International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-5, June 2020

Event Detection using Deep Learning

Ariveni Triveni, Guntur Kesava Sai Adithya, Sanneboina Karthik, Deepak Kumar Sahoo

Abstract: *Now a days, Twitter posts more than 400 million tweets every day can disclose real-world details as events grow. Event detection is a method to find real events which occur over time and space. Recent social media networks, such as Face Book, Instagram, Whatsapp and Twitter have been widely documented in real time. In the case of an earthquake, for example, people report earthquake-related information instantly, which allows the earthquake to be quickly detected. In this paper, we have developed a data filter based on functions like keywords, numbers and context. Every user feed is viewed as a sensor and such sensor selection provides a device capable of alerting registered users immediately. Using word embedding models, tweets are converted into numerical vectors. Tweets are classified into political, criminal, social, medical, disaster and miscellaneous predefined classes. Classification task is done by using long short-term memory networks (LSTM). A large number of tweets for the creation and testing of our proposed model are obtained via the Twitter API endpoint, which is marked as an effective technique.*

Keywords: ***Event Detection, Long-short term memory (LSTM),Training data Creation, Word Embedding.***

I. INTRODUCTION

Twitter is a social media site for millions of people to post life notifications. Also, these tweets talk about local app activities. Though local reports are made, it takes an organization a considerable time to learn about the incident, investigate it, and report on it, particularly in comparison with the duration of the event.Users track the incidents in real time, and before a news outlets post this data, they are able to listen to an incident on Twitter. In this project we analyze tweets from a certain geographic region to see if there has been an accident. Then we publish the shortest tweet of the incident. The solution of this problem would be a simple way to alert a consumer to a local incident. Our method divides the information into position buckets, calculates tweet spikes, and then categorizes tweets on a similar basis. Users can be informed of local incidents in real-time before news outlets can announce them. The exact specification varies according to the type of case. For example, a user can be able to escape a delay by following an alternative path, if a traffic delay is identified. Our program will also provide news outlets with information for further investigation. We needed geo-tagged tweets for our project over a short time. Because of measurement limitations, we decided to examine tweets in a geographical area. We began by using the Twitter Search API to gather data from the Metropolitan Area of NYC. Nonetheless, it was not possible to obtain the appropriate amount of information due to our time limit and the rates cap for the Twitter API.

Twitter is one of the world's largest micro blogging platforms, with over 330 million active users. Twitter allows users to post short tweets, called 'what's going on.' They can track real-life activities over time and space via the twitter website. In the past decade, Twitter event identification has grown. The observation and analysis of tweet text flow help to distinguish real activities at a given time and place.[7] Mechanical techniques to automatically recover context from unstructured tweets. Until implementing machine learning techniques in variable named messages, tweets should be converted into digital function vectors in the fixed-size format. This procedure divides tweets into tokens (example:words) and assigns a single integer to each possible tokens, and then defines tweets using a dimensionally structured matrix where different tokens are accessible[8]. The standard term TF-IDF frequency-inverted paper. A matrix with 1 row per tweet and 1 column per token per tweet is required for the concatenation of all Tweet Vectors. This matrix is applied to an algorithm for supervised classification functions. Firstly, the temporal order of words is ignored, and the tweet relationship between words is semantic and syntactic. Secondly, because any twitch uses a small subset of words, the numbers of terms possible are quite large, it is a matrix of barely defined characteristics, such as zero and dimension curse. We use supervised deep learning algorithms to model tweets and supervised classification research architectures. In addition to representing tweet text in low-dimensional regions, Word embedding models are employed for consideration of the semanticization between tweets.[9][10]

II. RELATED WORK

G.Kumaran [1] has improved New Event Detection by using text classification techniques and named entities in a new way. He has explored improvements in document presentation using a vector-based space-based NED framework. It has shown that it is only better useful to discuss named entities in certain instances.

Revised Manuscript Received on May 15, 2020.

* Correspondence Author

A.Triveni*, Department of Computer Science and Engineering , Koneru Lakshmaiah Educational Foundation,Vaddeswaram,Guntur district ,Andhra Pradesh ,India.

G.Kesavsai, Department of Computer Science and Engineering , Koneru Lakshmaiah Educational Foundation,Vaddeswaram,Guntur district ,Andhra Pradesh ,India.

S.Karthik, Department of Computer Science and Engineering , Koneru Lakshmaiah Educational Foundation,Vaddeswaram,Guntur district ,Andhra Pradesh ,India.

Deepak Kumar Sahoo, Assistant Professor, Department of Computer Science and Engineering , Koneru Lakshmaiah Educational Foundation,Vaddeswaram,Guntur district ,Andhra Pradesh ,India.

In particular, he has developed a multi-stage NED system that is much better than traditional NED systems. He has tokened, removed stop words and stuck and generated data document vectors using the Lemur method.. The 418 stop words contained in the In query stop list have been used as a part of the Lemur k-stemming algorithm implementation. Incremental TF-IDF weighting and the standardization of document parity was performed before the end of a plot. They recognized that the main cosine-like parallel metric is the error and decided to improve our dependence on the score by analyzing other criteria, such as the category (financials, injuries, etc.).And they established basic rules which would represent the human being's questions before determining if a story was old or new. Firstly, for each paper we construct three vector depictions α, β and γ. That's the way we work. The first α represents the terms in the text; the second β is just a specified individual, and the third γ represents all terms in the text. BBN Identifier has been used to classify designated entities. Only ß (and thus γ) as applicable, GPE, language, place, nationality, organization, human, Cardinal, Ordinary and date, and time have we considered the named entities. Only when two stories were compared for each document was the corresponding story for the other document comparable. The correlation with β and γ was also calculated in the most comparable tale to the stories with an γ correlation. Chao Zhang [2] suggested that GEOBURST, an efficient local event detector in real-time, recognize local event discovery in real time. The main goal is to locate a particular occurrence that typically contributes to a large amount of tweets on the spot. In the first step of GEOBURST the geographic and semantical contribution of the same geographic tweets can be found in all geo-topic clusters in the question window. The kernel function is used for the geo-topic measurement and the semantic portion will be collected by a random colloquial graph. Phase one of second classification of the GEOBURST is based on time and space explosion. Both decisions can be made. We continuously summarize the flow and store the effects of the operating time in an open space structure. A new method for the quest of applicants and the GEOBURST rating module helps to check efficiency by using streamly forwarded data. spatiotemporal bursting. Pivots serve for future local events as symbolic tweets. Naturally, for events of candidates, they draw identical tweets. GEOBURST summarizes current tweets and contrasts past spatial engineering operations for the collection of specific local events from the list of candidates in the real-time context. Finally GEOBURST now has an upgrade module, which identifies new pivots that do not take much time to adjust the question window. GEOBURST is therefore able to keep track of continuous streams in real time. David Chen, Ashwin Gupta, ShruthiKrish, Raghav Prakash, Wei Wang[3].They used geo-tagged tweets for a short period. Firstly, they filter data on latitude and longitude, concentrate on tweets, where all non-English tweets are posted and deleted. Various models have been attempted, such as k-means, hierarchical clustering, DBSCAN or HDBSCAN. They have also used HDBSCAN as noisy clusters are robust and variable. Groups of tweets that most likely represent a case were found for each site bin. It then sets standard deviation thresholds to mark hours per tweet for potential events. They said that more work needs to be done to reduce noise and probably find more cases, reducing the misclassification rate. You divide tweets into local buckets using an established date set, use the DBSCAN to identify major events and pick a tweet headline to reflect the best of any occurrence. The article proposed a scalable event detection system called Twitter News, designed for the identification and tracking of newsworthy events from Twitter in real time, from Mahmud Hassan, Mehmet Orgun and Rolf Schwitter[4].Twitter News is a new approach to helping to slowly compile tweets associated with specific events within a given time frame using a random indexation-based vector model using responsive localization..The Twitter data stream event detection problem is divided into two main phases in an incremental clustering sense. In the first step the number of tweets that debate a subject / event is bursting, and in the second process tweets that address the same event are grouped or grouped. After a tweet is prepared, it is up to the first stage to decide whether a previous topic is addressed in the actual input tweet for the framework. When the input tweet addresses a subject already used, a soft tweet burst linked to a theme or occurrence has taken place, and the output of the first step states that the input tweet is not special. The first stage operation is implemented in accordance with the Locality Sensitive Hashing vector model based on Random indexing (RI). The second phase implements a defragmentation technique for dealing with the fragments generated when a single event is observed as a new event on multiple occasions. The second phase is applied using a innovative approach with the implementation of the traditional incremental clustering algorithm.[5][6] The newsworthy events of the candidate event clusters will be replicated after the second cycle. The use of a Word-level, publicly available scheme and a representative tweet to assess the effectiveness of Twitter News for each event cluster is documented with respect to the use of a Longest Commons Subsequence (LCS)-based scheme.

III. PROPOSED METHODOLOGY

The Following methods can be used to extract the features from raw tweets.

1. Preprocessing raw tweets
2. Classification of tweets
3. Ranking of Tweets in each class

3.1 Preprocessing Raw Tweet :

For machine learning the pre-processing of data is an important step, since data quality and useful knowledge affect the ability of our system to learn. We should note also that the quality of training data dictates the output of a model while a machine learning algorithm is training. In reality, the data you obtain will in most cases not be clean. This means that the data is not consistent with data formats, lacking meanings, descriptions and features with very different ranges. It is therefore important for us to preprocess our data prior to integrating it into our model. We took the following preprocessing steps:

a. Removing symbols, tags
b. Removing Urls
c. Tokenization
d. Removal of Stop Words
e. Removal of Repeating words
f. Stemming
g. POS Tagging

a. Removing Symbols, tags: If they are not applicable to our research, eliminates symbols and tags. This collection of symbols is dropped {:;!@#$%^&*(_)>/?.<,"|'\}[+=`~]

*b. Removal of url:*Tweets may consist of urls. In Order to process the data for classification we need to remove unnecessary links. It shows links to other Web pages and Websites.

c. Tokenization: It is the process of dividing the sentence into words.

d. Removal of Stop Words: This can be used to remove stop words i.e., Prepositions and Conjunctions like is, the, with, to, and etc..,

e. Removal of repeating Words: Remove the repeating letters in the sentence if there is a repetition of a word.

f. Stemming: Stemming is a mechanism whereby words are reduced to their word stem, Root form or base shape.

g. POS Tagging: It is the process of making each word a tuple consisting of word and its part of speech.

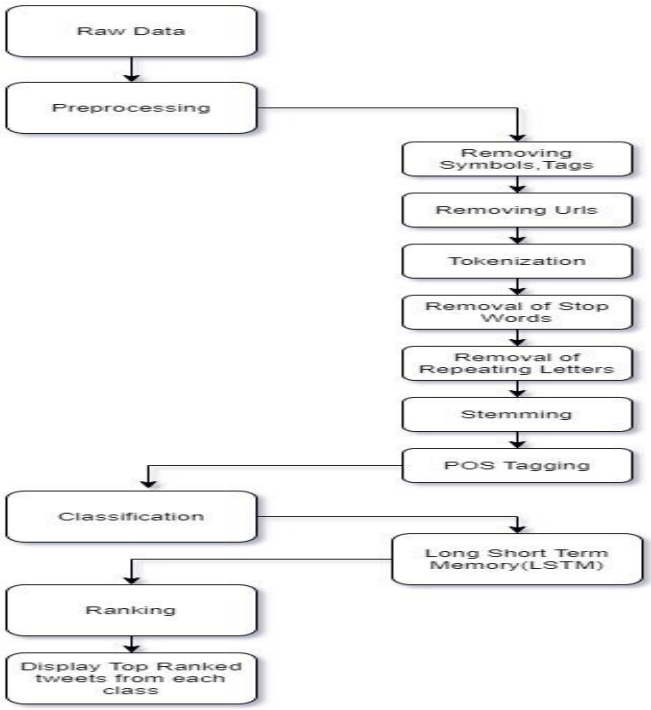

Fig. 1.*Block Diagram*

Example:

Input: Our new integration with @Zendex is improving #customerexperience by making support even easier. https://t.co/z8YXpkPw3I @Forbes is https://t.co/NMLJz4eQr

Output:

Step-a: Our new integration with Zendex is improving customerexperience by making support even easier httpstcoz8YXpkPw3I Forbes is httpstcoNMLJz4eQr

Step-b: Our new integration with Zendex is improving customerexperience by making support even easier Forbes is

Step-c: Our, new, integration, with, Zendex, is, improving, customerexperience, by, making, support, even, easier, Forbes, is

Step-d: Our, new, integration, Zendex, improving, customerexperience, making, support, even, easier, Forbes

Step-e: Our, new, integration, Zendex, improving, customerexperience, making, support, even, easier, Forbes

Step-f: Our, new, integr, Zendex, improv, customerexperi, make, support, even, easier, Forb

step-g: [('integr', 'NN'), ('our', 'PRP$'), ('new', 'JJ'), ('even', 'RB'), ('forb', 'VBP'), ('zendex', 'NN'), ('make', 'VBP'), ('easier', 'JJR'), ('customerexperi', 'NN'), ('support', 'NN'), ('improv', 'NN')]

After Completing preprocessing we will get POS Tagged Words, In that we need to consider only the Proper Nouns.

3.1.1Word Embeddings:

The embedding of a word is an acquired text representation where the words have a common meaning. Embedding was usually implemented using a One-Hot Encoding Method, using an array with a length of equal to the number of different words of the vocabulary in each word of the tweet. After getting Some Distinct Words from preprocessing we need to create a training data Vector. For example, Limit the data collection to the top three thousand words. Set the vector size in each tweet at 250.

3.1.2 Training Data Creation :

Tweet-1: Another day, another lie, another crime Anyone surprised?

Tweet-2: We find Ourselves at a moment of National Emergency.

Tweet-3: FSU social aid it again

Distinct words:

Surpris,day,another,lie,crime,anyone,we,find,outside,moment ,nation,emerg,social,FSU,it,again -16

f1f2f3f4f5f6f7f8f9f10f11f12f13f14f15f16	Target value
T1:- 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0	2
T2:- 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0	1
T3:- 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	5

3.2 Classification Of Tweets :

It is a challenge for data analytics, i.e. to find a model defining and separating data classes and concepts. It poses the question of deciding, by virtue of the training set of data containing observations which of a number of categories, new observations belong to.

The following predefined classes can be used:

1. Political
2. Criminal
3. Social
4. Medical

5. Disaster
6. Miscellaneous

We are using Long Short Term Memory(LSTM) Algorithm to classify the data. A cell, an input gate, an output gate and a forgotten gate form the LSTM basic unit. The cell recalls values at random moments, and the three gates manage the information flows in and out of the cell. Not only single data items, but also entire data sequences can be analyzed.

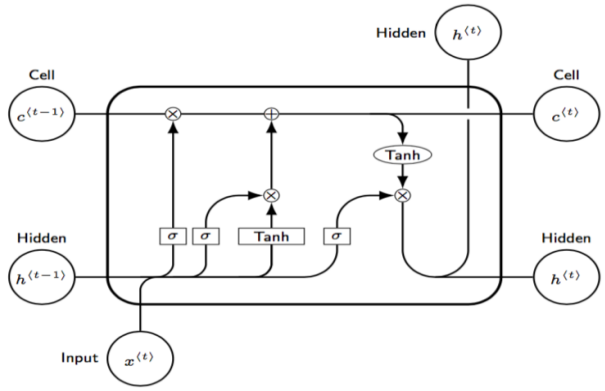

Fig. 2. Structure of LSTM(Long -Short Term Memory)

3.3 Ranking Of Tweets :

1. Based on the author
2. Based on tweets
3. On the basis of content
4. User-based material

We are working to achieve the weight of a tweet by using a content-based approach:

Weight of the total value(Wn) =Statistical weight + hash tag+proper nouns.

3.3.1 Author based:

The user's profile can be inferred.
She has a lot of supporters (10 K – 50 K)

3.3.2Tweet-based:

This function includes hashtags, urls, http:/xyz, connection (@user), marks of question? The first person pronoun (I), (exclamation mark!) (quotation mark (''), three fold (e.g. coool).)

3.3.3 Content-based:

The content-based functionality relates to the tweet details. It is the uniqueness of the tweet (i.e. the total distance from the other tweets in the history of the user in the previous week).

Ex:

a. D1: He is a lazy boy. She is also lazy.
b. D2: Smith is a lazy person.
c. The list created would consist of all the unique tokens in the corpus C.
d. = [‘He’, ’She’,’ lazy’, ’boy’,’ Smith’,’ person’]
e. We have, W=2, T=6

	He	She	lazy	boy	Smith	person
D1	1	1	2	1	0	0
D2	0	0	1	0	1	1

TABLE-1: IDF VALUES

From the above table, To make TF-IDF from scratch in python, we need two separate steps. First we have to create the TF function to calculate total word frequency for all documents. If you match up the above numbers with their respective location, you will see they match up with the build from scratch method. In the first column, 0 is the first sentence and 1 is the second sentence.

3.3.4 User-based:

These characteristics are linked to the recipient of the Twitter feed. From the users' point of view, the features of the site are as follows: is the author my friend, is the author my friend in conversation, did I mention her, used hash tag in the tweet (wherever), used the domain url in tweet (if there is one)? I have tweeting to the consumer.

i. Features of IDF:

This is a particular approach based on the frequency system. A single word (or tweet) in the whole body except in one text.

ii. Importance in tf- idf:

TF = (Number of times t in a document)/(Number of word s in a document) .
IDF = log(N / n), where N is the number of documents and n is the number of documents in which the word t occurs.
TF-IDF compared to TF*IDF.
TFIDF score for term I in document j= TF(i,j) * IDF(i)

Where

IDF=Inverse Document Frequency

TF=Term Frequency

$$TF(i,j)=\frac{\text{term i frequency in document j}}{\text{Total words in document j}}$$

$$IDF(i)=\left(\frac{\text{document with term i}}{\text{document with term i}}\right)$$

And

T= Term

J = Document

$$tf_{i,j}=\frac{n_{i,j}}{\sum_k n_{i,j}}$$

$$w_{i,j}=tf_{i,j}\times\log\left(\frac{N}{df_i}\right)$$

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

TABLE-2: TF-IDF

From the above table, we can see that TF-IDF of common words was zero, which shows they are not significant. On the other hand, the TF-IDF of "car", "truck", "road", and "highway" are non-zero. These words have more significance.

We need two separate steps to make TF-IDF in python from scratch. To measure the maximum word frequency for all documents, we first have to construct the TF function.

IV. RESULT AND DISCUSSION

Determining how to calculate precision has been challenging because the dataset has not been specified and events are described subjectively. To see how well our project has been performed, we have checked the data and identified all events in our data set. That was then compared to the events contained in our project. In fact, for validation purposes, we marked our dataset.

In our project, the events typically occur in a certain geographic region. We used the extreme learning model, and the neural network accuracy score is 83 percent and we will create a long-term storage network to increase the accuracy score. We have learned a lot during this process. Most importantly, we needed more detail, we tried to change the cluster size. This could explain occurrences in various geographical areas**.**

V. CONCLUSION AND FUTURE WORK

Our goal for this project was to classify local events with geo tagged tweets. We split tweets into location buckets using an existing dataset, used LSTM to identify significant events, and picked a headline tweet that best represents each situation. Besides, we hypothesized that tweets will be used more frequently than not to comment on some sort of case. But this was not so, because more users use the website to update them easily and personally. We found it very difficult to assess performance in unmonitored issues from the perspective of machine learning, since sometimes there are no objective metrics to verify. We provided an event detection model for the identification of the most important geographical area incidents. Our dataset contains 3000 tweets. We considered 80 percent data for training, and 20 percent remaining for testing. First, we have preprocessed the data by deleting icons, marks, URLs, words to stop, words to repeat. For the classification of the data, we used long-term memory networks. We eventually used the ranking algorithm to learn that the most interesting thing that happened on a specific day.

We 'd start gathering additional data in the future. We would tend to collect data for denser areas and locations. There are also ways that we can develop our anomaly detection process. We may analyze the tweets' time dependent intensity to detect events of various sizes and allow us to theoretically differentiate between events of an hour and a day. We may also try to adjust the size of the location. This could illustrate incidents in a specific geographical region.

REFERENCES

1. Kumaran, G., and Allan, J. 2004. Text classification and named entities for new event detection. In SIGIR '04: Proceedings of the 27th annual international ACM SIGIR,CRDIR 297–304. New York, NY, USA: ACM
2. Chao Zhang, Guangyu Zhou, Quan Yuan, Yu Zheng, Lance M. Kaplan, Shaowen, and Jiawei Han. 2016. GeoBurst: Real-Time Local Event Detection in Geo-Tagged Tweet Streams. In SIGIR. 513–522
3. David Chen, Ashwin Gupta, ShruthiKrish, Raghav Prakash, Wei Wang: finding local events using twitter data.
4. Hasan, Mahmud ;Orgun, Mehmet A. ; Schwitter, Rolf. / A survey on real-time event detection from the Twitter DataStream. In: Journal of IS. 2018 ; Vol. 44, No. 4. pp. 443-463.
5. ALLAN, J., R. PARKA, and LAVRENKO V. 1998. On-line new event detection and tracking. In Proceedings of the 21st Annual International ACM SIGIR CRDIR , SIGIR '98, ACM, New York, NY, pp. 37–45.
6. ATEFEH, F. & KHREICH, W. 2015. A survey of techniques for event detection in twitter.CI , 31, 132-164.
7. C. C. Aggarwal and K. Subbian, "Event detection in social streams." in Proceedings of the SIAM Int. Conf on Data Mining, SDM, vol. 12. SIAM, 2012, pp. 624–635.
8. J. McMinn and J. M. Jose, "Real-time entity-based event detection for Twitter," in Proceedings of the Experimental IR Meets Multilinguality, Multimodality, and Interaction: 6th Int. Conf of the CLEF Association, ser. CLEF '15. Springer, 2015, pp. 65–77.
9. W. Dou, K. Wang, W. Ribarsky, and M. Zhou, "Event detection in social media data," in Proceedings of the IEEE VisWeek Workshop on IVTA - Task Driven Analytics of Social Media Content, 2012, pp. 971–980.
10. DABIRI, S. & HEASLIP, K. 2019. Developing a Twitter-based traffic event detection model using deep learning architectures. Expert Systems with Applications, 118, 425-439.
11. Lam, W.; Meng, H. M. L.; Wong, K. L.; and Yen, J. C. H. 2001. Using contextual analysis for news event detection. Int. J. Intell. Syst. 16(4):525–546.
12. R. Li, K. H. Lei, R. Khadiwala, K. C.-C. Chang, "TEDAS: A Twitter-based event detection and analysis system", Proc. 28th IEEE ICDE, pp. 1273-1276, 2012.

WEB REFERENCES

Wikipedia, From Wikipedia, the free encyclopedia, "Long Short -Term Memory "[online], *https://en.wikipedia.org/wiki/Long_short-term_memory* [Accessed-on-21-March]

AUTHORS PROFILE

A.Triveni , is a B.Tech Student of the Computer Science and Engineering Department at the Koneru Lakshmaiah Educational Foundation at Vaddeswaram, Guntur District ,Andhra Pradesh, India . Currently, pursuing her final year Under graduation, Recently placed in a reputed IT Company. She is well specialized in Computational Intelligence.

G.Kesava sai Adithya , is a B.Tech Student of the Computer Science and Engineering Department at the Koneru Lakshmaiah Educational Foundation at Vaddeswaram, Guntur District ,Andhra Pradesh, India . Currently, pursuing his final year Under graduation, Recently placed in a Top IT Company. His Research interests are Machine Learning and Data Mining.

S. Karthik, is a Student of the Computer Science and Engineering Department at the Koneru Lakshmaiah Educational Foundation at Vaddeswaram, Guntur District, Andhra Pradesh, India. Currently, pursuing his final year Under graduation, Looking for pursuing Higher Education in Abroad . His Research interest is Data Mining.

Deepak Kumar Sahoo earned his Master of Technology (M.Tech) from IIITBH in the year 2009. Worked as faculty in Dept. Of computer Science and Engineering at KL University. Current Research interest Natural Language processing and Information Retrieval.