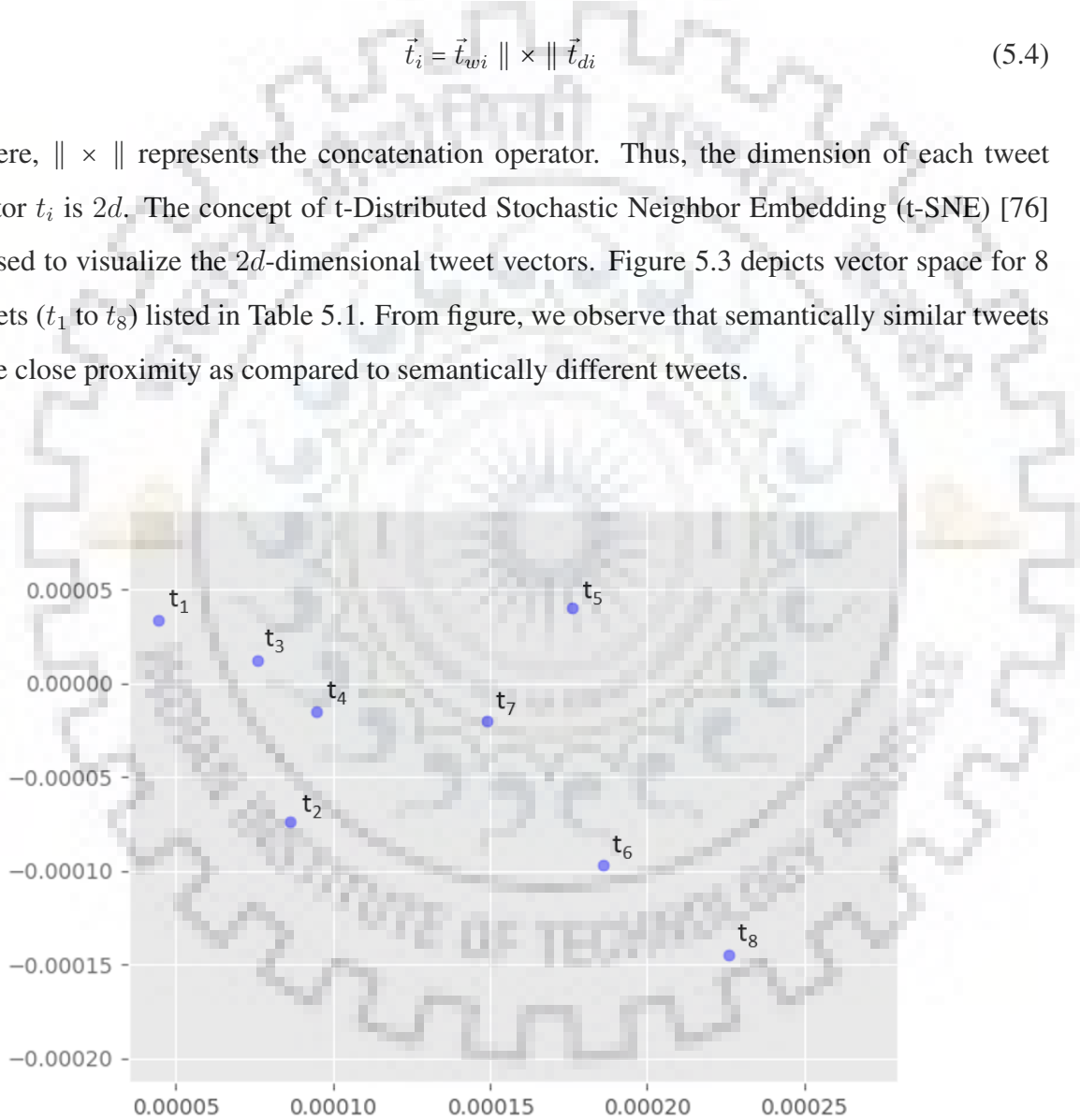


Figure 5.3: Vector Space for 8 tweets ( $t_1$  to  $t_8$ ) listed in Table 5.1.

Table 5.1: Sample of detected events.

E. No.	T. No.	Tweets
$e_1$	$t_1$	Lady Gaga Beauty Lady Gaga Fame Fluid Eau De Parfum Spray, Black eBay
	$t_2$	1 Cookie Crumbles Verano MTV 2017 Lady Gaga
	$t_3$	Now Playing Lady Gaga Just Dance feat Colby O'Donis.
	$t_4$	06:18 Lady Gaga Dancing Her Heart Out During the 2016 Super Bowl Beyonce
$e_2$	$t_5$	On this day in 2013 Dhoni become first captain and still the only Captain to win all three ICC trophies.
	$t_6$	Decision to Withdraw From ICC Still Stands Says Ruling Party South Africa.
	$t_7$	Pakistan hit nine sixes the most by any team in a final in ICC Champions Trophy history IN vs PAK CT 17
	$t_8$	The accused had also uploaded an objectionable picture of Indian cricket team online after ICC Champion Trophy Final.
$e_3$	$t_9$	My Teen Choice vote for Choice Female Artist is.
	$t_{10}$	My Teen Choice nominee for Choice Male Artist is.
	$t_{11}$	I am voting for Choice Female Artist Teen Choice
	$t_{12}$	Witness the Weirdest Moments From Katy Perry's Live Stream So Far.
$e_4$	$t_{13}$	I've never seen a president attack a former president like you do.A bit jealous.
	$t_{14}$	Infosec 17 Bored Staff the Biggest Cause of Human Error at Work Infosecurity Magazine.
	$t_{15}$	Giuliani works for the Trump Administration How is this in ANY WAY legal.
	$t_{16}$	Can you share how we would integrate a crypto payment gateway into Shopify.

### 5.2.3 Unsupervised Event Detection

To group similar type of tweets, we perform two-level clustering approach. The first level works to identify the initial training data using DBSCAN clustering and final level iteratively updates the cluster information based on self-learning-based max-margin clustering.

#### A Tweet Clustering using DBSCAN

In this subsection, our aim is to group semantically similar tweets which are  $2d$ -dimensional semantic feature vectors. As we do not know the prior specification of the number of clusters, we employ a density-based clustering algorithm (DBSCAN). Clusters in DBSCAN are defined as density connected sets and it requires two parameters:  $\epsilon$  (eps) and the minimum number of points (tweet vectors) required to form a cluster (minPts) [38]. DBSCAN clusters the tweet vectors with following conditions:

$\epsilon$ -Neighborhood: Points within a radius of  $\epsilon$  from a point.

$$N_\epsilon(p) : \{ q \mid d(p, q) \leq \epsilon \} \quad (5.5)$$

Cluster forms when  $\epsilon$ -Neighborhood of a point contains at least MinPts of a point.

$$| N_\epsilon(p) | \geq MinPts \quad (5.6)$$

By following the above conditions mentioned in Equation 5.5 and Equation 5.6, DBSCAN clustering algorithm clusters the tweet vectors and outputs the list of clusters.

#### B Self-learning-based max-margin clustering

With the influence of Active learning [127] and multi-class SVM [31] concept, we employ a novel self-learning-based max-margin clustering approach to train and predict the refined cluster label of each tweet vector. Self-learning-based max-margin clustering is an unsupervised machine learning algorithm, which clusters the  $n$ -dimensional vectors, where each dimension is the feature value. Self-learning-based max-margin clustering clusters the vector points based on hyperplanes plotted with the help of training data. Vector points near to hyperplane are known as support vectors.

To find the right hyperplane, the distances between support vectors and hyper-plane should be maximized. For linear max-margin clustering, hyperplane is represented as Equation 5.7:

$$(\vec{w}_0 \cdot \vec{t}) + \vec{b} = 0 \quad : \quad \vec{w}_0 \in R^N, \vec{b} \in R \quad (5.7)$$

Where  $\vec{w}_0$  is weight vector,  $\vec{t}$  is tweet vector, and  $\vec{b}$  is bias between hyperplane and support vector. The distance between the a vector point ( $\vec{t}_i$ ) and hyperplane is calculated as  $d(\vec{t}_i)$  in Equation 5.8.

$$d(\vec{t}_i) = \frac{(\vec{w}_0 \cdot \vec{t}_i) + \vec{b}}{\|\vec{w}_0\|} \quad (5.8)$$

Here, the clusters obtained from tweet clustering phase are considered as prior knowledge to train the learner. Self-learning-based max-margin clustering consists of following steps:

- **Initialization:** k-nearest points of each cluster centroid, obtained from DBSCAN clustering, are selected to build initial learner.
- **Convergence:** Hyperplanes are iteratively modified based on selecting k-random vector points of each cluster till there is a slight change in hyperplanes that does not affect the cluster points.
- After convergence, the remaining tweet vectors are classified based on updated hyperplanes.

Algorithm 10 shows the step by step procedure to find the refined clusters of tweet vectors using self-learning-based max-margin clustering. An input to the max-margin clustering is the output of DBSCAN clustering algorithm that is the list of clusters and each cluster contains the list of tweet vectors. To find the hyperplanes of self-learning-based max-margin clustering, initially, we calculate the centroid ( $Cent_i$ ) of each cluster ( $C_i$ ) (lines 3-5). Further k-nearest points to each cluster centroid are extracted (lines 6-11). ( $\delta_i$ ) contains the list of pairs of tweet vector point and its cluster number.  $\delta$  is considered as training data to draw hyperplanes (line 12). Hyperplanes are updated in iterative fashion till the convergence by

---

**Algorithm 10:** Self-learning-based Max-margin Clustering.

---

**Input** : List of Clusters ( $C$ ) : ( $C_i$ )  $\implies$  List of  $X_i$

$X_i$  : Tweet Vector of cluster  $i$

**Output:** Cluster Labels ( $L \implies$  List of pairs  $\langle A, B \rangle$ )

$A$  : Tweet Vector Point;

$B$  : Refined Cluster Label

```

1  $L \leftarrow \phi$ 
2  $\delta \leftarrow \phi$   $\triangleright \langle \text{tweet vector}, \text{cluster label} \rangle$ 
3 foreach  $C_i \in C$  do
4   |  $Cen_i \leftarrow$  Calculate Centroid ( $C_i$ )
5 end
6 foreach  $C_i \in C$  do
7   |  $\delta_{points} \leftarrow$  k-nearest points( $C_i, Cen_i$ )
8   |  $C_i \leftarrow C_i.$ Remove ( $\delta_{points}$ )
9   |  $\delta_i \leftarrow \langle \delta_{points}, i \rangle$ 
10  |  $\delta \leftarrow \delta \cup \delta_i$ 
11 end
12 Classifier  $\leftarrow$  Max-margin Clustering ( $\delta$ )
13 while not converged do
14   | foreach  $C_i \in C$  do
15     |  $R_{point} \leftarrow$  k-random tweet vector points
16     |  $C_i \leftarrow C_i.$ Remove ( $R_{point}$ )
17     |  $l \leftarrow$  Predict ( Classifier,  $R_{point}$  )
18     |  $L \leftarrow L \cup l$ 
19     | Classifier  $\leftarrow$  Max-margin Clustering ( $L$ )
20   | end
21 end
22 foreach  $C_i \in C$  do
23   | foreach  $Vector_{point} \in C_i$  do
24     |  $l \leftarrow$  Predict ( Classifier,  $Vector_{point}$  )
25     |  $L \leftarrow L \cup l$ 
26   | end
27 end

```

---

predicting the cluster label of selected k-random points of each cluster (lines 13-21). The rest of the tweet vectors of each cluster predict its refined cluster label based on updated hyperplanes (lines 22-27). The output of self-learning-based max-margin clustering( $L$ ) is refined clusters of tweet points in the form of pairs of tweet vector and its cluster number.



Table 5.2: Sample of raw tweets.

S. No.	Raw Tweets
$t_1$	#LadyGaga #Beauty Lady Gaga Fame Fluid Eau De Parfum Spray, Black <a href="https://t.co/KFqG45u49x">https://t.co/KFqG45u49x</a> #eBay
$t_2$	India vs Pakistan in Last 7 ICC events
$t_3$	RT @KelemenCari: According to Loretta Lynch: FBI is now FBM - Federal Bureau of Matters
$t_4$	India out to 5/1 after today's defeat to Sri Lanka #ICCTrophy
$t_5$	BeeSchilling #VeranoMTV2017 Lady Gaga
$t_6$	#Gold off lows but remains under pressure; UK election eyed <a href="https://t.co/kFYDNG4kjO">https://t.co/kFYDNG4kjO</a>
$t_7$	RT @S4IFF: DJ KHALED FT RIHANNA AND BRYSON TILLER <a href="https://t.co/IHsyBbjtbc">https://t.co/IHsyBbjtbc</a>
$t_8$	Adhering to the rules offers you a sense of safety, allowing y... More for Capricorn <a href="https://t.co/IHRZqOjI4N">https://t.co/IHRZqOjI4N</a>
$t_9$	Labeling images for a dataset is weirdly therapeutic. #deeplearning #machinelearning
$t_{10}$	#ArtificialIntelligence will automate work not entire occupations#AI #machinelearning.

## 5.3 Experiments

In this section, we describe the experimental evaluation performed based on the proposed framework. Note that, there is no ground-truth associated with the tweets and we rely on manually annotating a subset of tweets for assessing the performance of the clustering system.

### 5.3.1 Data Collection and Statistics

The tweet repository for our experiments is built by capturing 1000 streaming tweets every 5 minutes for one month. We randomly collected tweets during the period of popular events: ICC Championship trophy, MTV lady gaga in June 2017 using publicly available tweet streaming API<sup>5</sup>. We collected total 65,86,913 tweets over the one-month duration, out of which 21,48,360 tweets are in English language. Table 5.2 shows the sample raw tweets that depicts each of the aforementioned events and there are some of the tweets from a completely different context. Figure 5.4 shows the distribution of English tweets reported each day of the month of June 2017. After pre-processing, we got total 61,115 tweets, which are considered for further processing.

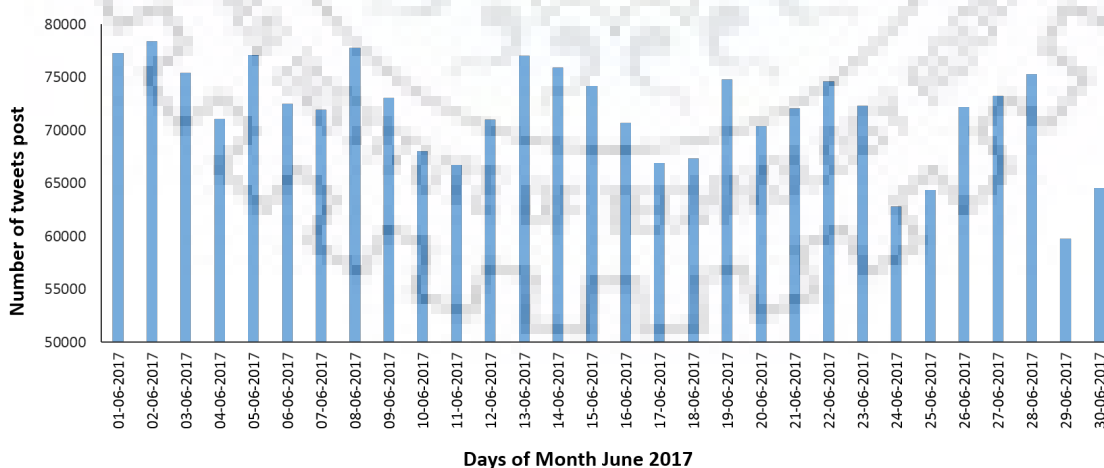


Figure 5.4: Tweets posted against days of June 2017.

<sup>5</sup><https://pypi.python.org/pypi/twitter>

### 5.3.2 Experimental Design

For the sake of performance assessment, we use: (i) precision measure which denotes the fraction of true positive samples given the classified samples, (ii) Silhouette score [104], and (iii) Calinski-Harabasz score [21]. The Calinski-Harabasz index examines the cluster viability with respect to between clusters dispersion mean and the within-cluster dispersion [21]. Silhouette index evaluates the clustering metrics with respect to the pairwise difference of between and within cluster distances [104].

After pre-processing, tweets are further processed for tweet embedding. For tweet embedding, we consider  $d = 100$  for word representations and distributed representations of tweets. So, each tweet is represented as 200-dimensional vector. The performance of our system can be influenced by various parameter values such as the value of  $\epsilon$ : maximum radius of the neighborhood of a tweet vector point and  $\gamma$ : used as a similarity measure between two tweet vector points. For our experiments, we empirically choose the values of  $\epsilon = 0.05$ ,  $\gamma = 0.8$  and  $MinPts = 10$ .

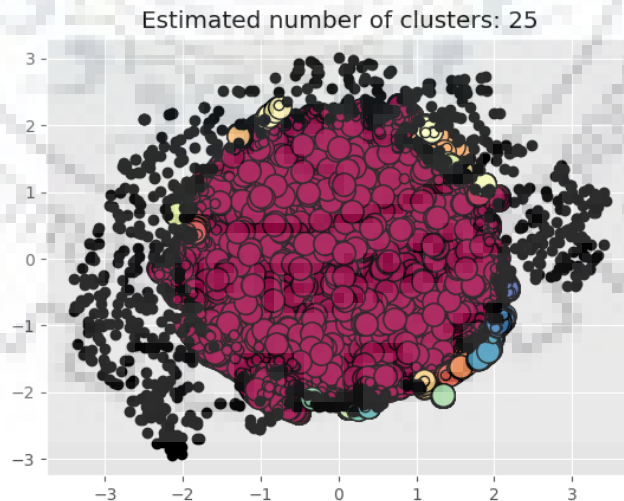
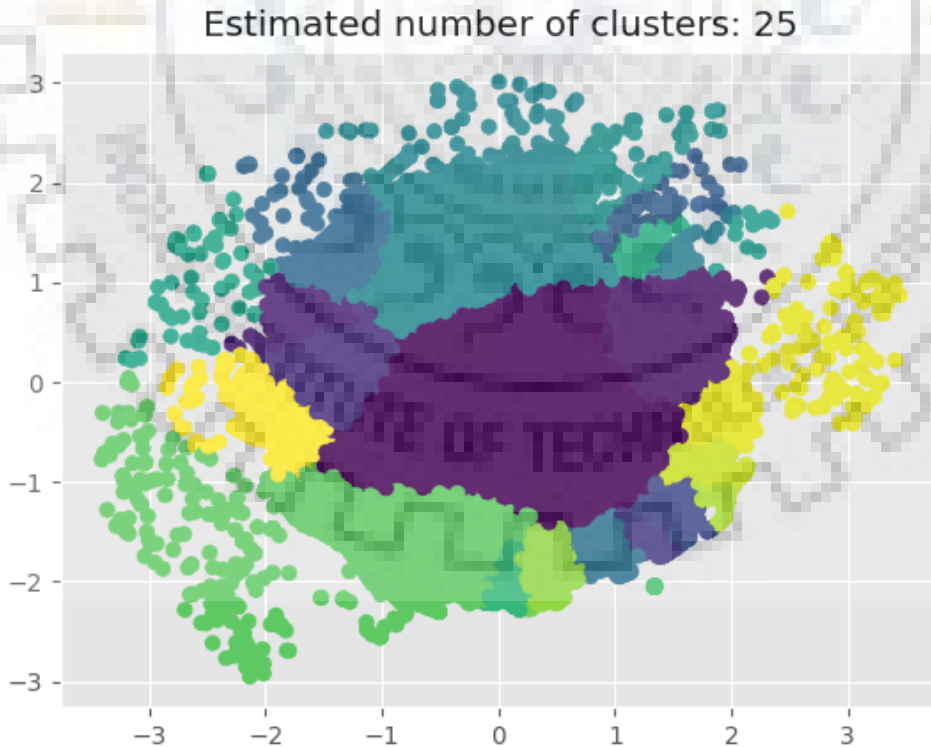


Figure 5.5: Clusters of 61, 115 tweets using DBSCAN clustering algorithm against  $\epsilon = 0.05$  and  $MinPts = 10$

### 5.3.3 Event Detection

As mention in Section 5.2.3, Event detection is two level clustering approach. First, we use DBSCAN clustering to process tweets. Figure 5.5 shows the clusters of tweets obtained against  $\epsilon = 0.05$  and  $MinPts = 10$  using DBSCAN clustering algorithm. From the figure, we observe that points in black color are noisy tweet vector points and the maximum number of tweets lie in a cluster (red color). The subsequent self-learning-based clustering stage is capable of alleviating such errors. Our proposed method also deals with noisy tweet vector points.

Figure 5.6 depicts the clusters obtained against the different values of  $\epsilon$  and fixed value of  $Minpts$  using self-learning-based max-margin clustering. In Figure 5.6(a), the tweet vectors are grouped in 25 clusters. Subsequent Figure 5.7 plots the precision (Figure 5.7(a)) and number of detected events or clusters (Figure 5.7(b)) of tweet clustering respectively, against the  $\epsilon$  values which vary between 0.01 to 0.09. From figure 5.7(a), we observe that the



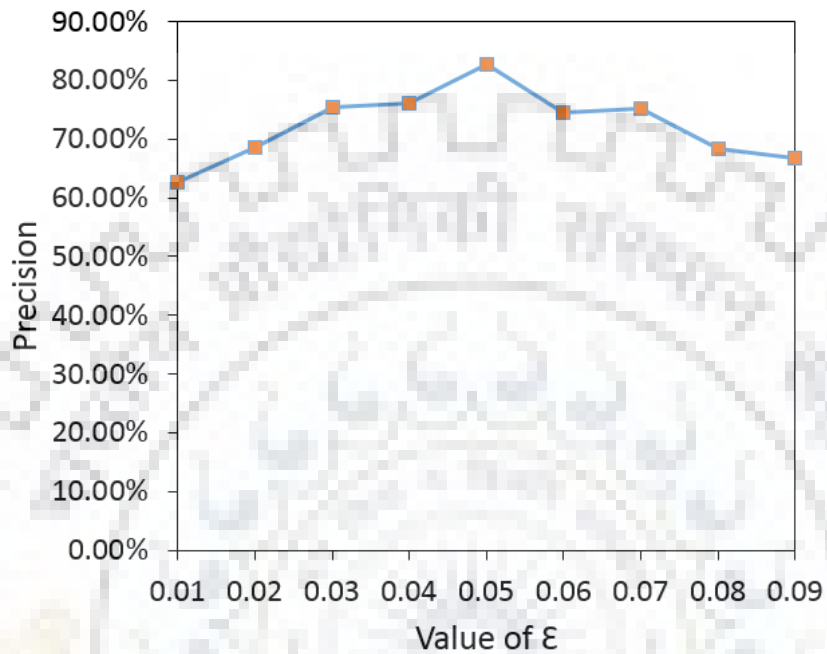
(a) Estimated Cluster against  $\epsilon = .05$



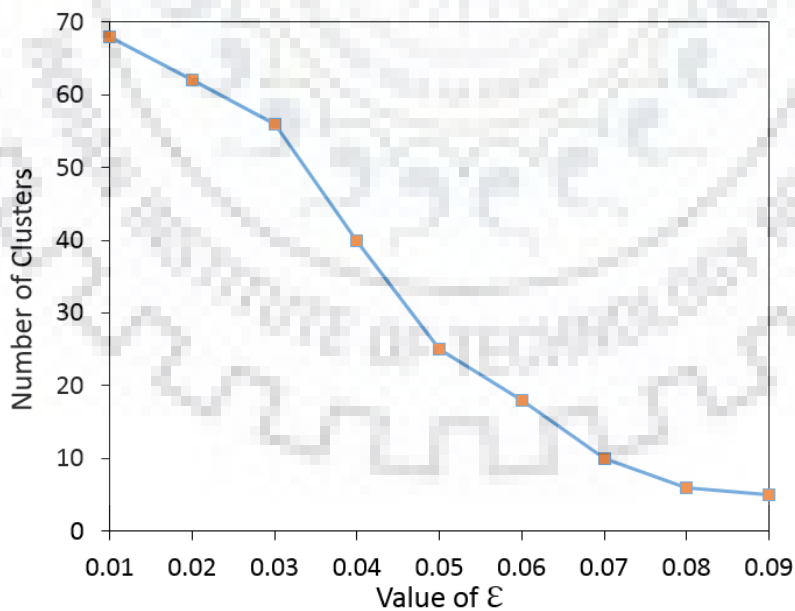
Figure 5.6: Clusters of 61, 115 tweets against the different  $\epsilon$  values.

### 5.3 Experiments

tweet clustering precision for the  $\epsilon = 0.05$  is high as compared to other values. By observing Figure 5.6 and Figure 5.7, it can be inferred that the value of  $\epsilon$  is inversely related to the number of obtained clusters. There is a high impact of  $\epsilon$  value on event detection.



(a) Precision



(b) Detected events

Figure 5.7: Precision and Number of detected events against the various  $\epsilon$  values.

We compared our proposed method with DBSCAN clustering without max-margin clustering, k-means clustering, and web-scale k-means clustering given its simplicity and good performance in handling large datasets, in respective of Calinski-Harabasz index, and Silhouette index. Based on the measures, we find that the proposed strategy is comparable to the popular techniques from the literature, but without the requirement of costly annotations. Table 5.3 depicts the detected number of clusters, Calinski-Harabasz index, and Silhouette index for the aforementioned existing approaches and proposed approach. According to Table 5.3, the score of Calinski-Harabasz index for self-learning-based max-margin clustering is high as compared to other approaches and Silhouette score is near to +1 for self-learning-based max-margin clustering in comparison to DBSCAN clustering, k-means clustering, and web-scale k-means clustering method which reflects that the clusters are dense and well separated.

Table 5.3: Comparison of the self-learning-based max-margin clustering with the other existing clustering techniques against the considered performance metrics.

<b>Method</b>	<b>Self-learning based Max-margin Clustering</b>	<b>k-means Clustering</b>	<b>DBSCAN Clustering</b>	<b>Web-Scale k-means Clustering</b>
<b>No. of clusters</b>	25	25	25	25
<b>Calinski-Harabasz index</b>	2401.005	913.699	92.526	891.231
<b>Silhouette Coefficient</b>	0.456	0.105	0.091	0.099

The samples of detected events with their semi-processed tweets are shown in Table 5.1 and  $e_i$  represents the cluster of an event  $i$ . In Table 5.1, sample tweets of 4 events are shown. Tweets from  $t_1$  to  $t_4$  are related to a popular event “*MTV Lady Gaga 2017*”, and tweets from  $t_5$  to  $t_8$  are related to another popular event “*ICC Championship Trophy 2017*”. Therefore, self-learning-based max-margin clustering group the semantically similar tweets to form an event in a very efficient manner. This proposed approach also detects the outliers and groups them to form a cluster as a useful information in some other way, but these tweets are semantically different.

## 5.4 Summary

In this chapter, we explained a novel self-learning-based max-margin clustering approach to detect events from tweets. This method comprises four stages. The first stage is based on a collection of streaming tweets. Subsequently, we perform the pre-processing task to cope with noisy nature of tweets. Further stage learns the semantic context of tweets in terms of word embeddings. In the final stage, a novel self-learning-based max-margin clustering method is used to cluster the semantically similar tweets. Our experiments show the effectiveness of this proposed approach.





## Chapter 6

# Conclusion and Future Works

In this thesis, we extracted two different type of events i.e. Named events (specific name of events) from news headlines and events from tweets. We also extracted event associated information i.e. type of named events, categories to which they belong, popular durations, involved person entities, location entities, organization entities, and related key-phrases from news media. Experiments and evaluation of the proposed methods have been done on our collected datasets (News headline and Twitter dataset.).

In the first work, mentioned in Chapter 3, named events are extracted from headline dataset along with type, categories, popular durations, and support. Headline dataset contains news headlines, headline URLs, and publication time of headlines. For named events, we used a filter-and-refine-based approach. Filter step uses prominent features of headlines to extract key-phrases. Refine step uses patterns to extract named events from key-phrases. We use URL information to find out the category of the named events. For popular durations, publication time of headlines are processed and based on extracted popular durations, type of named events is decided.

In the second work, mentioned in Chapter 4, we extracted key-phrases from news media for a given query. The extracted key-phrases are labeled as person entities, location entities, organization entities using NLP techniques. For key-phrase extraction, we use syntactical and linguistic features of the text. The extracted key-phrases are classified into novel, active, and emerged categories.

---

In the third work, mentioned in chapter 5, tweets are analyzed and processed to extract events in unsupervised setup. In this method, tweets are clustered based on their semantic context. A series of refinement steps are used to cope with the noisy nature of tweets. Then word embedding-based approach is used to find the semantic context of tweets. To group semantically similar tweets, we use an iterative method based on max-margin clustering, which uses the idea of SVM in unsupervised manner.

## Future Works

Based on this thesis, potential research directions for future work are:

- The proposed  $NE^2$  system extracts the named events from news media. This method can be used to obtain the reaction about named events on Twitter media.
- To improve the quality of extracted named events, the problem of named event disambiguation can be addressed.
- The knowledge-driven named event extraction method is described in this thesis. The concept of RNN or deep learning methods can be used to extract named events from the news media and social media.
- Clusters obtained using self-learning-based max-margin clustering can be classified to the categories or labels can be provided to each cluster.
- To compute tweet cluster overlapping, TF-IDF weights can be included.
- There is the scope to improve the key-phrase extraction method (mentioned in Chapter 4) in terms of running time.

# Appendix A

## News Recommendation System

With the advancement of web and internet, now there are plenty of online news sources available, and their number and popularity are on a rise. If users want to read or explore articles related to any subject, they swamped with available news sources and browse many of them to read the articles. Thus, navigating the online user to the right news source is an interesting research problem.

News sources publish news in a wide variety of news categories. Some of them are generic and others are dynamic. The existing efforts in organizing news sources into various news categories are in the form of Yahoo Directory<sup>1</sup> and Dmoz which aim to categorize any website by tagging them manually. Since, these solutions rely on human tagging, they are prone to errors on the part of humans, providing limited coverage for the news sources, and limiting the scale. Moreover, the news categories employed in these systems tend to be static and fixed instead of evolving. So, to cope with the aforementioned issues, we need an intelligent way to build the repository of news category or news URLs that can help the user to fulfil their needs in an effective way. News sources report news in different categories and cover different aspects of the same event in different ways. For example, Ebola news would come under “*Health*” category in African news sources while it could be

---

The content of this appendix is presented in the paper:  
Swati Gupta, Sagun Sodhani, Dhaval Patel and Biplab Banerjee, “*News Category Network based Approach for News Source Recommendations*”. In Proceedings of the 7<sup>th</sup> International Conference on Advances in Computing, Communications and Informatics (ICACCI’18), Bangalore, India.

<sup>1</sup><http://dir.yahoo.com>

---

put under “*International*” category in Indian news sources. Moreover, a news category may be associated with other categories, such as “*Cricket*” and “*Sports*” are related. Generally, news categories are either generic or time specific. For example, “*Business*” and “*Sports*” are examples of generic news category, whereas “*ICC Champions Trophy 2017*” and “*Delhi Earthquake*” are examples of time-specific news category. In this work, we aim to arrange news sources in some order, in some categories so that it becomes easier to search across various news sources and it can deal with dynamism nature of news media. This way, if a user wants to track news related to say “*Cricket*”, he can look out for news sources related to cricket domain only and not for all sports domains thus saving his time and resources.

This work presents a novel and automated solution to the problem of news source aggregation by creating a news category network. Services like GDELT<sup>2</sup> and MediaCloud<sup>3</sup> continuously track a large number of news articles across the world. We use GDELT to obtain an exhaustive list of online news sources. Then, we develop an algorithm to discover the news categories from each of these online news sources along with the news category URLs. We can extract useful information from the news category URLs itself and build an external knowledge base and use this knowledge to build a large news category network. The technical contribution of this work includes:

- We have proposed a novel approach for automatically building a news category network from news category URLs extracted from a very large number of news sources (11370). The approach suggested in [115] considers only static categories and in [35] only 3000 news sources have been considered.
- We have extended the scope of Never-Ending Learning Systems [24] by using its concept to the domain of news data and using it to generate knowledge from news category URLs. As per our knowledge, this is the first application of Never Ending Learning Paradigm in the news domain.
- We are considering three specific parameters for news source ranking while [35] considered only one parameter - namely, article freshness. The method described in [4]

---

<sup>2</sup><https://www.gdeltproject.org/>

<sup>3</sup><http://mediacloud.org>

uses content-based and social network factors and [42] ranks URLs by combining the classification, clustering, and categorization of URLs.

## A.1 Proposed Approach

In this section, we propose an approach to automatically discover news category URLs from news sources and then build a news category network using these URLs. We observed that the static (i.e., permanent) news categories are present on the home page of the news website. Other than the links to the news category URLs, the home page of a news source contains news articles, advertisements, and links to other websites (e.g., link to Facebook page). Static news category URLs like “Sports”, “Politics”, etc. are always present on news source websites. On the other hand, the dynamic categories are not always prominent on the homepage itself. While major events like “General Elections” get prominent coverage on the homepage, not all dynamic categories get listed there. Our aim is to analyze all the URLs present on news source websites and extract all the news categories - static and dynamic.

### A.1.1 Overall Architecture

Figure A.1 describes the overall architecture of the proposed system. We download a set of news source URLs from GDELT and feed it into our system which extracts news category URLs. Once, news categories and news category URLs are discovered, we proceed to identify two types of relationships: (i) the relationship among news categories, and (ii) the relationship between news categories and news category URLs. Using these two types of relationships and news categories, we build a news category network. Finally, we rank news sources in each category using parameters.

Now, when a user fires a query, for example: “tech”, we forward the query to the category network to find the related categories like mobile-tech. Primarily, we prefer to select a node in a category network that exactly matches the user query. In case, we do not find the matching node, then the node that contains the query string is selected to further processing. Once, the node is selected, we return the top news sources from the selected categories.

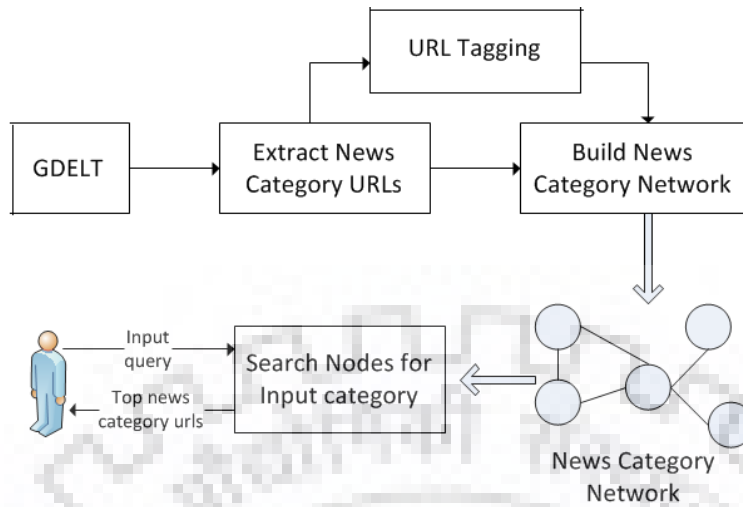


Figure A.1: Overall architecture of the system.

### A.1.2 Preliminaries

Given a URL  $w$ , obtained from the news sources, we break it down into different components such as domain, root, path, and article name. To explain these components, we use the following two URLs as examples.

$w_1$ : <https://sports.cnn.com/cricket/newArticle.aspx>

$w_2$ : <https://htimes.com/AssamElections/article1.aspx>

**Domain:** Domain name is one of the components inside a URL. The most common domains are “.com”, “.in”, “.au”, etc. Sometimes we have multi-words domains as well. For example “.co.uk”.

**Root:** Root reflects the name of the newspaper and is generally succeeded by domain name. For above examples, the root would be “cnn” and “htimes” for  $w_1$  and  $w_2$  respectively.

**Path:** All single word strings, other than certain pre-marked strings (Table A.1), that appears between the slashes (“/”) and after the domain or before the root are considered as path. In the first URL  $w_1$ , “sports” and “cricket” are the paths. In the second URL  $w_2$ , “AssamElections” is a path. We ignore the string that is in form of a date. Table A.1 shows the manually prepared list of words that should not be considered as a path.

**Article Name:** Article name comes after all the path strings and is followed by a file extension. For example, “newArticle.aspx” and “article1.aspx”.

### A.1.3 URL Tagging

Given a URL  $w$ , the process of obtaining all of its components, i.e., domain, root, path, and ArticleName, is known as URL tagging. This process goes as following: given a news URL  $w$ , we first remove the “http://” or “https://” part from the URL. Then, we use the list of pre-identified domains to tag the Domain in  $w$ . The immediate preceding string of the Domain is tagged as a Root. We split the remaining untagged URL at “/” and refer the last part as ArticleName. The remaining URL is split at “/” to obtain the Path strings.

### A.1.4 Extract News Category URLs

This section explains the process of extracting the news categories and news category URLs from a given news source  $S$ . We first set up a web crawler to crawl all the links (i.e., URLs) presents on the webpage of  $S$ . Let  $uLinks_S$  be the list of all URLs crawled from the news source  $S$ . As mentioned in Section A.1, the home page of a news source contains common URLs, static and dynamic category URLs. In order to separate out the news category URLs from the others, we again run a crawler for each URL in  $uLinks_S$  and discover the URLs present on those web pages. Our intuition for the second crawling is that we are expecting to encounter the news category URLs more frequently than the others URLs.

Table A.1: pre-marked strings.

www	http	https	tags	register
feedback	privacy	rss feeds	term-service	user
advertise	about us	contact us	password	press-release
subscribe	sitemap	copyright	disclaimer	archive

The discovered URLs are later added to the  $uLinks_S$ . At this stage,  $uLinks_S$  may contain some URLs multiple times. Thus, we count the frequency of each URL in  $uLinks_S$  and filter out URLs with the frequency less than the user supplied threshold  $k$ . We keep the value of  $k$  to be 5. Higher value means only very popular categories will be captured while for  $k=1$  each news article becomes a category. The remaining URLs in  $uLinks_S$  are news category



URLs. The abovementioned process is repeated for all the news sources present on GDELT. Let,  $catLinks$  be the list of URLs belonging to unique news category.

### A.1.5 Building News Category Network

News Category Network captures two types of relationships: (i) the relationship among news categories, and (ii) the relationship between news categories and news category URLs. Initially, news category network is an empty graph. Given a news category URL  $w$  from  $catLinks$ , we first perform URL tagging of  $w$ . In our case, each string in  $w.path$  is considered as one category. Next, we create a new node, having label  $c$ , for a category  $c \in w.path$ , if news category network does not contain any node with label  $c$ . In this manner, we process each news category URL in  $catLinks$ . Note that each node in the network maintains the news category URLs such that news category URLs has a category that matches with the node label. Similarly, two nodes,  $n_1$  and  $n_2$  in the network are connected, if there is a news category URL  $w$  that contains label of both  $n_1$  and  $n_2$  as a category in  $w$ .

**Example.** Figure A.2, shows a small category network developed using five news category URLs given in Table A.2. In Figure A.2 the weight of a vertex denotes the number of news sources associated with this category while the weight of an edge denotes the number of times two categories occur together. Here, you can see that concept ‘‘Business’’ and ‘‘Economy’’ occur together two times which shows more strong relationship than other concepts.

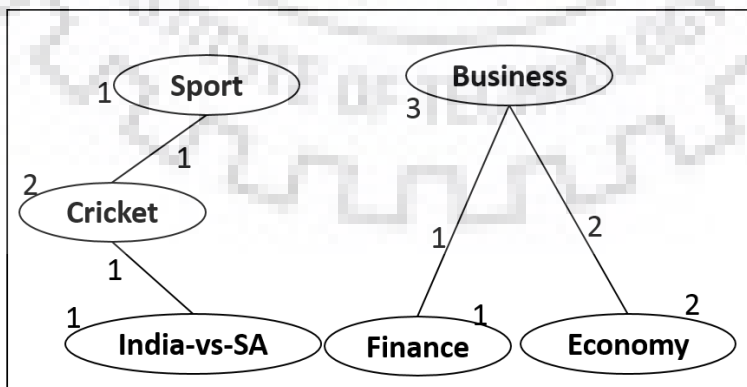


Figure A.2: Example of small news category network.



Table A.2: Example of news source category url.

---

```

http://www.mirror.co.uk/sport/cricket
http://thenational.ae/business/economy/
http://www.nytimes.com/business/finance
http://www.thehindu.com/business/Economy/
http://www.ibtimes.co.in/cricket/India-vs-SA

```

---

## A.2 News Source Ranking

A user is always interested in visiting the popular news sources for a given category, we also present a method to rank the different news sources that publish news in the same category. To rank news sources for a given category, we have considered two factors: (i) Global importance, and (ii) Local importance. The global importance is factored for using website traffic and social media popularity of the news source. The local importance is recorded based on the freshness of the news articles published by the news source in the category of interest. We explain these parameters as follows:

### A.2.1 Traffic-based Website Importance

This factor calculates website popularity using Alexa Rank <sup>4</sup> and Google PageRank [95]. Alexa Rank ranks the website based on the traffic received in last three month. Google Page Rank algorithm assigns a score between 0 and 10 based on the total number of links and number of quality links on a website. Traffic-based Website Importance of news category URL  $w$  is calculated as:

$$WebImp_w = PageRank_w - AlexRank_w + 1 \quad (A.1)$$

Where  $PageRank_w$  is Google Page Rank, and  $AlexRank_w$  is normalized Alexa Rank between 0 to 10.

### A.2.2 Social Media-based Popularity

With the success of Online Social Networks (OSNs) such as Facebook, Twitter, etc., news media started using OSN to publish their contents. In our case, the social media popularity of a news source, denoted as  $OSNImp_w$ , is calculated by extracting the followers count on

---

<sup>4</sup><http://www.alexa.com>

Twitter page and likes on the Facebook page. We normalize both the parameters on a scale of 0 to 10. The value of  $OSNImp_w$  is calculated by taking the average of both the normalized parameters.

#### A.2.3 Category-wise Article Freshness Score

This measure identifies the average number of fresh articles generated in a week by a particular news category URL. It is denoted by  $Fresh_w^c$ . The freshness parameter is normalized on a scale of 0 to 10.

Now, we define the popularity  $P_w^c$  of news category url  $w$  for category  $c$  as follow:

$$P_w^c = (w_1 * WebImp_w) + (w_2 * OSNImp_w) + (w_3 * Fresh_w^c) \quad (A.2)$$

Where  $w_1$ ,  $w_2$ , and  $w_3$  are the weights assigned to different scores. In our case,  $WebImp_w$  and  $Fresh_w^c$  are given higher weight than  $OSNImp_w$ . Empirically, we set  $w_1$  and  $w_3$  to 0.4 and  $w_2$  to 0.2.

## A.3 Experiments

**News Source DataSet.** To obtain a list of news sources, we downloaded news event URLs from the GDELT project. GDELT project monitors web news from every corner of the world and captures the news events URL from thousands of news sources on daily basis. We have downloaded 1,01,81,225 news events URLs collected during April 2016 to December 2016 and extracted 11,370 news sources. After applying the approach described in Section A.1.4, we finally obtain 2,50,150 news category URLs from these 11,370 news sources.

**Web Crawler.** We have set up a distributed crawler to obtain the count of fresh articles published by a news source for a given category on a weekly basis. In particular, our crawler fetches the fresh articles for 2,50,150 news category URLs. The fresh (i.e., new) articles are recorded by requesting the news websites for each category URL at a frequency of 12 hours. We started our system on 1<sup>st</sup> April 2017 and stopped on 29<sup>th</sup> April 2017 (4 weeks). Finally, we obtain the average number of fresh articles, per week, for each news source category URL. Table A.3 shows the subset of score values, obtained for several News Category URLs.

Table A.3: Article freshness score of news category URL.

News Category URL $w$	Score
<a href="http://www.dailymail.co.uk/sport">http://www.dailymail.co.uk/sport</a>	4.244645
<a href="http://www.dailystar.co.uk/sport">http://www.dailystar.co.uk/sport</a>	4.361081
<a href="http://www.southcoasttoday.com/business">http://www.southcoasttoday.com/business</a>	3.631604
<a href="http://www.nzherald.co.nz/hockey">http://www.nzherald.co.nz/hockey</a>	0.205458
<a href="http://journaltimes.com/sports/basketball">http://journaltimes.com/sports/basketball</a>	0.084274

The score value is directly proportional to the published news. We observed that generic news categories such as “*sports*”, “*business*”, “*politics*” have higher score value than the specific one like “*basketball*” and “*hockey*”.

**Social Media Crawler.** We have used BeautifulSoup API<sup>5</sup> in Python to extract the Alexa traffic rank, Google Page Rank, Twitter followers, Facebook likes, etc. Note that, in Alexa traffic rank, the most popular website is given a rank 1, the second most popular website is given a rank 2, and so on. Thus, we first obtain the Alexa traffic rank for each news source present in our system and then normalize the score using min-max normalization on the scale of 0-10 (0 = High, 10 = Low). The popular websites are given a score near 10 by Google Page Rank Algorithm, whereas the unpopular website is given the score near to 0.

### A.3.1 Efficiency of News Category URL Extraction

In this section, we evaluate the results of proposed news category URL extraction approach. We prepared a ground truth dataset with fifty news sources and manually extracted categories from them using their category URLs. Table A.4 shows the sample of ground truth data. For example, the news source<sup>6</sup> has news category URLs like<sup>7</sup> with categories “*market*”, “*companies*” and news source<sup>8</sup> with category “*finance*”. On an average, each news source has 40 news categories in the ground truth dataset.

We compare the results obtained by our algorithm with the ground truth. For each news source in the ground truth, we compare the categories obtained by our algorithms and calculated the precision and recall for each news source. Figure A.3 reports the average precision and average recall of our proposed technique.

<sup>5</sup><https://pypi.org/project/beautifulsoup4/>

<sup>6</sup>[www.forbes.com](http://www.forbes.com)

<sup>7</sup><http://www.forbes.com/market/companies>

<sup>8</sup>[www.forbes.com/finance](http://www.forbes.com/finance)

Table A.4: Example of news source category url.

News Source	Categories
www.forbes.com	Market, Companies, Finance, etc.
nytimes.com	Politics, Business, Technology, etc.
gadgets.ndtv.com	Apple, Android, Google, etc.
www.thehindu.com	Business, Sports, Nation, Science, etc.
techcrunch.com	Technology, Startups, Mobile, Asia, etc.



Figure A.3: Precision and Recall of proposed approach.

### A.3.2 Analysis of News Categories in News Category Network

We build news category network using news category URLs obtained from all 11370 news sources. The total number of news categories obtained is 1,10,906. The top ten news categories which are present in maximum number of news sources are “sports”(≈ 4000), “business”(≈ 3000), “opinion”(≈ 3000), “category”(≈ 2000), “entertainment” (≈ 2000), “politics” (≈ 2000), “weather” (≈ 2000), “lifestyle” (≈ 1500), “health” (≈ 1500), and “education” (≈ 1500). The number next to the news category represents the number of times the news category was found.

#### Dmoz vs Extracted News Category

We use Selenium browser automation API<sup>9</sup> to send a request to Dmoz site for verifying whether a news source is categorized by Dmoz or not. We noted that only 33% of the news sources are present in the Dmoz directory. For example, Figure A.4(a) and Figure A.4(b)

<sup>9</sup><https://www.seleniumhq.org/download/>

shows the categories obtained by our method and Dmoz respectively for the home page of “The Hindu”<sup>10</sup>. We can compare that, our approach categorizes news sources into various categories. We also observe that most of the times, the Dmoz assign news sources to the “Newspaper” or “News and Media” category. Moreover, Dmoz generates on an average 10 categories per news source. Compared to this, our approach generates an average 40 categories per news source.

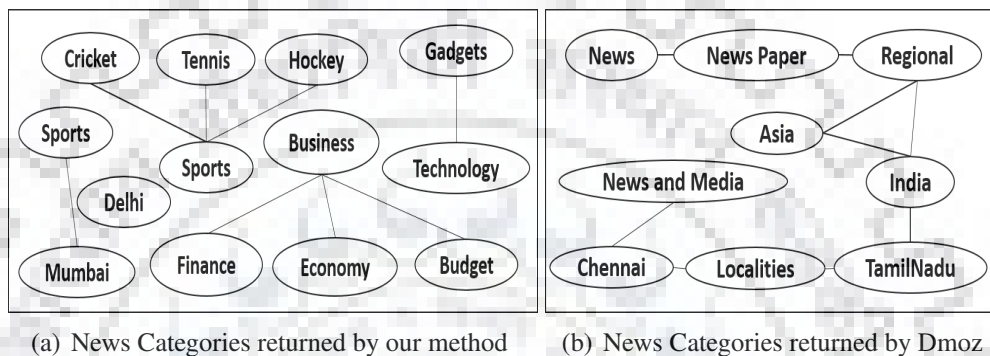


Figure A.4: News categories returned for <http://www.thehindu.com>.

### A.3.3 Analysis of News Source Ranking in News Category Network

Table A.5 shows the result of top five news category URLs obtained for “sports” category by our method. All the parameters in the Table A.5 are normalized between 0 and 10. Table A.5 shows that the social media score for category link<sup>11</sup> is low, but the other two parameters are making it suitable to be considered it in top five category URLs as the weight of other parameters are high.

We did not find any existing work which ranks a large number of news sources in all the categories. Hence, to judge the quality of our ranking, we use Google News for obtaining the news sources of top news categories. In particular, given a news category, we monitor Google News at different time intervals and obtain URL of news articles that appear in the top five result page of Google News. Table A.6 shows the top five news category URLs obtained from the Google News for sports category. As we can see, some of the results

<sup>10</sup><http://www.thehindu.com>

<sup>11</sup><http://www.dailymail.co.uk/sport>

Table A.5: Top 5 category links for “sports” category.

Source Category Link	Traffic Score	Social Media Score	Fresh Article Weekly	Ranking Score
nytimes.com/pages/sports	8.9991	5.784	5.3716	6.84388
dailymail.co.uk/ sports	6.99996	0.44912	4.05179	4.51052
reuters.com/news/ sports	7.99987	4.1749	1.1109	4.479484
time.com/sports	7.99987	2.6903	1.66348	4.4034
espn.go.com/sports	7.99997	0.8936	1.86995	4.126688

Table A.6: Results using Google News for sport Category.

News category URLs generated by Google News
mirror.co.uk/sports/football/news
businessinsider.com
irishtimes.com/sport/soccer
dailymail.co.uk/sport/football
sports.yahoo.com/blogs

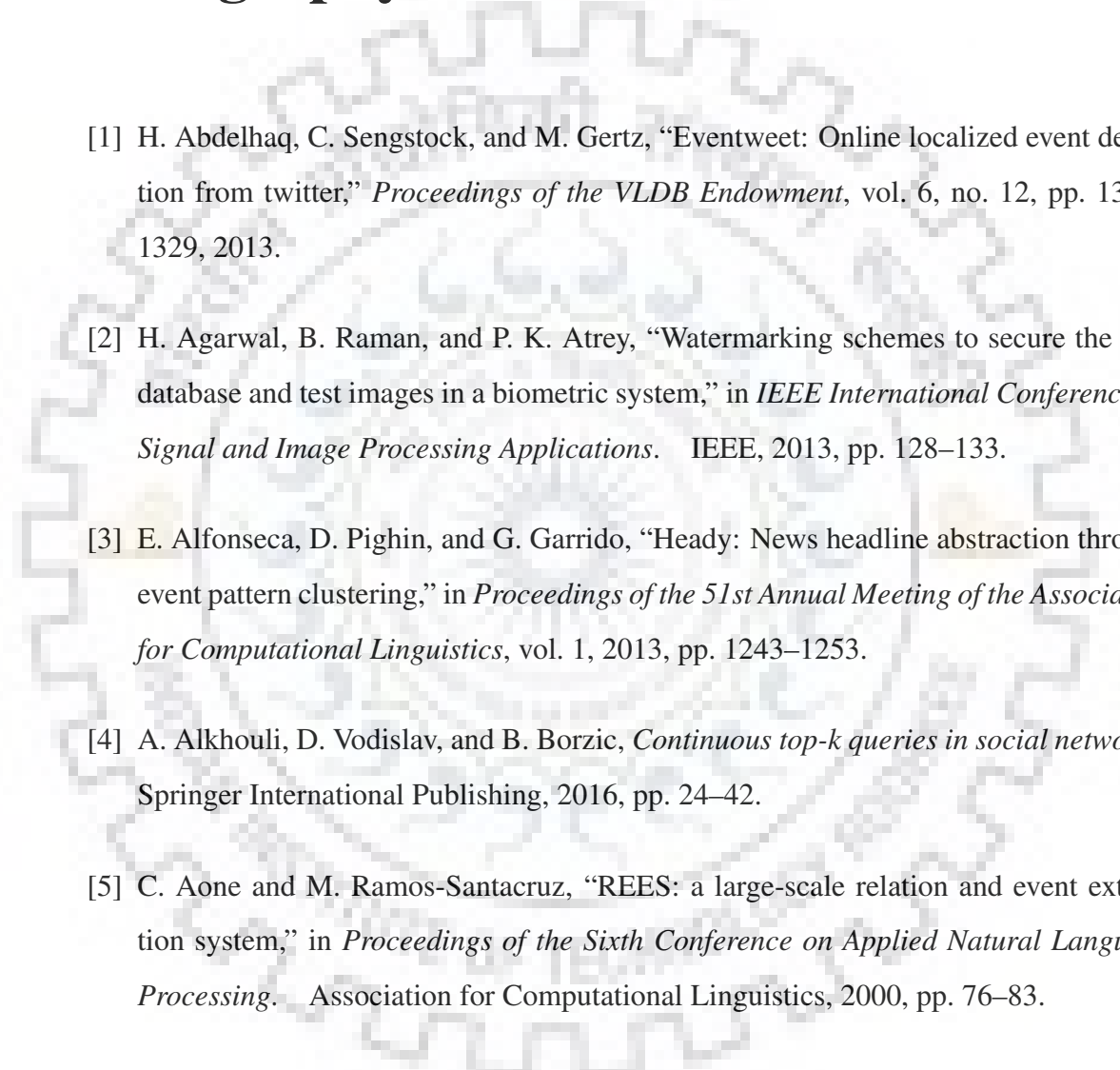
generated by Google News do not actually talk about sports primarily but get included as they published some articles about sports.

## A.4 Summary

In this work, we have proposed a novel approach of automatically building a news category network by discovering news category URLs from a large number of news sources. We have observed that a huge set of news category URLs can generate a large number of concepts. These concepts can be used as an external knowledge base for building a news category network. This network could be useful for recommending more refined concepts related to a category. We also propose an approach to rank news sources in various categories so that the news category network could be used for recommending news sources for different categories. The proposed system can be used as a dynamic system which would automatically learn changes on the websites, in rankings, and adapt to these changes accordingly.



# Bibliography

- 
- [1] H. Abdelhaq, C. Sengstock, and M. Gertz, “Eventweet: Online localized event detection from twitter,” *Proceedings of the VLDB Endowment*, vol. 6, no. 12, pp. 1326–1329, 2013.
- [2] H. Agarwal, B. Raman, and P. K. Atrey, “Watermarking schemes to secure the face database and test images in a biometric system,” in *IEEE International Conference on Signal and Image Processing Applications*. IEEE, 2013, pp. 128–133.
- [3] E. Alfonseca, D. Pighin, and G. Garrido, “Heady: News headline abstraction through event pattern clustering,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2013, pp. 1243–1253.
- [4] A. Alkhouli, D. Vodislav, and B. Borzic, *Continuous top-k queries in social networks*. Springer International Publishing, 2016, pp. 24–42.
- [5] C. Aone and M. Ramos-Santacruz, “REES: a large-scale relation and event extraction system,” in *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Association for Computational Linguistics, 2000, pp. 76–83.
- [6] E. Arendarenko and T. Kakkonen, “Ontology-based information and event extraction for business intelligence,” in *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer, 2012, pp. 89–102.
- [7] Z. Ashktorab, C. Brown, M. Nandi, and A. Culotta, “Tweedr: Mining twitter to inform disaster response.” in *ISCRAM*, 2014, pp. 354–358.

## BIBLIOGRAPHY

---

- [8] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5. IEEE, 2006, pp. V–V.
- [9] P. K. Atrey, M. S. Kankanhalli, and R. Jain, "Information assimilation framework for event detection in multimedia surveillance systems," *Multimedia Systems*, vol. 12, no. 3, pp. 239–253, 2006.
- [10] J. Barua, D. Patel, and A. K. Agrawal, "Removing noise content from online news articles," in *Proceedings of the 20th International Conference on Management of Data*. Computer Society of India, 2014, pp. 113–116.
- [11] J. Barua, D. Patel, and V. Goyal, "Tide: Template-independent discourse data extraction," in *International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 2015, pp. 149–162.
- [12] A. Behera, M. Chapman, A. G. Cohn, and D. C. Hogg, "Egocentric activity recognition using histograms of oriented pairwise relations," in *International Conference on Computer Vision Theory and Applications*, vol. 2. IEEE, 2014, pp. 22–30.
- [13] A. Behera, A. G. Cohn, and D. C. Hogg, "Real-time activity recognition by discerning qualitative relationships between randomly chosen visual features," in *Proceedings of the British Machine Vision Conference (BMVC)*. British Machine Vision Association, BMVA, 2014.
- [14] C. Best, J. Piskorski, B. Pouliquen, R. Steinberger, and H. Tanev, "Automating event extraction for the security domain," in *Intelligence and Security Informatics*. Springer, 2008, pp. 17–43.
- [15] J. Björne and T. Salakoski, "Biomedical event extraction using convolutional neural networks and dependency parsing," in *Proceedings of the BioNLP workshop*, 2018, pp. 98–108.
- [16] J. Björne, F. Ginter, S. Pyysalo, J. Tsujii, and T. Salakoski, "Complex event extraction at pubmed scale," in *Bioinformatics [ISMB]*, 2010.



- [17] J. Björne *et al.*, “Biomedical event extraction with machine learning,” Ph.D. dissertation, Turku Centre for Computer Science Dissertations, 2014.
- [18] J. Borsje, F. Hogenboom, and F. Frasincar, “Semi-automatic financial events discovery based on lexico-semantic patterns,” *International Journal of Web Engineering and Technology*, vol. 6, no. 2, pp. 115–140, 2010.
- [19] T. Brants, “Tnt: A statistical part-of-speech tagger,” in *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ser. ANLC ’00. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 224–231. [Online]. Available: <https://doi.org/10.3115/974147.974178>
- [20] Y.-G. Byun, Y.-K. Han, and T.-B. Chae, “Road extraction from high resolution satellite image using object-based road model,” *Korean Journal of Remote Sensing*, vol. 27, no. 4, pp. 421–433, 2011.
- [21] T. Caliński and J. Harabasz, “A dendrite method for cluster analysis,” *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.
- [22] J. Capdevila, J. Cerquides, J. Nin, and J. Torres, “Tweet-scan: An event discovery technique for geo-located tweets,” *Pattern Recognition Letters*, vol. 93, pp. 58–68, 2017.
- [23] P. Capet, T. Delavallade, T. Nakamura, A. Sandor, C. Tarsitano, and S. Voyatzi, “A risk assessment system with automatic extraction of event types,” in *International Conference on Intelligent Information Processing*. Springer, 2008, pp. 220–229.
- [24] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, “Toward an architecture for never-ending language learning,” in *AAAI*, vol. 5. Atlanta, 2010, p. 3.
- [25] C.-Y. Chang, Z. Teng, and Y. Zhang, “Expectation-regulated neural model for event mention extraction,” in *HLT-NAACL*, 2016, pp. 400–410.

## BIBLIOGRAPHY

---

- [26] Y. Chen, S. Liu, X. Zhang, K. Liu, and J. Zhao, “Automatically labeled data generation for large scale event extraction,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2017, pp. 409–419.
- [27] N. Chowdhury and D. Saha, “Unsupervised text document classification using neural networks,” in *Proceedings of ICSLT-O-COCOSDA-iSTRANS 2004 International Conference*, vol. 1, pp. 62–68.
- [28] N. Chowdhury, D. Saha, and K. Gupta, “English and bengali text document classification using mst of data points,” in *Journal of Special Issue on Computations in Computer Engineering, International Journal of Computer Science and System Analysis*, vol. 2, 2008, pp. 241–248.
- [29] M. Ciaramita and M. Johnson, “Supersense tagging of unknown nouns in wordnet,” in *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2003, pp. 168–175.
- [30] K. B. Cohen, K. Verspoor, H. L. Johnson, C. Roeder, P. V. Ogren, W. A. Baumgartner Jr, E. White, H. Tipney, and L. Hunter, “High-precision biological event extraction with a concept recognizer,” in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Association for Computational Linguistics, 2009, pp. 50–58.
- [31] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [32] X. Dai, S. Karimi, and C. Paris, “Medication and adverse event extraction from noisy text,” in *Proceedings of the Australasian Language Technology Association Workshop*, 2017, pp. 79–87.
- [33] M.-C. De Marneffe and C. D. Manning, “The stanford typed dependencies representation,” in *proceedings of the Workshop on Cross-framework and Cross-domain Parser Evaluation*. Association for Computational Linguistics, 2008, pp. 1–8.

- [34] E. Dede, Z. Fadika, J. Hartog, M. Govindaraju, L. Ramakrishnan, D. Gunter, and R. Canon, “Marissa: Mapreduce implementation for streaming science applications,” in *IEEE 8th International Conference on E-Science*. IEEE, 2012, pp. 1–8.
- [35] G. M. Del Corso, A. Gulli, and F. Romani, “Ranking a stream of news,” in *Proceedings of the 14th International Conference on World Wide Web*. ACM, 2005, pp. 97–106.
- [36] J. Dutkiewicz, M. Nowak, and C. Jedrzejek, “R2E: rule-based event extractor,” in *Proceedings of the RuleML 2014 Challenge and the RuleML 2014 Doctoral Consortium hosted by the 8th International Web Rule Symposium, Challenge+DC@RuleML, 2014*.
- [37] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han, “Scalable topical phrase mining from text corpora,” *Proc. VLDB Endow.*, vol. 8, no. 3, pp. 305–316, Nov. 2014.
- [38] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34, 1996, pp. 226–231.
- [39] A. Fader, S. Soderland, and O. Etzioni, “Identifying relations for open information extraction,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1535–1545.
- [40] N. Farajidavar, S. Kolozali, and P. Barnaghi, “A deep multi-view learning framework for city event extraction from twitter data streams,” *arXiv preprint arXiv:1705.09975*, 2017.
- [41] J. Ferguson, C. Lockard, D. Weld, and H. Hajishirzi, “Semi-supervised event extraction with paraphrase clusters,” in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2. Association for Computational Linguistics, 2018, pp. 359–364.

## BIBLIOGRAPHY

---

- [42] M. N. Feroz and S. Mengel, “Phishing url detection using url ranking,” in *IEEE International Congress on Big Data (BigData Congress)*. IEEE, 2015, pp. 635–638.
- [43] J. G. Fiscus and G. R. Doddington, *Topic Detection and Tracking Evaluation Overview*. Springer, 2002, pp. 17–31.
- [44] J. Foley, M. Bendersky, and V. Josifovski, “Learning to extract local events from the web,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 423–432.
- [45] A. Frias-Velazquez, P. Van Hese, A. Pižurica, and W. Philips, “Split-and-match: A bayesian framework for vehicle re-identification in road tunnels,” *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 220–233, 2015.
- [46] S. N. Ghoreishi and A. Sun, “Predicting event-relatedness of popular queries,” in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, pp. 1193–1196.
- [47] Y. Han, B. Kim, Y. Kim, and W. H. Lee, “Automatic cloud detection for high spatial resolution multi-temporal images,” *Remote Sensing Letters*, vol. 5, no. 7, pp. 601–608, 2014.
- [48] J. Hartog, R. DelValle, M. Govindaraju, and M. J. Lewis, “Configuring a mapreduce framework for performance-heterogeneous clusters,” in *IEEE International Congress on Big Data (BigData Congress)*. IEEE, 2014, pp. 120–127.
- [49] M. Hasan, M. A. Orgun, and R. Schwitter, “A survey on real-time event detection from the twitter data stream,” *Journal of Information Science*, p. 0165551517698564, 2017.
- [50] J. Hoffart, D. Milchevski, and G. Weikum, “Stics: searching with strings, things, and cats,” in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2014, pp. 1247–1248.

- [51] S.-H. Hung, C.-H. Lin, and J.-S. Hong, "Web mining for event-based commonsense knowledge using lexico-syntactic pattern matching and semantic role labeling," *Expert Systems with Applications*, vol. 37, no. 1, pp. 341–347, 2010.
- [52] W. IJntema, J. Sangers, F. Hogenboom, and F. Frasincar, "A lexico-semantic pattern language for learning ontology instances from text," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 15, pp. 37–50, 2012.
- [53] A. Ilyas, "Microfilters: Harnessing twitter for disaster management," in *Global Humanitarian Technology Conference*. IEEE, 2014, pp. 417–424.
- [54] F. Jalled and I. Voronkov, "Object detection using image processing," *arXiv preprint arXiv:1611.07791*, 2016.
- [55] K. Jang, K. Lee, G. Jang, S. Jung, M.-G. Seo, and S.-H. Myaeng, "Food hazard event extraction based on news and social media: A preliminary work," in *International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2016, pp. 466–469.
- [56] F. Jungermann and K. Morik, "Enhanced services for targeted information retrieval by event extraction and data mining," in *Proceedings of the 13th International Conference on Natural Language and Information Systems*, 2008, pp. 335–336.
- [57] N. Kanhabua, T. Ngoc Nguyen, and W. Nejdl, "Learning to detect event-related queries for web search," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 1339–1344.
- [58] Y. Keneshloo, J. Cadena, G. Korkmaz, and N. Ramakrishnan, "Detecting and forecasting domestic political crises: a graph-based approach," in *Proceedings of the 2014 ACM Conference on Web Science*. ACM, 2014, pp. 192–196.
- [59] B. N. Keshavamurthy, M. Sharma, and D. Toshniwal, "Efficient support coupled frequent pattern mining over progressive databases," *arXiv preprint arXiv:1005.5434*, 2010.

## BIBLIOGRAPHY

---

- [60] B. N. Keshavamurthy, A. M. Khan, and D. Toshniwal, "Privacy preserving association rule mining over distributed databases using genetic algorithm," *Neural Computing and Applications*, vol. 22, no. 1, pp. 351–364, 2013.
- [61] F. H. Khan, S. Bashir, and U. Qamar, "Tom: Twitter opinion mining framework using hybrid classification scheme," *Decision Support Systems*, vol. 57, pp. 245–257, 2014.
- [62] A. Košmerlj, E. Belyaeva, G. Leban, M. Grobelnik, and B. Fortuna, "Towards a complete event type taxonomy," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15 Companion. New York, NY, USA: ACM, 2015, pp. 899–902.
- [63] A. Kumar, D. Patel, and N. Jain, "Lightweight system for ne-tagged news headlines corpus creation," in *2016 IEEE International Conference on Big Data (Big Data)*, Dec 2016, pp. 3903–3912.
- [64] M. Kumar, S. Das, and S. Govil, "Analysis of stock volatility clustering using ann," *Information Resources Management Journal (IRMJ)*, vol. 28, no. 2, pp. 32–45, 2015.
- [65] F. Kunneman and A. Van den Bosch, "Open-domain extraction of future events from twitter," *Natural Language Engineering*, vol. 22, no. 5, pp. 655–686, 2016.
- [66] E. Kuzey and G. Weikum, "Evin: building a knowledge base of events," in *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014, pp. 103–106.
- [67] E. Kuzey, J. Vreeken, and G. Weikum, "A fresh look on knowledge bases: Distilling named events from news," in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*. ACM, 2014, pp. 1689–1698.
- [68] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning*, 2014, pp. 1188–1196.



- [69] G. Leban, B. Fortuna, J. Brank, and M. Grobelnik, "Event registry: learning about world events from news," in *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014, pp. 107–110.
- [70] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. ACM, 2012, pp. 155–164.
- [71] W. Liao, H. Zhang, J. Li, S. Huang, R. Wang, R. Luo, and A. Pizurica, "Fusion of spectral and spatial information for land cover classification," in *IEICE Information and Communication Technology Forum (ICTF)*, 2016.
- [72] K. W. Lim and W. Buntine, "Twitter opinion topic model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon," in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*. ACM, 2014, pp. 1319–1328.
- [73] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, "Mining quality phrases from massive text corpora," in *Proceedings of ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1729–1744.
- [74] M. Liu, Y. Liu, L. Xiang, X. Chen, and Q. Yang, "Extracting key entities and significant events from online daily news," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2008, pp. 201–209.
- [75] Z. Liu, P. Li, Y. Zheng, and M. Sun, "Clustering to find exemplar terms for keyphrase extraction," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 257–266.
- [76] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

## BIBLIOGRAPHY

---

- [77] M. Macesic, V. Jelaca, J. Niño-Castaneda, N. Prodanovic, M. Panic, A. Pizurica, V. Crnojevic, and W. Philips, “Real-time detection of traffic events using smart cameras,” in *Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, vol. 8301. International Society for Optics and Photonics, 2012, p. 83010E.
- [78] A. Majumder, A. Ekbal, and S. K. Naskar, “Biomolecular event extraction using a stacked generalization based classifier,” in *Proceedings of the 13th International Conference on Natural Language Processing*, 2016, pp. 55–64.
- [79] S. Malviya and U. S. Tiwary, “Knowledge based summarization and document generation using bayesian network,” *Procedia Computer Science*, vol. 89, pp. 333–340, 2016.
- [80] M. Mathioudakis and N. Koudas, “Twittermonitor: trend detection over the twitter stream,” in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 1155–1158.
- [81] Mausam, M. Schmitz, R. Bart, S. Soderland, O. Etzioni *et al.*, “Open language learning for information extraction,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 523–534.
- [82] S. Mazumder, B. Bishnoi, and D. Patel, “News headlines: What they can tell us?” in *Proceedings of the 6th IBM Collaborative Academia Research Exchange Conference (I-CARE) on I-CARE 2014*. ACM, 2014, pp. 1–4.
- [83] M. McCandless, E. Hatcher, and O. Gospodnetic, *Lucene in action: covers Apache Lucene 3.0*. Manning Publications Co., 2010.
- [84] O. Medelyan, “Human-competitive automatic topic indexing,” Ph.D. dissertation, The University of Waikato, 2009.
- [85] R. Mihalcea and P. Tarau, “Textrank: Bringing order into text,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 404–411.



- [86] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [87] J. Mun, M. Cho, and B. Han, “Text-guided attention model for image captioning.” in *AAAI*, 2017, pp. 4233–4239.
- [88] P. Nakov, S. Rosenthal, S. Kiritchenko, S. M. Mohammad, Z. Kozareva, A. Ritter, V. Stoyanov, and X. Zhu, “Developing a successful semeval task in sentiment analysis of twitter and other social media texts,” *Language Resources and Evaluation*, vol. 50, no. 1, pp. 35–65, 2016.
- [89] M. Naughton, N. Kushmerick, and J. Carthy, “Event extraction from heterogeneous news sources,” in *Proceedings of the AAAI Workshop Event Extraction and Synthesis*, 2006, pp. 1–6.
- [90] R. Navigli and S. P. Ponzetto, “Babelnet: Building a very large multilingual semantic network,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 216–225.
- [91] T. H. Nguyen, K. Cho, and R. Grishman, “Joint event extraction via recurrent neural networks,” in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016, pp. 300–309.
- [92] Y. Nishihara, K. Sato, and W. Sunayama, “Event extraction and visualization for obtaining personal experiences from blogs,” in *Symposium on Human Interface*. Springer, 2009, pp. 315–324.
- [93] A. Nurhudatiana and A. W.-K. Kong, “On criminal identification in color skin images using skin marks (rppvsm) and fusion with inferred vein patterns,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 916–931, 2015.
- [94] M. Okamoto and M. Kikuchi, “Discovering volatile events in your neighborhood: Local-area topic extraction from blog entries,” in *Asia Information Retrieval Symposium*. Springer, 2009, pp. 181–192.

## BIBLIOGRAPHY

---

- [95] L. Page, S. Brin, R. Motwani, T. Winograd *et al.*, “The pagerank citation ranking: Bringing order to the web,” in *Proceedings of the 7th International World Wide Web Conference*. Technical report, Stanford digital library technologies project, 1998, pp. 161–172.
- [96] A. Pérez, A. Casillas, and K. Gojenola, “Fully unsupervised low-dimensional representation of adverse drug reaction events through distributional semantics,” in *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM)*, 2016, pp. 50–59.
- [97] K. Radinsky, S. Davidovich, and S. Markovitch, “Learning causality for news events prediction,” in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 909–918.
- [98] P. V. Rahul, S. K. Sahu, and A. Anand, “Biomedical event trigger identification using bidirectional recurrent neural network based models,” *arXiv preprint arXiv:1705.09516*, 2017.
- [99] P. Rani, V. Raychoudhury, S. S. Sandha, and D. Patel, “Mobile health application for early disease outbreak-period detection,” in *IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE, 2014, pp. 483–488.
- [100] S. Rao, D. Marcu, K. Knight, and H. Daumé III, “Biomedical event extraction using abstract meaning representation,” in *BioNLP 2017*. Association for Computational Linguistics, 2017, pp. 126–135.
- [101] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
- [102] E. S. Ristad and P. N. Yianilos, “Learning string-edit distance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, 1998.

- [103] A. Ritter, O. Etzioni, S. Clark *et al.*, “Open domain event extraction from twitter,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 1104–1112.
- [104] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
- [105] D. Rusu, J. Hodson, and A. Kimball, “Unsupervised techniques for extracting and clustering complex events in news,” in *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, 2014, pp. 26–34.
- [106] S. K. Sahu, A. Anand, K. Oruganty, and M. Gattu, “Relation extraction from clinical texts using domain invariant convolutional neural network,” *arXiv preprint arXiv:1606.09370*, 2016.
- [107] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake shakes twitter users: real-time event detection by social sensors,” in *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010, pp. 851–860.
- [108] S. Santiso, A. Casillas, A. Pérez, M. Oronoz, and K. Gojenola, “Document-level adverse drug reaction event extraction on electronic health records in spanish,” *Procesamiento del Lenguaje Natural*, no. 56, pp. 49–56, 2016.
- [109] D. Sculley, “Web-scale k-means clustering,” in *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010, pp. 1177–1178.
- [110] T. J. Siddiqui and U. S. Tiwary, “Integrating relation and keyword matching in information retrieval,” in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2005, pp. 64–73.
- [111] T. J. Siddiqui and U. S. Tiwary., “Query based summary for assessing document relevance,” in *1st International Conference on Digital Information Management*. IEEE, 2006, pp. 314–319.

## BIBLIOGRAPHY

---

- [112] K. Sim, C. Phua, G.-E. Yap, J. Biswas, and M. Mokhtari, "Activity recognition using correlated pattern mining for people with dementia," in *Annual International Conference of the Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 7593–7597.
- [113] V. K. Singh, S. Verma, and M. Kumar, "Dbcs: A decomposition based compressive sensing for event oriented wireless sensor networks," *Wireless Personal Communications*, vol. 99, no. 1, pp. 351–369, 2018.
- [114] V. Solovyev and V. Ivanov, "Knowledge-driven event extraction in russian: corpus-based linguistic resources," *Computational Intelligence and Neuroscience*, vol. 2016, p. 16, 2016.
- [115] S. Stamou, V. Krikos, P. Kokosis, A. Ntoulas, and D. Christodoulakis, "Web directory construction using lexical chains," in *International Conference on Application of Natural Language to Information Systems*, 2005, pp. 138–149.
- [116] R. Steinberger, B. Pouliquen, and E. Van der Goot, "An introduction to the europe media monitor family of applications," *arXiv preprint arXiv:1309.5290*, 2013.
- [117] L. Sterckx, T. Demeester, J. Deleu, and C. Develder, "When topic models disagree: Keyphrase extraction with multiple topic models," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 123–124.
- [118] R. Stern, B. Sagot, and F. Béchet, "A joint named entity recognition and entity linking system," in *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, ser. HYBRID '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 52–60. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2388632.2388640>
- [119] J. Strötgen and M. Gertz, "Heideltime: High quality rule-based extraction and normalization of temporal expressions," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2010, pp. 321–324.

- [120] J. Strötgen and M. Gertz., “Event-centric search and exploration in document collections,” in *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, 2012, pp. 223–232.
- [121] M. Subrahmanyam, R. Maheshwari, and R. Balasubramanian, “Local maximum edge binary patterns: a new descriptor for image retrieval and object tracking,” *Signal Processing*, vol. 92, no. 6, pp. 1467–1479, 2012.
- [122] A. Sun, M. Lachanski, and F. J. Fabozzi, “Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction,” *International Review of Financial Analysis*, vol. 48, pp. 272–281, 2016.
- [123] R. Sun, Y. Zhang, M. Zhang, and D. Ji, “Event-driven headline generation,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, 2015, pp. 462–472.
- [124] H. Tanev, J. Piskorski, and M. Atkinson, “Real-time news event extraction for global crisis monitoring,” in *International Conference on Application of Natural Language to Information Systems*. Springer, 2008, pp. 207–218.
- [125] C. Tang, A. W.-K. Kong, and N. Craft, “Using a knowledge-based approach to remove blocking artifacts in skin images for forensic analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1038–1049, 2011.
- [126] A. Tolstikov, C. Phua, J. Biswas, and W. Huang, “Multiple people activity recognition using mht over dbn,” in *International Conference on Smart Homes and Health Telematics*. Springer, 2011, pp. 313–318.
- [127] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *Journal of Machine Learning Research*, vol. 2, no. Nov, pp. 45–66, 2001.

## BIBLIOGRAPHY

---

- [128] M.-V. Tran, M.-H. Nguyen, S.-Q. Nguyen, M.-T. Nguyen, and X.-H. Phan, “Vnloc: A real-time news event extraction framework for vietnamese,” in *Fourth International Conference on Knowledge and Systems Engineering*. IEEE, 2012, pp. 161–166.
- [129] P. D. Turney, “Coherent keyphrase extraction via web mining,” in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2003, pp. 434–439.
- [130] S. K. Vipparthi, S. Murala, A. B. Gonde, and Q. J. Wu, “Local directional mask maximum edge patterns for image retrieval and face recognition,” *IET Computer Vision*, vol. 10, no. 3, pp. 182–192, 2016.
- [131] M. Walther and M. Kaisser, “Geo-spatial event detection in the twitter stream,” in *European Conference on Information Retrieval*. Springer, 2013, pp. 356–367.
- [132] K. Wang, C. Thrasher, E. Viegas, X. Li, and B.-j. P. Hsu, “An overview of microsoft web n-gram corpus and applications,” in *Proceedings of the NAACL HLT 2010 Demonstration Session*, ser. HLT-DEMO '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 45–48. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1855450.1855462>
- [133] X. Wang, F. Zhu, J. Jiang, and S. Li, “Real time event detection in twitter,” in *International Conference on Web-Age Information Management*. Springer, 2013, pp. 502–513.
- [134] F. Wu and D. S. Weld, “Open information extraction using wikipedia,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 118–127.
- [135] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International Conference on Machine Learning*, 2015, pp. 2048–2057.
- [136] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, “Textrunner: open information extraction on the web,” in *Proceedings of Human Language*



- Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations.* Association for Computational Linguistics, 2007, pp. 25–26.
- [137] H. Yu, L. Gao, W. Liao, B. Zhang, A. Pizurica, and W. Philips, “Multiscale superpixel-level subspace-based support vector machines for hyperspectral image classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 11, pp. 2142–2146, 2017.
- [138] D. Zhou, L. Chen, and Y. He, “An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization.” in *AAAI*, 2015, pp. 2468–2475.
- [139] D. Zhou, T. Gao, and Y. He, “Jointly event extraction and visualization on twitter via probabilistic modelling,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2016, pp. 269–278.
- [140] D. Zhou, X. Zhang, and Y. He, “Event extraction from twitter using non-parametric bayesian mixture model with word embeddings,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 1, 2017, pp. 808–817.
- [141] D. Zimbra, M. Ghiassi, and S. Lee, “Brand-related twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks,” in *Proceedings of the 2016 49th Hawaii International Conference on System Sciences.* IEEE, 2016, pp. 1930–1938.