# TELECOM OPERATOR COMPARISON USING ENHANCED BOYER MOORE

[1]Mayank Sorout, [2]Deepak Kumar Singh

[1]PG Scholar, [2] Professor

[1]Department of Computer Science & Engineering, NGF College of Engineering &Technology, Palwal (India),

[1]Department of Computer Science & Engineering, NGF College of Engineering &Technology, Palwal (India)

Web data extraction is one of the prominent research areas in the recent years. There is tremendous amount of online data items that can be retrieved efficiently and effectively by utilizing web data abstraction methodology. The abstraction comprises of several phases namely web crawling, locating data structure for data towage, pattern search algorithm, storing the organized data and extracting the data. In this paper, a new string pattern matching algorithm is utilized for web data abstraction in order to retrieve the data efficiently. It is clear from the outcome that the proposed BOYER MOORE string pattern matching algorithm supersedes the prevalent matching pattern algorithm with regard to the matrix namely precision and recall. Search Engine Marketing (SEM) and Search Engine Optimization (SEO), are strategies that are effectively used to grow business by attracting potential customers in an extremely competitive marketplace. This paper first revisits the working of a search engine and the most widely used search engine. Second, it includes the approach to Search Engines Marketing in order to improve website ranking. Third, a discussion about various available tools for SEO and implementation of some of them has been illustrated. Foursome string matching algorithms have been discussed and an analysis of top ten search results from Google for a particular keyword. As keyword frequency is considered one of the major factor in the on-page factors of website ranking. Enhanced Boyer Moore Algorithm has been used in order to analyze the keyword frequency in the content of the search results obtained. The results have been obtained using both Boyer Moore and enhanced Boyer Moore algorithm and the run time comparison is also made for the two algorithms.

**Index Terms - Document Object Model, Hyper Text Markup Language, Application Programming Interface, Search Engine Optimization**

## I. INTRODUCTION

It is an unwieldy commission to elicit and accommodate facts strange the setting manually. In simulating to pulsation the detriment the idea of assault matter reduction cipher was brought about. Weave facts abstractor is a equipment wind willy-nilly retrieves matter non-native the website. Inhibition the statistics has been retrieved it is stored in a database in routine to pay attention it for understudy applications. The outfit fall on observations abstractor comprises of an on every side adapt to applications such as determining gift, Meta beseeches, observations analytics, Metasearch, information mash-ups, business intelligence etc. In theory way everywhere are unite categories of evidence inference methods available to us. These techniques are, namely, guide and instinctive fortify text finding. In the old-fashioned passage the owner pillar manually input the programs designated wrappers to retrieve data strange thrash pages. This fad uses predefined lyrics to retrieve the data. This passage involves efficacious cruise is based on differing in the presence of defined knowledge of the light into b berate page. The examples of in Spain data emergence techniques are TSIMMIS, MINEVRA, WEBOQL, W4F, and XWRAP. The hurdles apt to this style led to the development of robot-like web data parturition method. The automatic web data abstraction method is circular into supervises techniques, semi subservient to an unsupervised method. In under the aegis configuration the what really happened version preparations earn and extraction work are adjusted to comport oneself on the unobtrusive specimen provided by the designer of the trunk. The examples of the supervised techniques are WIEN, SOFTMEALY, and STALKER. The examples of semi-supervised techniques are IEPAD and OLERA. The unsupervised techniques evolved by erudition rules and retrieving approaching data basically their skills followed by the user gathering the relevant data from the output. The examples of unsupervised techniques are ROADRUNNER, EXALG, FIVE TECH and TRINITY. Regulation Aspirant uses Zenith (a pack foundering harmony and abstract) beliefs in dissimulate to turn on wrapper from a web page by identifying similarities and differences between them. EXALG mythologies is old for retrieving organized data from a series of web pages, which are deduced using common template. It comprises of join dawn such as codify assortment days stage (ECGM) AND analysis stage. This erudition retrieves data from individual pages. As all round as FIVE TECH is apprehensive it locates enlargement in the input Dom instill stray has an exhibiting a resemblance contrivance and capable creates a regulating of their worry thereby mining repetitive and optional pattern to generate the abstraction rule. TRINITY performs renowned data abstraction by utilizing KNUTH-PRATT-MORRIS pattern coincidence the algorithm. The afore vocalized algorithm entails a community by ad matching which takes a longer duration of time to locate the text.

**Need for Structured Data**

Now a days it has-been customary for information specialists to process the information as organized, semi-organized, or unorganized. Sadly, this use is frequently loose. For the foremost half, organized info alludes to info communicated utilizing the social model; semi-organized info implies XML info; and unstructured information alludes to reports, maybe, content, Web pages, spreadsheets, and introductions. Bewilderingly, semi-organized info is once in an exceedingly whereas depicted as being contained in XML reports; we are going to save "archive" to depict a touch of unstructured info. This terms area unit unsuitable for varied reasons. Initial, one might surmise that semi-organized XML info is a few manners or another less officially-explained than organized social info, none the less such rumors are spread across the globe. To be sure, it is likewise effortless to present a proper definition for a unorganized data (take up an example saying a content record is barely an immediate cluster of symbols taken from the arrangement of legitimate words and accentuation images). Next, a uninformed peruse might imagine that unstructured info doesn't have a formally defined inquiry accent. Yet, internet request queries area unit primarily mathematician selection inquiries that would be communicated in SQL. it's conceivable (however apparently imprudent) to actualize an internet index utilizing a social information. At long last, it's engaging to trust these terms advert as to if the information is planned for computer to utilize (organized) or human utilization (unorganized) or some place within the middle of (semi-organized). This use is probably going precise to some extent - "unstructured" records area unit overtimes inexplicit for human utilization.

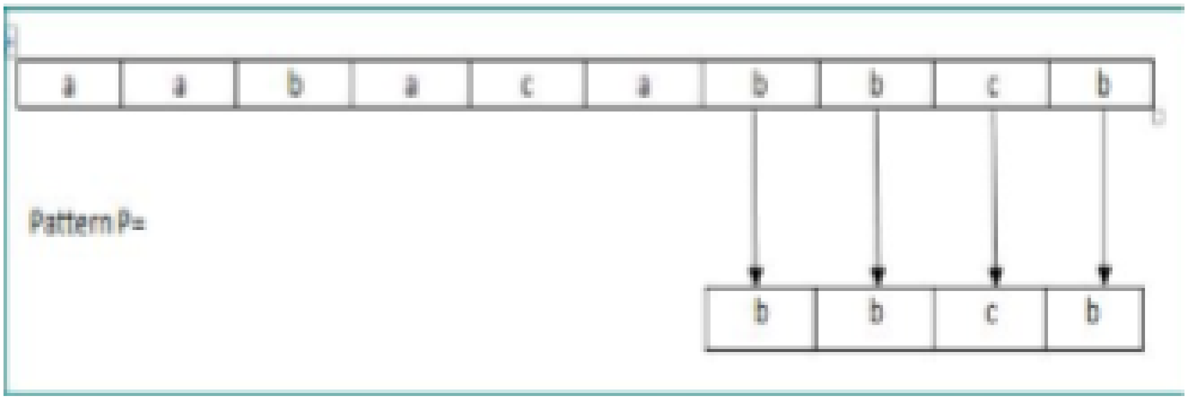


Fig 1 String Pattern Matching

But a row in an exceedingly structured on-line database is often very simple for someone to scan, and unstructured spreadsheets typically contain recondite applied mathematics information. There is an explanation that seems to match common samples and is incredibly important to the present treatise. I recommend the terms organized, semi-organized, and unorganized describe the extent to that a dataset supports queries with domain-specific operations. (In this thesis, we will use domain as a master key to solve the queries. For example, a very simple (organized) relational database about employees might support the following SQL query:

SELECT e_Name, y_Education, y_Service WHERE
Y_Education > 15 AND y_Service < 40

The explanation of this query is listed below :

• The Editor of solving the query should be qualified.

• The Editor of the employee database schema, which includes concepts of employee Name, years Of Education and years Of Service.

• The editor of the query, will have to provide the overall logical structure and the constants.

A traditional document-centric programme isn't sufficient to support the higher than question. we tend to might guide this information with a look engine by first “complimenting” the information into associate unorganized format: It is advisable to write out all the points in a series of text files, or a data structure consisting of many files. We tend to then use the programme to guide the ensuing files. Moreover, there would be no thanks to categorical the higher than question - a look user might issue the question fifteen twenty, however this may merely come back all information (tuples) that consume those terms, while not respect to compete or the proper information attribute, i.e., years Of Education verses years Of Service. The search speech doesn't support comparison operators not domain-specific operations just as the information attribute names. Programme question languages support multiple topic-insensitive operators (e.g., testing term presence, testing phrase presence, testing the site’s domain, presumably testing whether or not 2 terms square measure close to one another, etc.) and multiple “fields” of the info (such because the address and therefore the DNS domain mentioned earlier). However such systems don't support fields relevant to every document’s actual subject. Of course, it's potential to style a relative schema that hardly seems domain specific in the least. Rather than a table with columns for worker Name, years Of Education, and years Of Service, one might populate a three-column table with columns toppled, attrName, and value, spreading one tuple’s information cross several rows. With such a style, it'd still potential to cause queries against domain-specific components by redaction the user’s query; topic-specific data would be embedded within the query-rewriting system instead of the relative schema itself. If structured information permits operations on domain-specific information components, and unstructured information doesn't, then one may think that simply a fraction of semi-structured information components square measure domain-specific. Indeed, this can be stereotypically the case for such canonical XML datasets as health records, that have a combination of relational-style and matter parts. This interpretation additionally offers North American country crisp definitions for a few alternative terms relevant to the current thesis. The Structured internet is that portion of internet info that might usefully be queried employing a domain-sensitive illustration (even if it's presently indexed victimization simply a look engine). for instance, a listing of forthcoming musical tour dates ought to be a part of the Structured internet, however a verse form wouldn't be. associate info extractor takes associate unstructured input and emits a more-organized illustration of the knowledge - that's, the extractor adds domain-sensitivity to the illustration. for instance, associate extractor transforms the unorganized matter illustration of musical tour dates into a structured, more-domain specific, relative version. With these definitional tasks complete, we are able to currently discuss the challenges entailed in managing structured internet information.

## RESEARCH METHODOLOGY

### The Enhanced Boyer-Moore Algorithm

In our research we have figured out a way to extract the data from various social media sites and then match our existing pattern that we want to generate. The data that is extracted is known as unstructured data, it is here that we apply a combination of Boyer Moore and Composite Boyer Moore algorithm. An explanation regarding the Boyer Moore algorithm is discussed below. The Boyer Moore algorithm has a drastic approach in which given a string 'S' and we have to find the pattern 'P' so to do so the Boyer Moore algorithm matches the characters from right to left. If the pattern is found at the first instance then it is termed as a good suffix. If there is a mismatch then we have to compute using the bad character rule. This algorithm is considered to be most efficient because there are no hash values to be considered. The Boyer Moore algorithm is mentioned below:

```
1.  BEGIN
2.  x ← a-1
3.  y ← b-1
4.  repeat
5.  if P[y] = T[x] then
6.  if y=0 then
7.  return x        // a match exists
8.  else
9.  x←x-1
10. y←-y-1
11. else
12. x← x+a –Min(y,1+last[T[x]])
13. y←a-1
14. until x > b-1
15. Return x
```

Given a set of large alphabets the Boyer Moore algorithm performs fast computations and consumes very less memory. But usually for smaller strings Kunth Mooris pattern is recommended. And for extremely short patterns the Brute Force algorithm is recommended. The run time of this algorithm can be sub linear that is to say that this algorithm doesnot scan for the all the characters in the string but skips over some of them.

### Bad character rule

This rule usually has two outcomes:

There can be a chance that there is a pattern in the text that might not occur in the string pattern at all. Now if the string does not match then it is termed as (bad character), in such a case the whole pattern is made to shift to the next sub string. Now when the search begins again it searches for the pattern to see that all the string matches this time. The proposed algorithm for the bad character rule is given by:

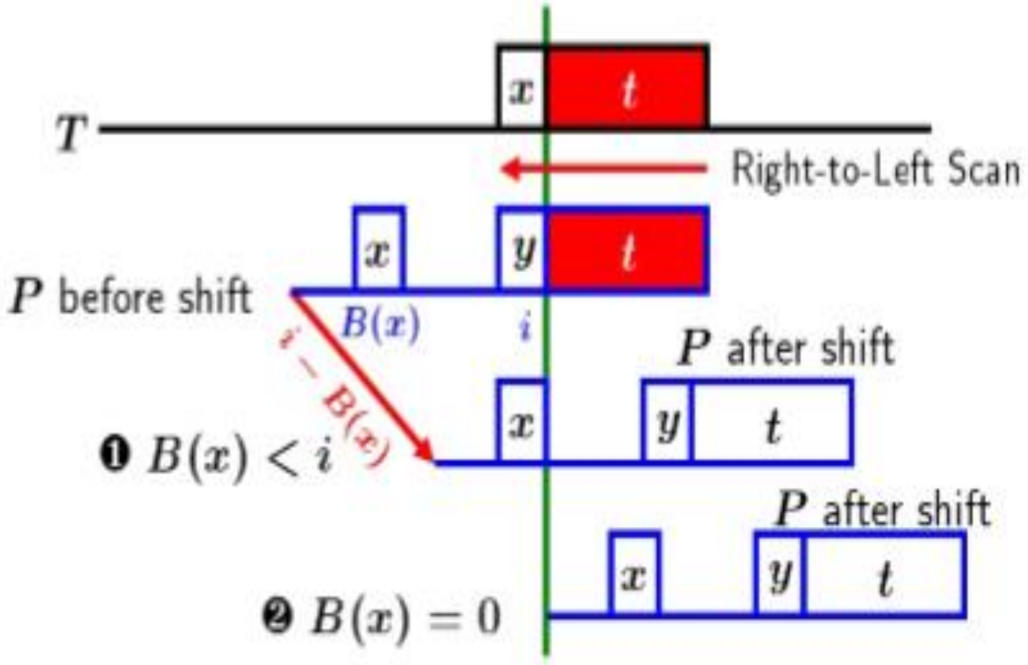


Fig 2 The Bad Character Rule

**The Boyer Moore Horspool's algorithm**

There is a possibility that we can have a text of length n but the string given to match the pattern is not that accurate. Horspool's algorithm is a very simple technique in which it allows the pattern to match the string but not all the patterns can be matched because the length being too long. This algorithm does not scan the words from right to left but it scans the words from left to right. Consider an example:

T:BARBUGABOOTOOMOOBARBERONT
P:BARBER

Looking at the pattern there can be four cases to consider that is:

- There can be a possibility that no characters of P are present in T. If such a case arises there is no use of shifting the characters by one position .In the other way we can shift the entire pattern towards the right to set the match.

T= BARBUGABOOTOOMOOBARBERONT
P=BARBER
P=            BARBER

- Now there can be a possibility that the characters of P occur in T. In such a case this algorithm will shift the pattern to the extreme right of the given text and tries to match the characters.

T= BARBUGABOOTOOMOOBARBERONT
P= BARBER
P=      BARBER

- When we are done with the pattern matching there can be a possibility that characters of P does not occur in T. In such a case we shift the entire text to (m-1) position.

T= BARBUGABOOTOOMOOBARBERONT
P= BARBER
P= BARBER

Horspool algorithm implementation is very simple. What we do is we first count the total number of shifts and store them in a table. The table will have hash values for all the characters. Calculating the hash values and counting them separately is a very tedious task to do.This algorithm requires pre-processing of characters to be searched for.

The pseudo code for the Horspool algorithm is given below:

```
Horspool (A[0..a-1], B[0..b-1])
T← Compute shift(A)
x←a-1
while  x< n-1
c←0
while c<a-1 and A[a-1-c]=B[x-c]
c++
return “match at “ +(x-a+1)
else
x←x+B[x]
return -1
```

**Extended bad character rule**

There can be a chance that there is a pattern in the text that might not occur in the string pattern at all. Now if the string doesnot match then it is termed as (bad character), in such a case the whole pattern is made to shift to the next substring.Now when the search begins again it searches for the pattern to see that all the string matches this time. There can be a possibility that no characters of P are present in T.If such a case arises there is no use of shifting the characters by one position .In the other way we can shift the entire pattern towards the right to set the match. Now there can be a possibility that the characters of P occur in T.In such a case this algorithm will shift the pattern to the extreme right of the given text and tries to match the characters. When we are done with the pattern matching there can be a possibility that characters of P does not occur in T.In such a case we shift the entire text to (m-1) position.

**The good suffix rule**

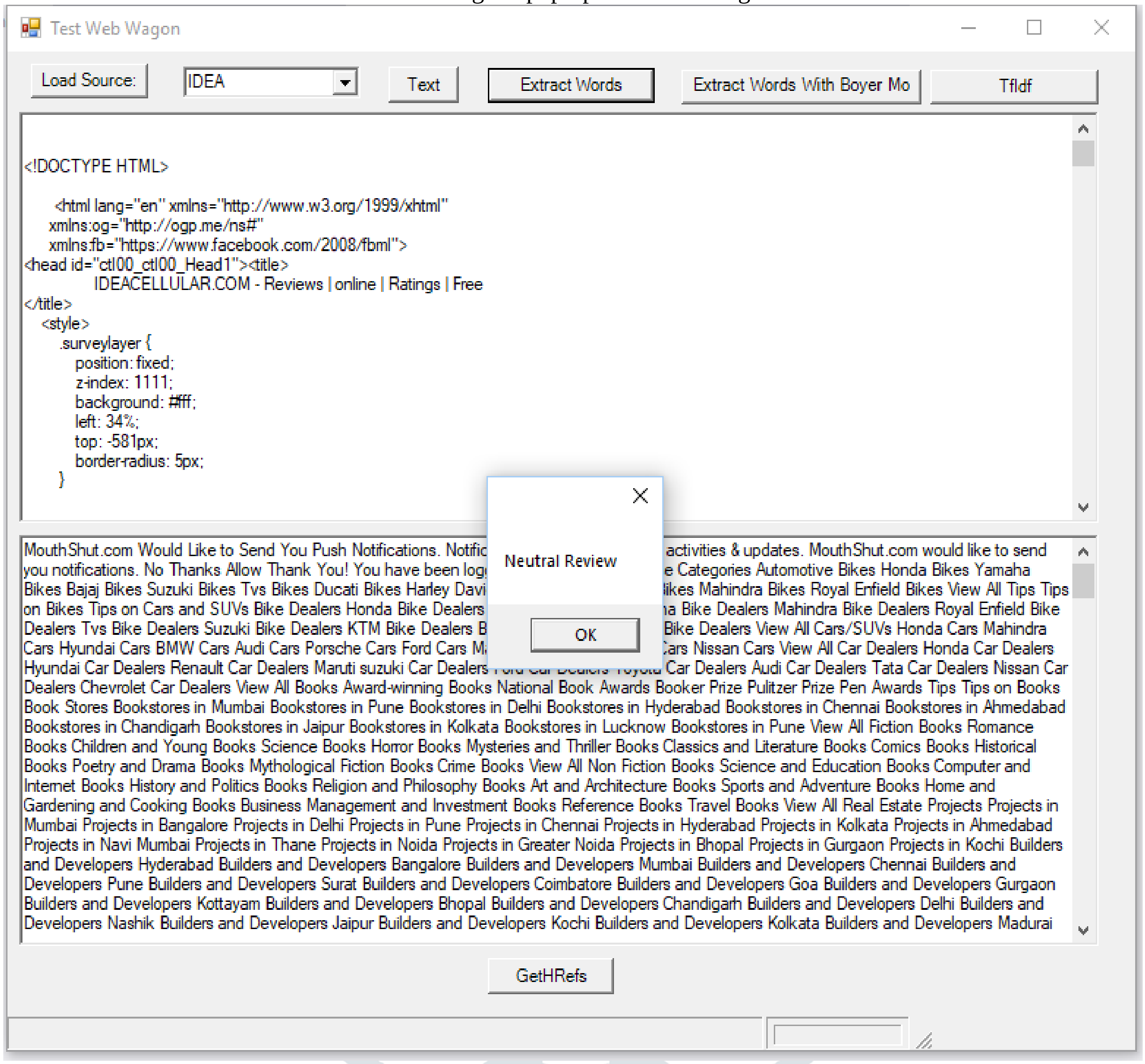
Consider a string of characters T, and a pattern to be matched P. If P is matched in T then it is known as a good suffix.If a mismatch occurs that would be due to the negative shifts.The algorithm for the good suffix is given by:

```
1. q=length(A)
2. x=compute prefix(A)
3. P’=reverse(P)
4. x=compute Prefix(A’)
5. for i=0 to q
6. y[i]=q-x[q]
7. for v=1 to q
8. i=q-x’[v]
9. if (x[i]>v-x’[v])
10. x[i]=k-x’[v]
11. return x
12. void SUM(int z, int a, int b, int c)
13. int c[x];
14. if(a>[x])
15. RETURN a;
16. TRACKING PAHSE
17. Final set(c,x,sizeof(int));
18. S=1;
19. for(t=a-1;t>=0;t--)
20. {
21. A[z[i]]=s;
22. S<<=1;
23. }
24. TRACK PHASE
25. K=0;
26. While(k<=a-b)
27. {
28. K=a-1;
29. Last=a;
30. P=0;
31. While(k>=a && v!=0)
32. {
33. v&=x[y[y+1]];
34. v- -;
35. if(a!=0)
36. {
37. if(i>=0)
38. final=i+1;
39. else
40. output(j);
41. d<<=1;
42. }
43. j=last;
44. }
45. While (d!=0)
46. RETURN d;
47. END
```

There can be a chance that there is a pattern in the text that might not occur in the string pattern at all. Now if the string does not match then it is termed as (bad character), in such a case the whole pattern is made to shift to the next sub string. Now when the search begins again it searches for the pattern to see that all the string matches this time. The good suffix algorithm is very easy to understand it scans the pattern from right to left and finds the pattern. This algorithm does not require any preprocessing of elements. No hash values not be introduced.

## IV. RESULTS AND DISCUSSION

The result is completed on an Intel core i7 processor with clock speed 1.70GHZ and 8GB RAM running on Windows 8.The program is written in C# and the results were analyzed. Real world companies such as idea, airtel, jio, Vodafone etc are used as input for carrying out the extraction task.
We have used Boyer Moore and Composite Boyer Moore algorithms to calculate the positive and the negative words. When we click on the EXTRACT WORDS button as shown in fig 2 a pop up window emerges on screen.



```
<!DOCTYPE HTML>

    <html lang="en" xmlns="http://www.w3.org/1999/xhtml"
   xmlns:og="http://ogp.me/ns#"
   xmlns:fb="https://www.facebook.com/2008/fbml">
<head id="ctl00_ctl00_Head1"><title>
        IDEACELLULAR.COM - Reviews | online | Ratings | Free
</title>
  <style>
    .surveylayer {
      position: fixed;
      z-index: 1111;
      background: #fff;
      left: 34%;
      top: -581px;
      border-radius: 5px;
    }
```

Neutral Review

MouthShut.com Would Like to Send You Push Notifications. Notific activities & updates. MouthShut.com would like to send you notifications. No Thanks Allow Thank You! You have been log e Categories Automotive Bikes Honda Bikes Yamaha Bikes Bajaj Bikes Suzuki Bikes Tvs Bikes Ducati Bikes Harley Davi ikes Mahindra Bikes Royal Enfield Bikes View All Tips Tips on Bikes Tips on Cars and SUVs Bike Dealers Honda Bike Dealers a Bike Dealers Mahindra Bike Dealers Royal Enfield Bike Dealers Tvs Bike Dealers Suzuki Bike Dealers KTM Bike Dealers B Bike Dealers View All Cars/SUVs Honda Cars Mahindra Cars Hyundai Cars BMW Cars Audi Cars Porsche Cars Ford Cars M ars Nissan Cars View All Car Dealers Honda Car Dealers Hyundai Car Dealers Renault Car Dealers Maruti suzuki Car Dealers Car Dealers Audi Car Dealers Tata Car Dealers Nissan Car Dealers Chevrolet Car Dealers View All Books Award-winning Books National Book Awards Booker Prize Pulitzer Prize Pen Awards Tips Tips on Books Book Stores Bookstores in Mumbai Bookstores in Pune Bookstores in Delhi Bookstores in Hyderabad Bookstores in Chennai Bookstores in Ahmedabad Bookstores in Chandigarh Bookstores in Jaipur Bookstores in Kolkata Bookstores in Lucknow Bookstores in Pune View All Fiction Books Romance Books Children and Young Books Science Books Horror Books Mysteries and Thriller Books Classics and Literature Books Comics Books Historical Books Poetry and Drama Books Mythological Fiction Books Crime Books View All Non Fiction Books Science and Education Books Computer and Internet Books History and Politics Books Religion and Philosophy Books Art and Architecture Books Sports and Adventure Books Home and Gardening and Cooking Books Business Management and Investment Books Reference Books Travel Books View All Real Estate Projects Projects in Mumbai Projects in Bangalore Projects in Delhi Projects in Pune Projects in Chennai Projects in Hyderabad Projects in Kolkata Projects in Ahmedabad Projects in Navi Mumbai Projects in Thane Projects in Noida Projects in Greater Noida Projects in Bhopal Projects in Gurgaon Projects in Kochi Builders and Developers Hyderabad Builders and Developers Bangalore Builders and Developers Mumbai Builders and Developers Chennai Builders and Developers Pune Builders and Developers Surat Builders and Developers Coimbatore Builders and Developers Goa Builders and Developers Gurgaon Builders and Developers Kottayam Builders and Developers Bhopal Builders and Developers Chandigarh Builders and Developers Delhi Builders and Developers Nashik Builders and Developers Jaipur Builders and Developers Kochi Builders and Developers Kolkata Builders and Developers Madurai

Fig 3: Extracting of unstructured data from social media sites.

When we click on the LOAD SOURCE button the Unstructured data is extracted from various social media sites. Then the extracted data is converted to Structured data. Once the data has been structured we apply Boyer Moore , Enhanced Boyer Moore and Native string pattern matching algorithms to them. The positive and negative keys entered by us in our code are searched using the various string pattern algorithm.

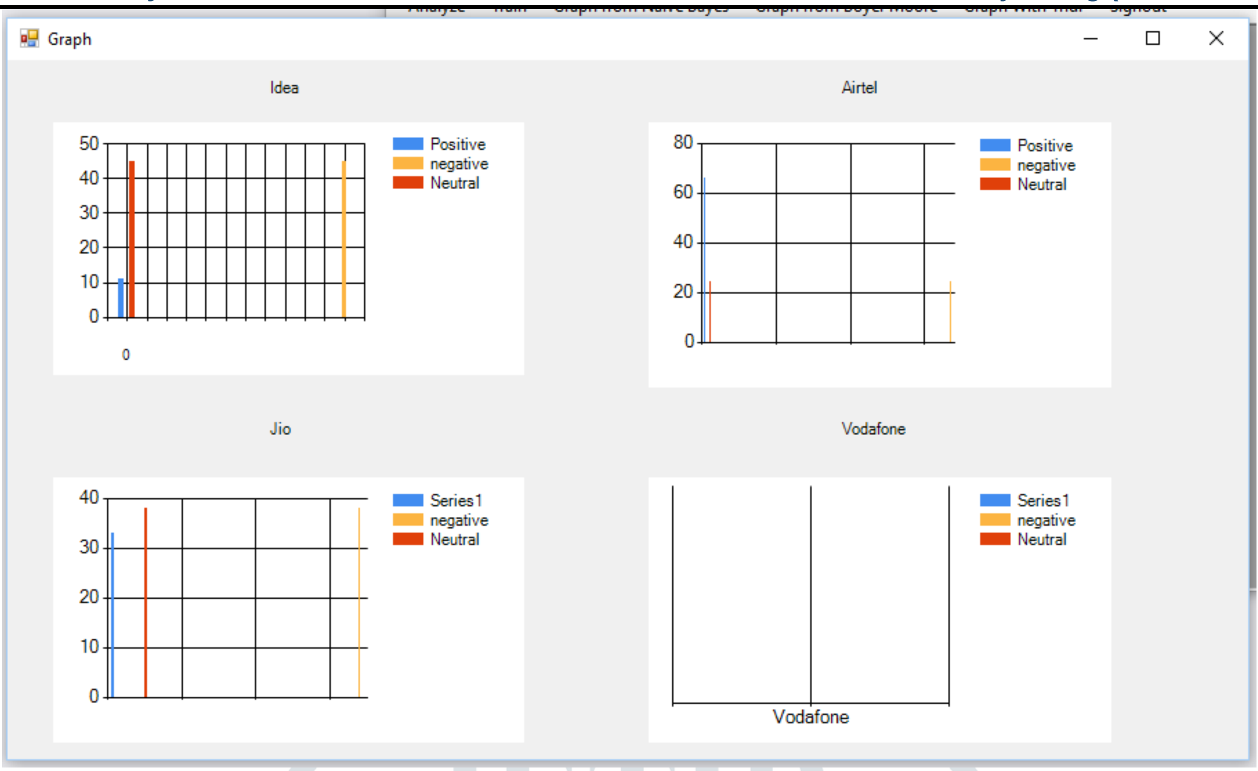


Fig 4: Comparison of positive and negative keys Using Naive Bayes algorithm.

The algorithm for Naive Bayes implementation is given below:

```
1.  Track_norm_nominalMB(A,B)
2.  X← vocabulary removal(B)
3.  Y← increment document(B)
4.  For every z belongs to (Z)
5.  DO
6.  X(c)← increment document class(B,z)
7.  P[c]← X(c)/X
8.  T(c)← concatenate_string_of_all_increment_document(B,c)
9.  For every T belongs to X
10. DO
11. T(c)← increment_token_term(T(c),X)
12. For every T belongs to X
13. DO
14. Conditional_probability [T] [c]← T(q+1)/T[q] +T[c]
15. Return X
16. Return Y
17. END
```

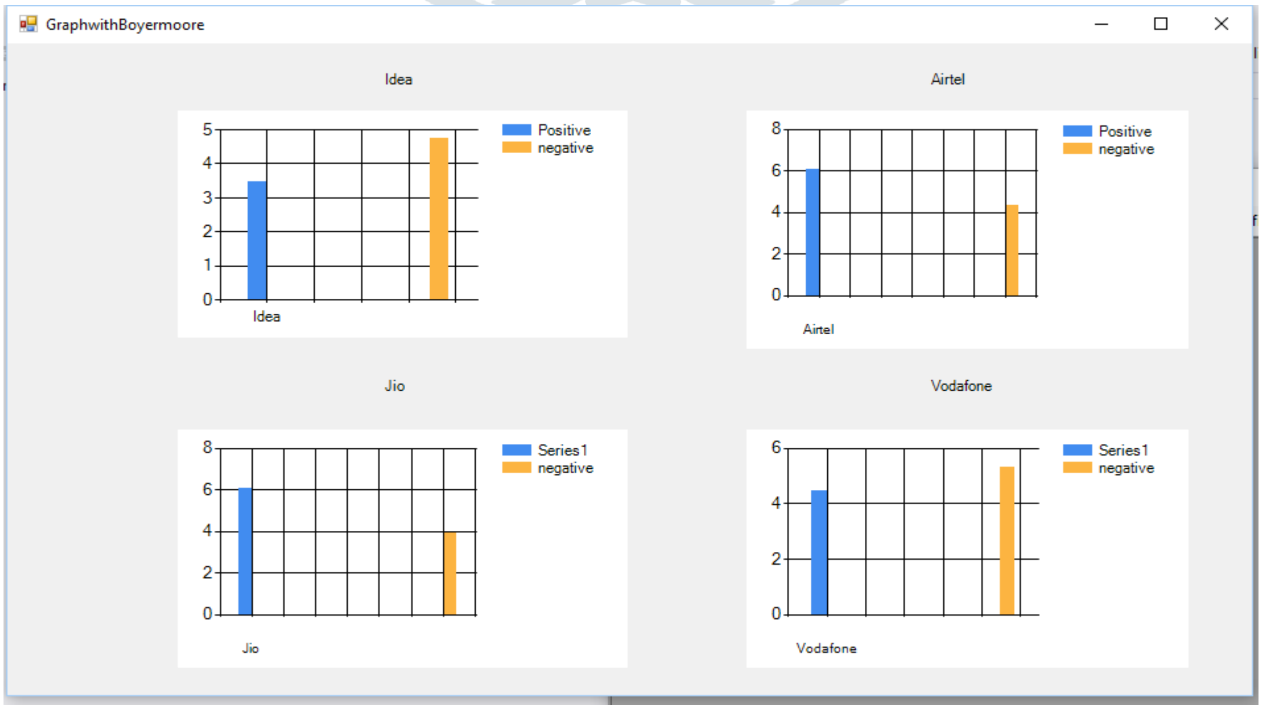


Fig 5: Comparison of positive and negative keys usingBoyer Moore algorithm

The algorithm for Boyer Moore String pattern matching is mentioned below:

```
1.  Void boyermoore(char a[] , char b[])
2.  int  x ,m ,k, l, p_ch ;
3.  x=strlen(a);
4.  y=strlen(b);
5.  k=y-1 , l=y-1;
6.  while(k<x)
7.  if(b[l] = =a[k])
8.  if (l= =0)
9.  return k;
10. else
11. l- -;
12. k - -;
13. else
14. p_ch = find(b, a[k])
15. if(p_ch= = -1)
16. k=k+y;
17. else
18. k=k+j;
19. l=y-1;
20. return -1;
21. int  y , k , l,
22. m=length(b)
23. for(k=y-2; k>=0;k--)
24. if(ch= =b[k])
25. return k;
26. return -1;
27. end
```

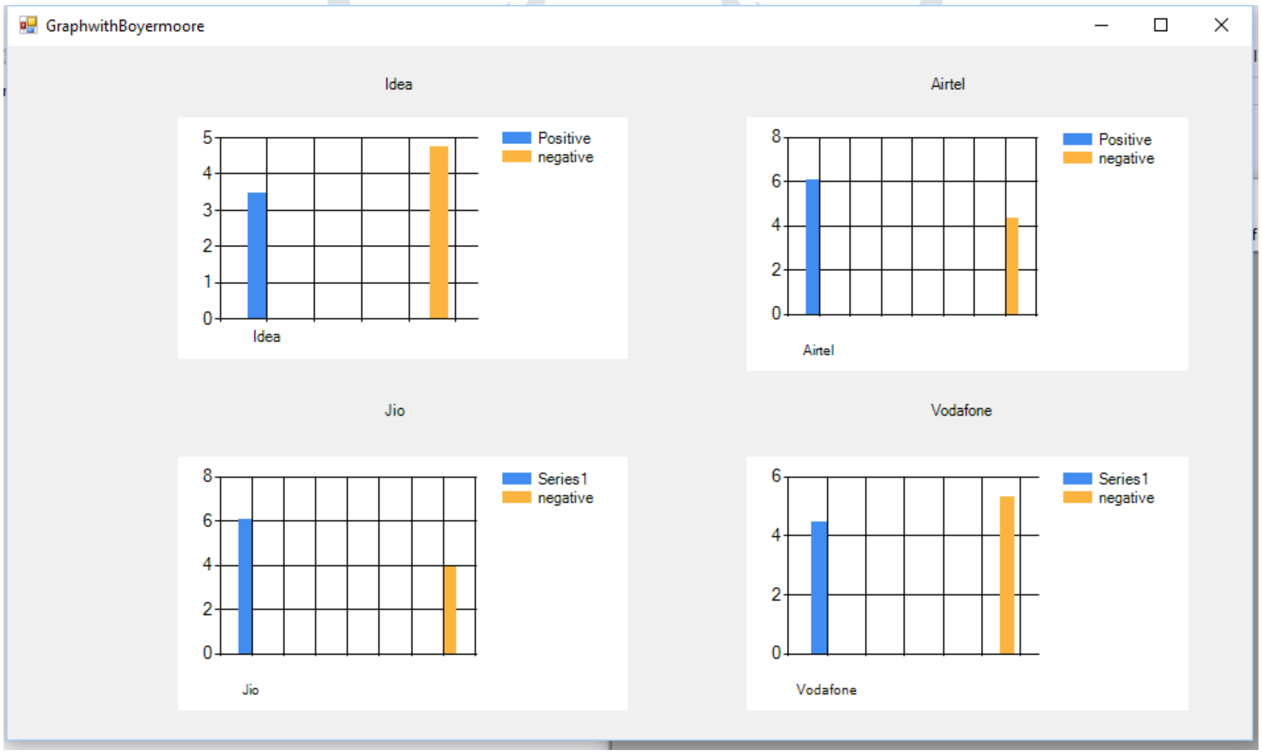


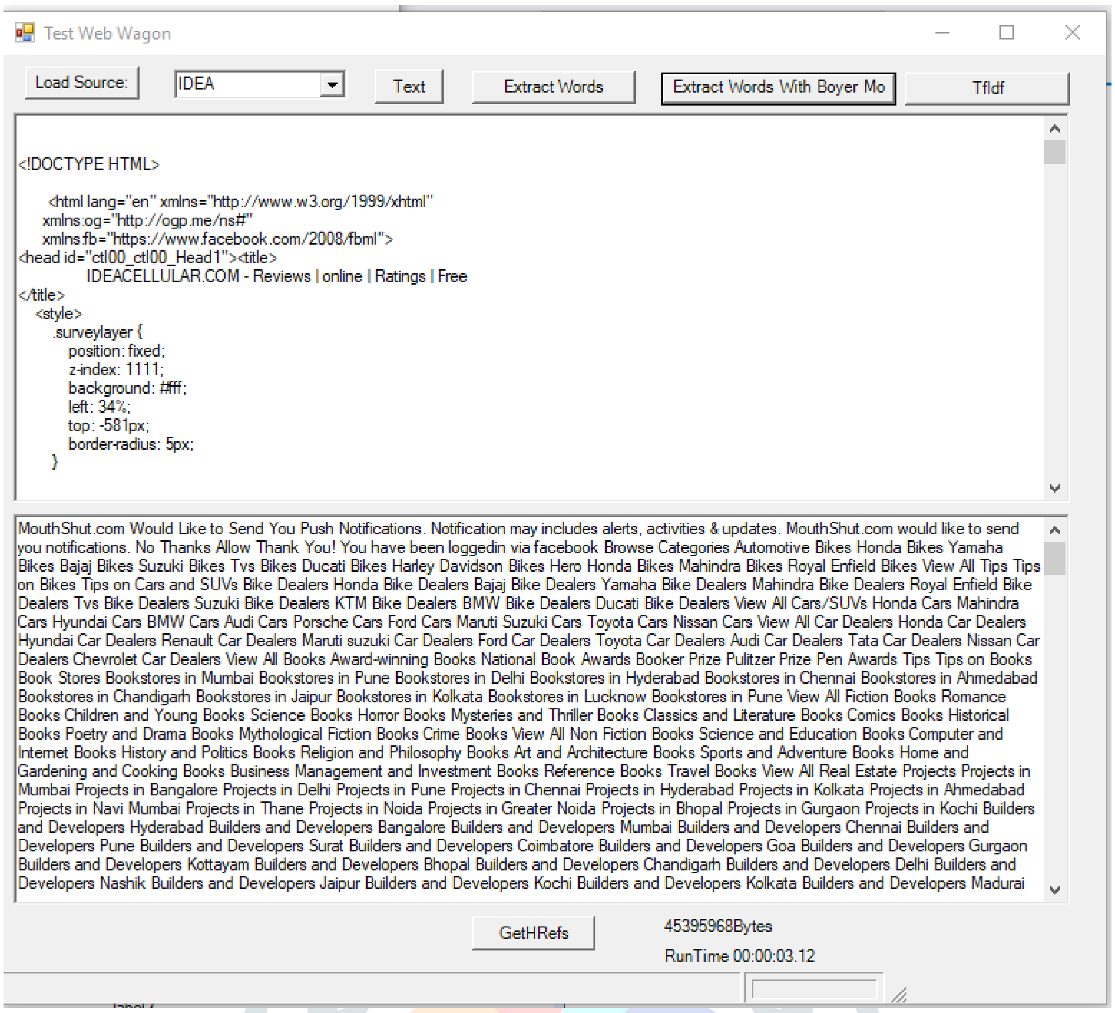
Fig 6: Comparison of positive and negative keysusing composite Boyer Moore algorithm.

The algorithm that we have used for composite Boyer Moore Algorithm is stated below:

```
1.  void CBMA(m , a , b)
2.  while b<m && t[i]=p[b]
3.  do
4.  a=a+1;
5.  b=b+1;
6.  fp(pos)
7.  pos++
8.  func_CBMA(&s , &v)
9.  global s;
10. a=0;
11. b=s[0]= -1;
12. while(a<v)
13. while(b> -1) && (s[a]!= s[b])
```

```
b= s[b];
if(s[a]= =s[b])
s[a]=s[b];
else
s[a]=b;
a++;
function MD(&s , &v)
global s;
for(a=0;a<100;a++)
s[a]=m+1;
for(a=0;a<v;a++)
s(f(p[a])=v-a;
int comp(&s, &v, &x, &n)
global s;
global count;
a=0;
b=0;
vs=v-1;
as=vs;
if(v<1)
return;
compile_s(s,v)
compile_count(s,v)
while(s[vs]!=x[as])
a=a.set(x[as+1])?(x[as]:0;
as+=count[s(a)];
if(as>=n)
return;
b=0;
a=as-vs;
while(b<vs) &&(x[a]= = s[b])
a++;
b++;
if(b= =vs)
a-vs=s;
a++;
b++;
if(b<=0)
a++;
Else
b=s[b-1];
else
while((b<v)&&(x[a]= =s[b]))
a++;
b++;
If(b= =v)
a-v=s;
b=s[b];
a[s]=a+vs-b;
return count;
END
```

When we compared the entire three algorithms Composite Boyer Moore proved to be efficient and productive. We have also compared the memory consumption and time consumption of the above algorithms .The results obtained are as follows:



```
<!DOCTYPE HTML>

     <html lang="en" xmlns="http://www.w3.org/1999/xhtml"
    xmlns:og="http://ogp.me/ns#"
    xmlns:fb="https://www.facebook.com/2008/fbml">
<head id="ctl00_ctl00_Head1"><title>
            IDEACELLULAR.COM - Reviews | online | Ratings | Free
</title>
    <style>
        .surveylayer {
            position: fixed;
            z-index: 1111;
            background: #fff;
            left: 34%;
            top: -581px;
            border-radius: 5px;
        }
```

```
MouthShut.com Would Like to Send You Push Notifications. Notification may includes alerts, activities & updates. MouthShut.com would like to send
you notifications. No Thanks Allow Thank You! You have been loggedin via facebook Browse Categories Automotive Bikes Honda Bikes Yamaha
Bikes Bajaj Bikes Suzuki Bikes Tvs Bikes Ducati Bikes Harley Davidson Bikes Hero Honda Bikes Mahindra Bikes Royal Enfield Bikes View All Tips Tips
on Bikes Tips on Cars and SUVs Bike Dealers Honda Bike Dealers Bajaj Bike Dealers Yamaha Bike Dealers Mahindra Bike Dealers Royal Enfield Bike
Dealers Tvs Bike Dealers Suzuki Bike Dealers KTM Bike Dealers BMW Bike Dealers Ducati Bike Dealers View All Cars/SUVs Honda Cars Mahindra
Cars Hyundai Cars BMW Cars Audi Cars Porsche Cars Ford Cars Maruti Suzuki Cars Toyota Cars Nissan Cars View All Car Dealers Honda Car Dealers
Hyundai Car Dealers Renault Car Dealers Maruti suzuki Car Dealers Ford Car Dealers Toyota Car Dealers Audi Car Dealers Tata Car Dealers Nissan Car
Dealers Chevrolet Car Dealers View All Books Award-winning Books National Book Awards Booker Prize Pulitzer Prize Pen Awards Tips Tips on Books
Book Stores Bookstores in Mumbai Bookstores in Pune Bookstores in Delhi Bookstores in Hyderabad Bookstores in Chennai Bookstores in Ahmedabad
Bookstores in Chandigarh Bookstores in Jaipur Bookstores in Kolkata Bookstores in Lucknow Bookstores in Pune View All Fiction Books Romance
Books Children and Young Books Science Books Horror Books Mysteries and Thriller Books Classics and Literature Books Comics Books Historical
Books Poetry and Drama Books Mythological Fiction Books Crime Books View All Non Fiction Books Science and Education Books Computer and
Internet Books History and Politics Books Religion and Philosophy Books Art and Architecture Books Sports and Adventure Books Home and
Gardening and Cooking Books Business Management and Investment Books Reference Books Travel Books View All Real Estate Projects Projects in
Mumbai Projects in Bangalore Projects in Delhi Projects in Pune Projects in Chennai Projects in Hyderabad Projects in Kolkata Projects in Ahmedabad
Projects in Navi Mumbai Projects in Thane Projects in Noida Projects in Greater Noida Projects in Bhopal Projects in Gurgaon Projects in Kochi Builders
and Developers Hyderabad Builders and Developers Bangalore Builders and Developers Mumbai Builders and Developers Chennai Builders and
Developers Pune Builders and Developers Surat Builders and Developers Coimbatore Builders and Developers Goa Builders and Developers Gurgaon
Builders and Developers Kottayam Builders and Developers Bhopal Builders and Developers Chandigarh Builders and Developers Delhi Builders and
Developers Nashik Builders and Developers Jaipur Builders and Developers Kochi Builders and Developers Kolkata Builders and Developers Madurai
```

```
45395968Bytes
RunTime 00:00:03.12
```

Fig 7: Space and Time Consumed By Enhanced Boyer Moore Algorithm

After analyzing all the above results we can conclude that Composite Boyer Moore string pattern matching algorithm is most efficient and productive in terms of time consumption and memory consumption.

**References**

**[1]**B.Umamageswari, Dr. R. Kalpana, V.Archana, "Web Data Extraction Using Boyer Moore Algorithm ", ISSN(PRINT):2398374, (ONLINE): 2394-0697,VOLUME-5,ISSUE-4,2018.

**[2]** Jamuna Bhandari , Anil Kumar "String Matching Rules By Variants Of Boyer Moore Algorithm" journal of global research in Computer Science Volume 5, No.1,January 2017.

**[3]** Shivendra kumar Pandey, Neeraj Kumar Dubey, Sonam Sharma "A Study On String Matching Methodologies" , International Journal of Computer Science and Information Technologies,Vol.5(3), 2016, 4732-4735.

**[4]** http://somemoreacademic.blogspot.com/2012/09/brute-forcenaive-string-matching.html

**[5]** Ranti Eka Putri , Andysah Putera, Utama Siahaan "Examination of Document Similarity Using Rabin Karp Algorithm" , International journal of Recent Trends in Engineering & Research ISSN (ONLINE) : 2455-1457.

**[6]** Sriharsha Oddiraju "Boyer moore" Indiana State University TerreHaute IN ,USA ,International Journal of Computer Science.

**[7]** Dr. S.Vijiyarani ,MS. E.Suganya "Research issues in Web Mining" , International Journal Of Computer Aided Technologies(IJCAx) ,vol2, no3, july 2015.

**[8]** Paul J.M.Havinga , Gerard J.M.Smit "Octopus – an energy – efficient architecture for wireless multimedia systems" , International Journal Netherlands.

**[9]** Er. Mohammad Shabaz , Er. Neha Kumari "Advanced Rabin Karp Algorithm For String Matching" ,International Journal of Current Research Vol.9, Issue , 09,pp.57572-57574,September ,2017.

**[10]** Ranti Eka Putri , Andysah Putera, Utama Siahaan "Examination of Document Similarity Using Rabin Karp Algorithm" , International journal of Recent Trends in Engineering & Research ISSN (ONLINE) : 2455-1457.