**International Journal of Innovative Research in Computer Science & Technology (IJIRCST)**
**ISSN: 2347-5552, Volume-6, Issue-4, July-2018**
**DOI: 10.21276/ijircst.2018.6.4.2**

# Using Big Data Technique for Building Edit Alert System for Wikipedia Infoboxes Based on Map-Reduce Method

**Khushboo Bhatia, Arnab Halder, Yashi Yadav, Ankush Sarsewar, Priyanka Singh, Khushboo Khurana**

***Abstract*****-**Wikipedia is an online encyclopedia and has become a vital information resource for users as well as for many knowledge bases derived from it. This information requires manual editing for update. Wikipedia provides an infobox on the right hand side of many articles. An infobox of a Wikipedia article generally contains key facts in thearticle and is organized as attribute-value pairs. All the Wikipedia's content is manually updated or maintained by contributors. This leads to the fact that its information is not updated regularly and completely. In this paper, we present a novel system that focuses onprediction of data items that are most likely to be updated, based on the category of page, record key, last time updated, etc. for alerting Wikipedia editors, about the data items that might need update soon, using Time series modeling. Concept of Bipartite graph is used to perform user based collaborative filtering to find similar editors who might be interested in editing the infobox. The update alert is sent to editors found using Bipartite graph along with the past editors of a particular infobox. The technique to deal with vandalic and erroneous edits is also discussed and its analysis is given. We have also presented various tasks that can be carried out on infoboxes.

***Keywords:*** Big Data, Map-Reduce, Wikipedia Infobox, update alert, bipartite graph

**Manuscript received June 18, 2018.**

**Khushboo Bhatia**, Student, Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur,India, (email: khushb99@gmail.com)

**Arnab Halder**, Student, Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur,India

**Yashi Yadav,** Student, Computer Science and Engineering,Shri Ramdeobaba College of Engineering and Management ,Nagpur, India,

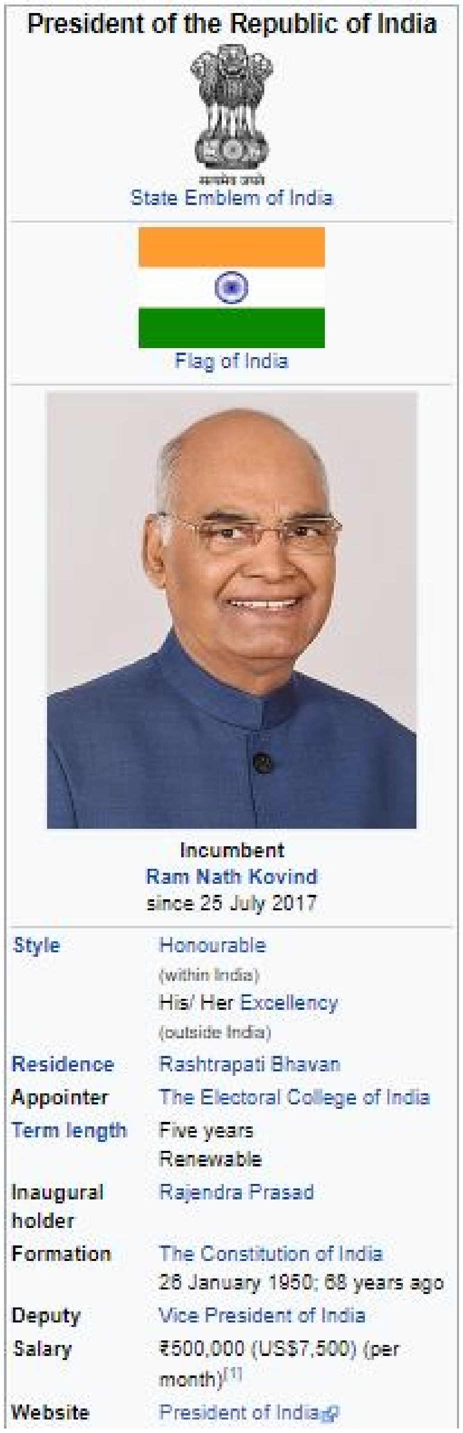
**Ankush Sarsewar**, Student, Computer science and engineering (CSE), Shri Ramdeobaba College of Engineering and Management, Nagpur, India,

**Priyanka Singh**, Student,Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur,India,

**Khushboo Khurana,** Assistant Professor**,** Computer Science and Engineering,Shri Ramdeobaba College of Engineering and Management,Nagpur, India,

## I. INTRODUCTION

There is a lot of information available over the web. There are many websites that provide information in structured or unstructured format. One such website is Wikipedia.
It is a free central hub of knowledge which provides information about almost every domain. It provides information in the form of text, along with images. Many pages also consist of table indicating key attribute values and a summarization of the text. This table is referred as infobox. That is, an infobox is a small tabulated box present on the right top corner that contains information about the described subject on a Wikipedia page in a summarized format highlighting important facts and figures. An example of "President of the Republic of India" infobox is as shown in figure 1, which is the infobox corresponding to the Wikipedia article "President of India". It enlists attributes like style, residence, appointer, term length, etc. Infoboxes not only allow readersto rapidly gather the most important information about some aspects of thearticles in which they appear, but also provide a source for many knowledge bases derived from Wikipedia. The articles and infoboxes are manually updated by editors and contributors.There are instances wherein the information often becomesobsolete and require updationsregularly and completely. Some articles and infoboxes may require updation frequently, some require updation periodically, and others may not require updation at all. Some instance values may change completely and some have minor changes. Taking into consideration all the observations, it is essential to have automated system that can ensure the availability of latest and correct information. In this paper we propose a novel method for prediction of attributes that are most likely to be updated. This is used to build a automated update alert system for alerting Wikipedia editors that some attributes that might need update soon.



President of the Republic of India

State Emblem of India

Flag of India

Incumbent
Ram Nath Kovind
since 25 July 2017

| | |
|---|---|
| Style | Honourable (within India) His/ Her Excellency (outside India) |
| Residence | Rashtrapati Bhavan |
| Appointer | The Electoral College of India |
| Term length | Five years Renewable |
| Inaugural holder | Rajendra Prasad |
| Formation | The Constitution of India 26 January 1950; 68 years ago |
| Deputy | Vice President of India |
| Salary | ₹500,000 (US$7,500) (per month)[1] |
| Website | President of India |

Fig. 1: Sample Infobox

## II. EXPERIMENTAL SECTION

### A. *Wikipedia Infobox edit history dataset*

For our research we have used Wikipedia edit history dataset (Can be accessed from: http://www.inf.ed.ac.uk/teaching/course/irds/ miniproject-dataset.html ). It contains Wikipedia's full edit history updated to March 23,2012.

The dataset contains a total of 510,102,778 individual infobox attributes updates (IAU), that correspond to 2,040,181 articles. A sample record from the dataset is as shown in Figure.2. Each record contains the title of infobox, timestamp, contributor_ip, key, old value, new value (if there is an update), infobox name, id and comment. Timestampis a sequence of characters or encoded information identifying when a certain edit or update occurred. The contributor_ip is a unique entity for identifying the contributor or editor of that article. Key denotes the attribute name. For instance, the above screenshot shows article titled "India", with its timestamp "1332431387". The unique contributor_ip for the editor of this infobox is "Dwaipayanc". The key for this article is 'time_zone_dst'.The infobox name is "Infobox country". The id for this infobox is 483380381.

```
{"title":"India",
"timestamp":1332431387,
"contributor_ip":"Dwaipayanc",
"key":"time_zone_dst",
"oldvalue":"not observed",
"infobox_name":"Infobox country",
"id":483380381,
"comment":"Revert to revision 483379135 dated 2012-03-22 15:40:36 by Dwaipayanc using [[:en:Wikipedia:Tools/Navigation_popups|popups]]"}
```

Fig.2: Sample Record in Wikipedia edit history dataset

### B. *Proposed Methodology*

In this paper we have proposed a novel system that can perform following tasks on the infobox:

- Finding the most edited infobox
- Update alert to Wikipedia editors based on time series modeling
- Finding vandalism from the edit history dataset.

Each of these tasks are discussed next.

#### a) Finding most-edited infobox

The edit history of infobox contains edits made to an attribute in the form tuple <title,key,oldvalue,newvalue>. Most edited infobox is the one whose attributes have been changed most number of times. Our implementation iterates through every infobox and counts the edits made to it. It associates with every infobox name a count that denotes the number of edits made. Article title can also be used instead of infobox name. We have designed Map-Reduce process to find the most edited infobox. Our proposed system uses a cascade of two Map-Reduce processes. <key,value> pair obtained as output of the first Map-Reduce phase is given as input to the second Map-

Reduce phase. In the second Map-Reduce phase, the mapper interchanges key and value. Sorting is performed by the reducer and only the highest value along with the article name is given as output. The first mapper accepts <infobox name, KEY> (infobox name is the key, whereas KEY is the value in the <key,value> pair). KEY denotes the attribute name as discussed in section III. The mapper produces as output a pair < infobox name, 1> corresponding to each input. The first reducer then counts the total number of attributes that were changed for an infobox. The output of Reducer is by default sorted in ascending order according to key (infobox name in our case). But we need to sort the value that indicates the total number of edits. So, we use another map-reduce. We have restricted the number of reducers in second map-reduce as one. Figure 3 shows an example dry run of Map-Reduce process.

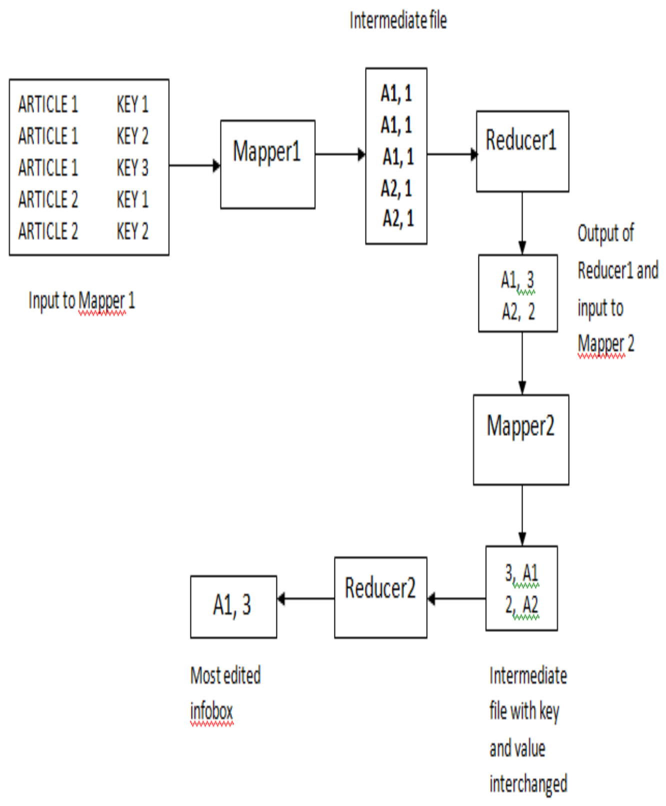


Fig.3: Example Map-Reduce process to find most edited infobox

**b) Update Alert to Wikipedia editors based on time series modeling**

The infoboxes of Wikipedia may require updates. Some may be updated very frequently and some after a long time. We intend to analyse the edit history dataset to find the time span after which the infobox is needed to be updated. For example, the president of India changes in every 5 years. Hence, the system must be capable of alerting the editors for update of President of India as the President changes after 5 years. So, we analyse the edit history dataset to find the update time for infoboxes. This is done using time series modeling. Consider for example there is an infobox ‘A’, which was updated on 12/12/92 then on 30/12/97 then on 29/12/2002 and so on. Time series modeling is used to find the difference between the dates of update. For infobox ‘A’ we can conclude that the update has to done in approximately 5 years. It is a two step process:

**Step 1:** To find the interval between consecutive updates
**Step 2:** Providing update Alert.
Next we discuss both the steps.

**Step 1:** To find the interval between consecutive updates

To find the interval between consecutive updates we need to covert the timestamp to date format. The input file has timestamp information. The dataset contains timestamps for various infoboxes in Unix style time format. These timestamps mark the time when the infobox was last edited. These timestamps need to be converted to date for further analysis. The java code snippet for conversion of timestamp to date is as follows:

```
Timestamp ts=new Timestamp (1317317616);
Date date=new Date(ts.getTime());
System.out.println(date);
```

The Timestamp class constructor takes the timestamp as an argument which is then converted into date format by the Date class Constructor.

**Constructor of Date class:** Date(long l).

For the input timestamp is 1317317616 the output date will be Fri Jan 16 11:25:17 IST 1970.

After finding the date for each update, we find the difference between consecutive updates to judge the approximate interval of updating of infobox or its attributes.

**Step 2: Providing update/edit Alert**

The editors of the infobox must be given an alert regarding the upcoming update before the interval. Say, if the interval of update is 5 years for an infobox ‘A’. Then we find all the editors ‘E’ of infobox ‘A’ from the dataset. All these editors (E) are sent the alert message regarding the upcoming update in infobox ‘A’. These alert messages can be sent before one month if the update interval is more or can be less depending on the update interval. The updates to the infoboxes are done by editors. Rather than sending alert to only the past editors of a particular infobox, we also try to find other editors who might be interested (or having knowledge of) in updating/ editing a particular infobox. To find the other editors to whom also the alert must be sent, we use the concept of bipartite graph. Bipartite graph is used to find the editors who have similar interests.

**c) Bipartite graph and update alert**

A bipartite graph, also called a bi-graph, is a set of graph vertices decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent [9]. We have used bipartite graph because it allows us to visualize the mapping of infoboxes with their editors to find similar users (editors) based on infobox which they have edited and suggest similar users/editors for updation of infoboxes of their topics of interests.
Consider for example,

- infobox A has editors a,b and c
- infobox B has editors b and d
- infobox C has editors e

Applying the concept of user-user based collaborative filtering [10] we find that all the users that edit the same infobox are similar users. So all the users who have edited infobox A will be similar {a,b,c}=S1. All the users who have edited infobox B will be similar {b,d}=S2 and for infobox c similar users will be {e}=S3. User a, b, c are similar according to S1. Also, users b and d are similar according to S2. So, we get the following similar users:

- S1={a,b,c,d} and S2={e}

Now, if we have to provide alert to editors of infobox A then we will send alert to a,b,c since they are past editors. The alert will also be sent to user d, as it is the similar user contained in the set S1.
Consider another example as shown in figure 4. The figure shows infobox name on LHS and users (editors) of the infobox. There is an arrow from infobox to user if the infobox is edited by the user in past. In the example, for infobox 1 the editors are user1 and user3. But users related to user1 and user3 include user 9; since, user 1 is similar to user 9 as they have edited infobox 4. So alert will be sent to user 1, user 3 and user 9.

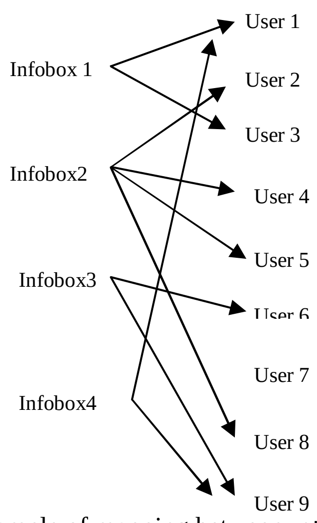


Fig.4: Example of mapping between infoboxes with their editors/users in form of Bipartite graph

**d) Finding vandalism from edit history dataset.**

The updates done in the infobox may be incorrect. These updates may then be corrected. In this phase, we find the vandalism in the updates. To find the vandalism we check if the update was done and then redone very frequently, generally before the update time.

## III. RESULTS AND DISCUSSION

The system is implemented using Big data technique of Map-Reduce in Apace Hadoop environment. Hadoop is installed in Pseudo-distributed mode. The edit history dataset is then loaded into Hadoop Distributed File System (HDFS).

### *A. Finding the most edited infobox*

The Mapper finds the infobox nameand writes the it along with 1 using the method context.write(new Text(e[1]),one). Then the reducer sums up the counts for each infobox name (it is considered as key) using the logic given below in the reducer 1

```
int count = 0;
for (IntWritable value:values)
{
count += value.get();
}
context.write(key, new IntWritable(count));
}
```

Then output of reducer 1 is given to Mapper 2 for further computation as discussed in Section IV.

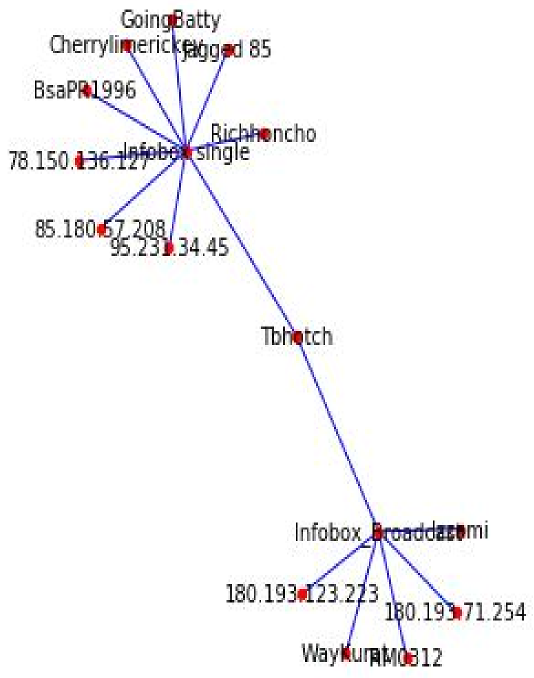


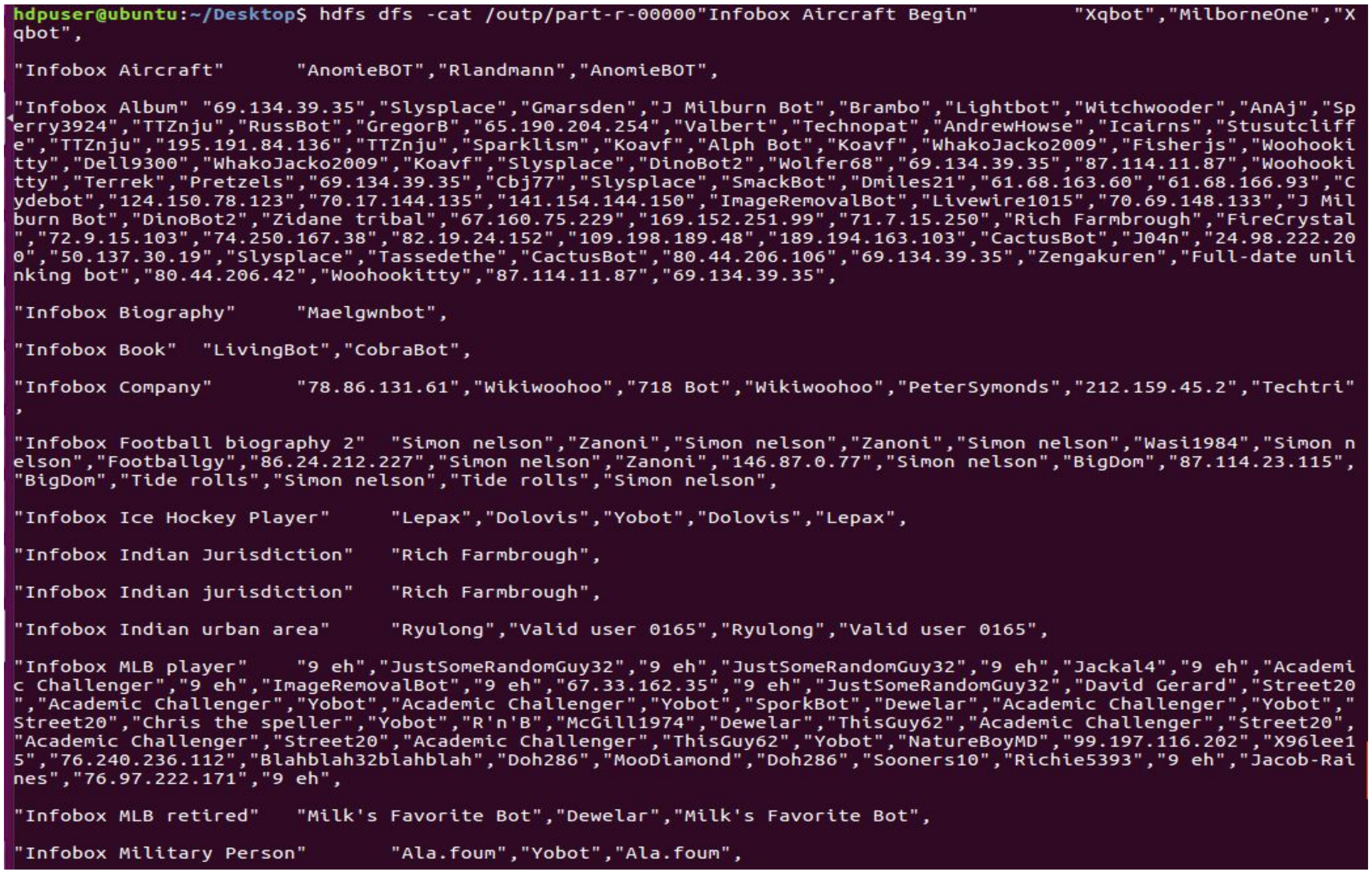
Fig. 5: Portion of bipartite graph

### *B. Bipartite Graph*

To find the similar users, the mapper extracts the infobox_name and contributor_ip fields from the input edit history dataset. The mapper method takes the input Text (value) as string (line).The variables s and s2 are used to find the values of infobox name and the contributors_ip (user or editor). The input Text is split on the "," character which is then checked for s and s2. After finding s or s2 the word is split again on the ":" character. When s is found the mapper writes the output in the format <infobox name,user>. The Iterable<Text> values is used to traverse the object produced by the mapper function which then sorts these objects into <infobox name,user1,user2,…,userN>. These users are similar users.As the Iterable<Text> is traversed a string variable duplicate is used to discard duplicate users. The reducer then writes the output as <key,user1,user2,…,userN> where key is the infobox name. A snippet of code is as below:

```
   String users = "",duplicate = "";
      for(Text value:values) {
                  if(!duplicate.equals(value.toString())) {
                                    users += value + ",";
                                    duplicate =
   value.toString();
                  }
             }
             context.write(key, new Text(users + "\n"));
```

From the file obtained by the Reducer class we created a visual representation of bipartite graph which maps the infobox name with the similar users of that infobox. The visual representation is created with help of "python's matplotlib" library. A small portion of Bipartite graph is as shown in figure 5. Output on a subset of dataset is shown in figure 6.



```
hdpuser@ubuntu:~/Desktop$ hdfs dfs -cat /outp/part-r-00000"Infobox Aircraft Begin"        "Xqbot","MilborneOne","X
qbot",

"Infobox Aircraft"      "AnomieBOT","Rlandmann","AnomieBOT",

"Infobox Album" "69.134.39.35","Slysplace","Gmarsden","J Milburn Bot","Brambo","Lightbot","Witchwooder","AnAj","Sp
erry3924","TTZnju","RussBot","GregorB","65.190.204.254","Valbert","Technopat","AndrewHowse","Icairns","Stusutcliff
e","TTZnju","195.191.84.136","TTZnju","Sparklism","Koavf","Alph Bot","Koavf","WhakoJacko2009","Fisherjs","Woohooki
tty","Dell9300","WhakoJacko2009","Koavf","Slysplace","DinoBot2","Wolfer68","69.134.39.35","87.114.11.87","Woohooki
tty","Terrek","Pretzels","69.134.39.35","Cbj77","Slysplace","SmackBot","Dmiles21","61.68.163.60","61.68.166.93","C
ydebot","124.150.78.123","70.17.144.135","141.154.144.150","ImageRemovalBot","Livewire1015","70.69.148.133","J Mil
burn Bot","DinoBot2","Zidane tribal","67.160.75.229","169.152.251.99","71.7.15.250","Rich Farmbrough","FireCrystal
","72.9.15.103","74.250.167.38","82.19.24.152","109.198.189.48","189.194.163.103","CactusBot","J04n","24.98.222.20
0","50.137.30.19","Slysplace","Tassedethe","CactusBot","80.44.206.106","69.134.39.35","Zengakuren","Full-date unli
nking bot","80.44.206.42","Woohookitty","87.114.11.87","69.134.39.35",

"Infobox Biography"     "Maelgwnbot",

"Infobox Book"  "LivingBot","CobraBot",

"Infobox Company"       "78.86.131.61","Wikiwoohoo","718 Bot","Wikiwoohoo","PeterSymonds","212.159.45.2","Techtri"
,

"Infobox Football biography 2"  "Simon nelson","Zanoni","Simon nelson","Zanoni","Simon nelson","Wasi1984","Simon n
elson","Footballgy","86.24.212.227","Simon nelson","Zanoni","146.87.0.77","Simon nelson","BigDom","87.114.23.115",
"BigDom","Tide rolls","Simon nelson","Tide rolls","Simon nelson",

"Infobox Ice Hockey Player"     "Lepax","Dolovis","Yobot","Dolovis","Lepax",

"Infobox Indian Jurisdiction"   "Rich Farmbrough",

"Infobox Indian jurisdiction"   "Rich Farmbrough",

"Infobox Indian urban area"     "Ryulong","Valid user 0165","Ryulong","Valid user 0165",

"Infobox MLB player"    "9 eh","JustSomeRandomGuy32","9 eh","JustSomeRandomGuy32","9 eh","Jackal4","9 eh","Academi
c Challenger","9 eh","ImageRemovalBot","9 eh","67.33.162.35","9 eh","JustSomeRandomGuy32","David Gerard","Street20
","Academic Challenger","Yobot","Academic Challenger","Yobot","SporkBot","Dewelar","Academic Challenger","Yobot","
Street20","Chris the speller","Yobot","R'n'B","McGill1974","Dewelar","ThisGuy62","Academic Challenger","Street20",
"Academic Challenger","Street20","Academic Challenger","ThisGuy62","Yobot","NatureBoyMD","99.197.116.202","X96lee1
5","76.240.236.112","Blahblah32blahblah","Doh286","MooDiamond","Doh286","Sooners10","Richie5393","9 eh","Jacob-Rai
nes","76.97.222.171","9 eh",

"Infobox MLB retired"   "Milk's Favorite Bot","Dewelar","Milk's Favorite Bot",

"Infobox Military Person"       "Ala.foum","Yobot","Ala.foum",
```

Fig. 6: Infobox and the users (similar users)

### *B. Finding vandalism from edit history dataset*

The result of vandalic edits done by registered users and from unregistered users (represented by IP address) is as shown in figure 7.

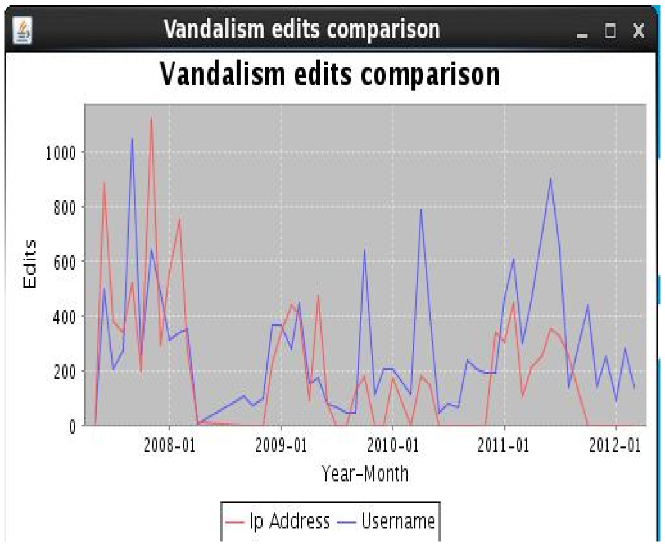


Fig.7: Graph showing comparative study between vandalic edits and genuine edits

## IV. CONCLUSION AND FUTURE SCOPE

The most edited infobox was found successfully. Further, analysis can be carried out on individual infoboxes and knowledge bases can be developed using infobox information. Update alert system was developed to alert the Wikipedia editors for upcoming alerts based on time series modeling. Concept of user-based collaborative filtering was used to find similar editors and provide alert to editors as well as to the similar editors. It was found that the history dataset contains vandalic updates due to which the update frequency of some infoboxes could not be found. To solve this issue an automated system can be developed based on Natural language processing (NLP) which can fetch the latest news and tweets and correlate them to the infobox to alert the editors about recent happening related to the infobox and suggest updation. For many of the articles there can be rapid change in the information, leading to development of automatic methods to update Wikipedia's content to ensure that it contains the latest information. This real time system can assist in proving the latest values to the attributes of the infobox. Moreover, vandalic edits can also be captured if NLP based system is designed.

### *A. Recommendations*

In this section we present various research scope in analysis of infobox, analysis and need of structured data.

- **Generation of knowledge bases by extraction of information from Wikipedia infobox**

A knowledge base is a collection of facts and reasoning mechanisms. Many natural language systems rely on knowledge bases for various tasks such as selection of solution and finding inferences. Knowledge-based selection is to resolve ambiguity problem in Natural Language Processing (NLP) system, where the systemmust choose between two or more possible solutions. However, there may be situations where there are zero choices to select from or there exist a gap inthe input. In such situations, knowledge-based inference is applied to infer missing information [1]. A knowledge base for real-world language processing applications should consist of a largebase of facts and reasoning mechanisms that combine them to induce novel and morecomplex information. Knowledge bases can be generated using the information proving websites over the internet by use of web crawlers. In [2], knowledge base is build using Wikipedia's existing network of categories. Rather than extraction of data from unstructured data, it is easy to extract only important information from the structured data provided by these websites. Infobox is one such structured table provided by Wikipedia.

- **Generation of ontologies.**

An ontology provides an explicit specification of real world concepts composed of classes, subclasses and instances and the relationship between them through properties and values. Concept in ontology represent any sort of object or entity. A relationship in an ontology represents a way in which two concepts can be connected to each other. The connection may representsome allegiance [3]. Ontologies are useful in various NLP tasks such as data summarization and relation extraction. The resolution of infoboxes into elements has given rise to new ontology like DBPedia and Freebase. Ontology-driven generation of wiki content and interfaces in presented in [4]. They represent and refactor information in correct, systematic and homogenous format maintaining the base integrity of articles.

- **Machine learning Applications**

These infoboxes have also been successful in paving their way in the domain of machine learning. The information obtained from infoboxes are used for training a supervised machine learning algorithm.

- **Helping in developing question-answer system (bots for auto Q-A)**

Question Answering (QA) is an area of natural language processing aimed at providing humanusers with an automated system for answering their queries. In a more elaborate manner, question answering is the task of asking questions in natural language by the user and automated answering by the system by finding and scanning the relevant articles. Such systems use the concept of information retrieval. A list of indexed web page is returned by the process of information retrieval. There is a need to develop accurate systems for automated question answering as there are a lot of structured knowledge-bases and the continuous demand to access information rapidly and efficiently. In [5], a Dbpedia based factoid question-answering system is presented. An open-domain QA system using a Wikipedia-based knowledge model is proposed in [6]. They used the combination of article content with varied other answer matching modules based on different types

of semi-structured knowledge such as infoboxes, article structure, category structure,and definitions.

- **Vandalism**

Vandalism is the act of editing any information in a malicious manner that is intentionally disruptive. It may include addition, removal, or modification of the text or other material that is either nonsensical, hoax, or that is an offensive, or otherwise incorrect in any manner [7]. This leads to incorrect information and is undesirable. Every editable article on Wikipedia has an associated page edit history. The use of edit history as data source comes with the challenge of dealing with the cost of processing such a vast edit history as well as dealing with errors which reduce data quality and reliability. Some of these errors are a repercussion of vandalism. Authors in [8], propose to the technique for historical structured data extraction and vandalism detectionfrom the Wikipedia edit history.Vandalbots and computer programs can be designed to detect and revert vandalism. ClueBot and VoABot II are some of the efforts in the direction. Machine learning techniques can also be used to build automated systems capable of distinguishing between legitimate edits and vandalic edits [7].

- **Handling large amount of infobox data**
- **Analysis of infoboxes**

## REFERENCES

[1]K. Mahesh, S. Nirenburg, “Knowledge-based systems for natural language processing,” New Mexico State University, Computing Research Laboratory, 1996.

[2] V. Nastase, M. Strube, “Transforming Wikipedia into a large scale multilingual concept network,” Artificial Intelligence, 194, 2013, pp.62-85.

[3]M.Synak, M. Dabrowski, S.R.Kruk, “Semantic web and ontologies,” In Semantic Digital Libraries, 2009, Springer, Berlin, Heidelberg, pp. 41-54.

[4]A. Di Iorio, A. Musetti, S. Peroni, F. Vitali, “Ontology-driven generation of wiki content and interfaces,” New Review of Hypermedia and Multimedia, 16(1-2), 2010, pp. 9-31.

[5]A. Tahri, O. Tibermacine, “DBPedia based factoid question answering system. International Journal of Web & Semantic Technology,” Vol. 4(3), 2013, p.23.

[6] Chen, Danqi, Adam Fisch, Jason Weston, and Antoine Bordes. "Reading wikipedia to answer open-domain questions." arXiv preprint arXiv:1704.00051 (2017).

[7] K. Smets, B. Goethals, B. Verdonk, “Automatic vandalism detection in Wikipedia: Towards a machine learning approach,” in AAAI workshop on Wikipedia and artificial intelligence: An Evolving Synergy, July, 2008, pp. 43-48.

[8] E. Alfonseca, G. Garrido, J.Y.Delort, A. Peñas, “WHAD: Wikipedia historical attributes data,” Language Resources and Evaluation, 47(4), Springer, 2013, pp.1163-1190.

[9] E.W.Weisstein, Complete Bipartite Graph, 2002.

[10] Z.D. Zhao, M.S. Shang, “User-based collaborative-filtering recommendation algorithms on hadoop,” In Third International Conference on Knowledge Discovery and Data Mining (WKDD'10), IEEE, Jan 2010, pp. 478-481.