IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Volume-4, Issue-I E-ISSN: 2347-2693

# Contributing Efforts of Various String Matching Methodologies in Real World Applications

Kapil Kumar Soni

Department of Computer Science and Engineering
*Truba Institute of Engineering and Information Technology, Bhopal, India*

**www.ijcseonline.org**

Received: Dec/13/2015 Revised: Dec/26/2015 Accepted: Jan/11/2016 Published: Jan/30/ 2016

***Abstract* —** String matching is a conventional problem in computer science. For a known text string 'T', the problem of string matching is to locate whether a pattern string 'P' occurs in 'T' or not, and if 'P' occurs then the position of 'P' in 'T' is reported. String matching sometimes called string searching has become an important aspect of the real world because it is being used in many applications where the string algorithm tries to find a location of one or several strings (also called Patterns) within a larger string or text (Text Data Set). A few of its essential applications are Spell Checkers, Spam Filters, Intrusion Detection System, Search Engines, Plagiarism Detection, Bioinformatics, Digital Forensics and Information Retrieval Systems, etc. The paper includes various string matching methodologies along with its historical contributory details in a variety of needful real world applications.

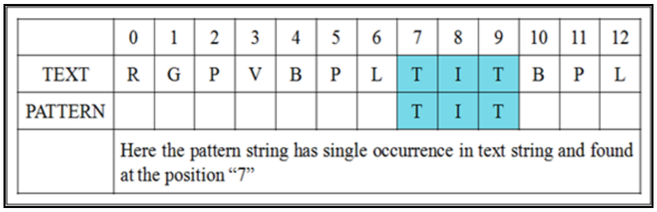
***Keywords***-String Matching, Spell checkers, Spam Filter, Intrusion Detection System, Search Engines, Plagiarism Detection, Bioinformatics

## I. INTRODUCTION AND LITERATURE REVIEW

The problem of string matching exists if there are two given strings, one is text T [0…. n-1] i.e. text dataset and the other is pattern P [0…. m-1] i.e. the search string which is to be matched within the main string where m<=n. Let us take a pattern string "P" of length "m"and text string 'T' of length "n", then start searching and if 'P' exists in that case the position of it should be reported in 'T' [1]. The string matching problem is best described by using the Example 1, Example 2 and Example 3.

**Example 1:** A real life Application of String Matching, Assume we are in the context of Bioinformatics. We want to know whether pattern **P = {C GGT AGC}** occurs in text **T** given below. Here the alphabet is composed of the four main bases found in DNA and RNA, **{C, G, T, A}**. [2] **T = {A G T C C T G A A G G T T A A *C G G T A G* C A G T C C T G A A G G T T A A *C G G T A G C* A A A T T T G G G C C C C G T A}**

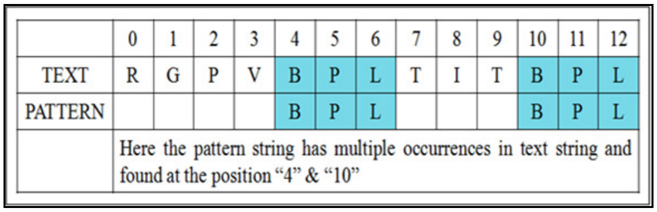
**Example 2:** Pattern String (**TIT**) is being searched in the Text String (**RGPVBPLTITBPL**) shown in Figure 1.



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TEXT | R | G | P | V | B | P | L | T | I | T | B | P | L |
| PATTERN | | | | | | | | T | I | T | | | |
| | Here the pattern string has single occurrence in text string and found at the position "7" | | | | | | | | | | | | |

Figure 1: Single Occurrence String Matching Solution

**Example 3:** Pattern String (**BPL**) is being searched in the Text String (**RGPVBPLTITBPL**) shown in Figure 2.



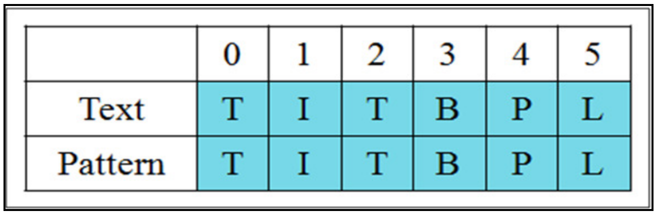
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TEXT | R | G | P | V | B | P | L | T | I | T | B | P | L |
| PATTERN | | | | | B | P | L | | | | B | P | L |
| | Here the pattern string has multiple occurrences in text string and found at the position "4" & "10" | | | | | | | | | | | | |

Figure 2: Multiple Occurrence String Matching Solution

String matching applications can be classified into two types - Exact and Approximate String Matching and both of which have its practical applications in real world scenarios their illustration are given in Example 4, Example 5 and Example 6. [3]

**Example 4:** Pattern String (**TITBPL**) is being searched in the Text String (**TITBPL**) shown in Figure 3.

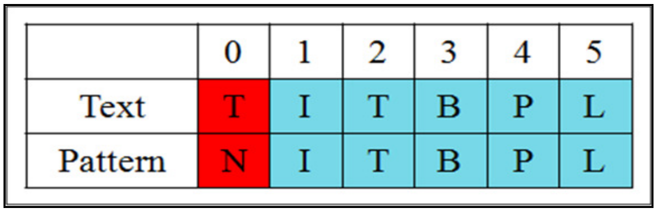
**Example 5:** Pattern String (**TITBPL**) is being searched in the Text String (**NITBPL**) shown in Figure 4.



| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Text | T | I | T | B | P | L |
| Pattern | T | I | T | B | P | L |

Figure 3: Exact Occurrence of Pattern Found

**Exact String Matching:** All exact occurrences of pattern in a text string are reported [4], Applications of Exact String Matching are: Intrusion Detection, Search Engines, Spam Filters and etc.

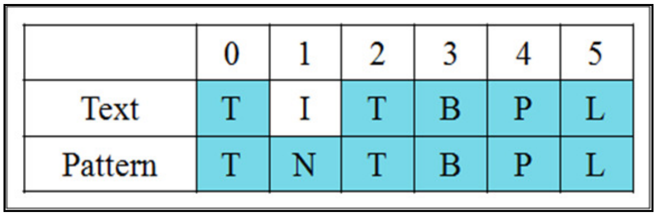


| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Text | T | I | T | B | P | L |
| Pattern | N | I | T | B | P | L |

Figure 4: Exact Occurrence of Pattern Could Not Find

**Approximate String Matching:** All approximate occurrences of pattern in a text string are reported inclusive of false matches [5], Applications of Approximate String Matching are: Spell Checkers, Plagiarism Detection, DNA Sequencing and etc.

**Example 5:** Pattern String (**TITBPL**) is being searched in the Text String (**TNTBPL**) shown in Figure 5.



| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Text | T | I | T | B | P | L |
| Pattern | T | N | T | B | P | L |

Figure 5: Approximate Occurrence of Pattern Found

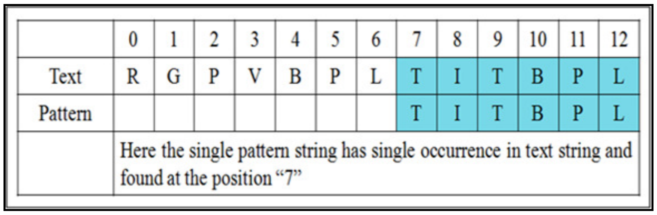
Classification can also be defined on the basis of:-

**Number of Patterns to be searched:** Single Pattern String Matching and Multiple Pattern String Matching, graphic can be seen by means of Example 6 and Example 7.

**Example 6:** Single Pattern String (**TITBPL**) is being searched in the Text String (**RGPVBPLTITBPL**) shown in Figure 6.

**Example 7:** Multiple Pattern Strings (**BPL & TIT**) are being searched in the Text String (**RGPVBPLTITBPL**) shown in Figure 7.

**Order of Searching**: Searching from left to right, searching from right to left and etc., illustration can be understood through Example 8 and Example 9.

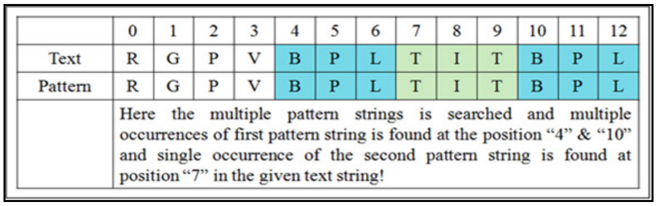


| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Text | R | G | P | V | B | P | L | T | I | T | B | P | L |
| Pattern | | | | | | | | T | I | T | B | P | L |
| | Here the single pattern string has single occurrence in text string and found at the position "7" | | | | | | | | | | | | |

Figure 6: Classification on Single Pattern Searching

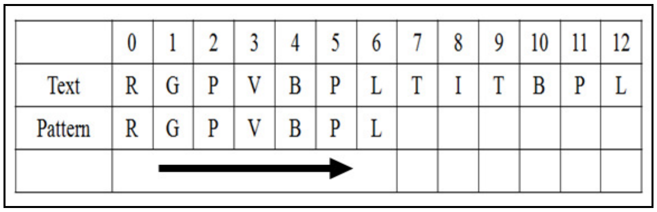
**Example 6:** Pattern String (**RGPVBPL**) is being searched in the Text String (**RGPVBPLTITBPL**) while order of searching is left to right, shown in Figure 8.

**Example 7:** Pattern Strings (**RGPVBPL**) is being searched in the Text String (**RGPVBPLTITBPL**) while order of searching is right to left, shown in Figure 9.



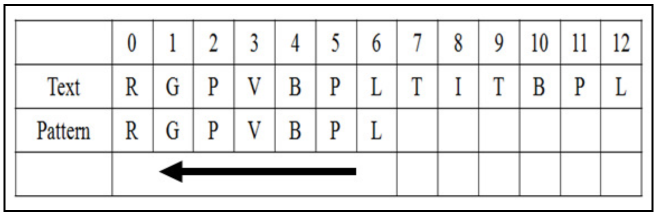
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Text | R | G | P | V | B | P | L | T | I | T | B | P | L |
| Pattern | R | G | P | V | B | P | L | T | I | T | B | P | L |
| | Here the multiple pattern strings is searched and multiple occurrences of first pattern string is found at the position "4" & "10" and single occurrence of the second pattern string is found at position "7" in the given text string! | | | | | | | | | | | | |

Figure 7: Classification on Multiple Pattern Searching



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Text | R | G | P | V | B | P | L | T | I | T | B | P | L |
| Pattern | R | G | P | V | B | P | L | | | | | | |
| | ⟶ | | | | | | | | | | | | |

Figure 8: Order of Pattern Searching Left to Right



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Text | R | G | P | V | B | P | L | T | I | T | B | P | L |
| Pattern | R | G | P | V | B | P | L | | | | | | |
| | ⟵ | | | | | | | | | | | | |

Figure 9: Order of Pattern Searching Right to Left

There are lots of applications in which string matching plays essential job. Applications like Spell Checkers, Spam Filters, Intrusion Detection System, Search Engines, Plagiarism Detection, Bioinformatics / DNA Sequencing and etc.

As already said this paper includes the brief discussion of tasks perform by string matching in all different above named applications along with the methodology, so the string matching strategies or algorithms are defined in next section.

## II. PAST ASPACTS OF STRING MATCHING

The fundamental string matching approach is Brute Force Algorithm which considers all possible cases and taking shifts only one place to right even match or mismatch condition occurs anywhere. This algorithm also known as Naives approach. [1, 6, 7]

In 1956 Kleene proved the equivalence between finite automaton and regular expression which could be use to solve the string matching problem. [7]

Avoiding numerous comparisons in brute force algorithm, in 1970 Morris and Pratt algorithm was proposed which has linear behavior. This algorithm is based on pre-processing of pattern and compares character from left to right and if mismatch occurs, it skips some character based on pre-processing phase. [8]

In 1977 Knuth Morris Pratt introduced an algorithm having a choice of improvements in Morris and Pratt algorithm.

KMP has same time complexity as Morris and Pratt algorithm but searching performance found to be much better than Morris and Pratt algorithm. [9]

In 1977 Boyer and Moore also proposed algorithm which compares character from right to left. [10]

There are so many multiple pattern string matching algorithms has already been proposed in past decades such as: In 1975 Aho-Corasick algorithm [11] was presented by Alfred V. Aho and Margaret J. Corasick, which constructs automata for patterns in pre-processing phase. Commentz Walter [7] proposed an algorithm which was based on Aho-Corasick and Boyer-Moore algorithm, Rabin Karp algorithm [12] is also used to search multiple patterns. Variety of algorithms based on different methodologies has already been suggested in the past decades, historical listing of various important string matching algorithms is being described in the Figure 10 which is showing the Historical View of String Matching. [10]

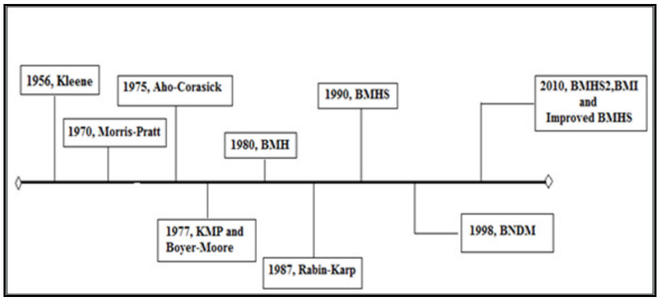


Figure 10: Historical View of String Matching [10]

## III. Applicability of String Matching

In awareness to the real world problems string matching is having several applications, only some of which are being described here.

### A. *Spell Checkers:*

A computer program which checks the spelling of words in files of text, typically by comparison with a stored list of words. In spell checkers pre-defined set of patterns stored in database dictionary along with document file is fed as input, now the string matching module of spell checker will check for such patterns if there occurs then it shows the occurrence by reaching to its final states otherwise reports miss spelled word list in output. [3, 7, 13] Spell Checkers basic module is shown in Figure 11.

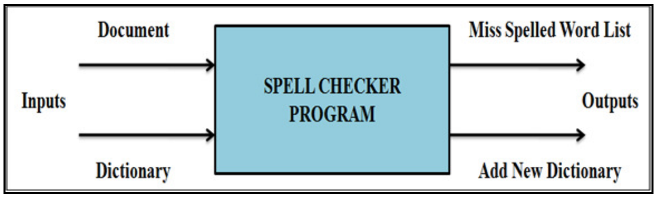


Figure 11: String Matching Methodology of Spell Checker

### B. *Spam Filters / Spam Detection System:*

Unsolicited and unwanted emails are called spam that engages lots of network bandwidth. A spam filter is a program that is used to detect unsolicited and unwanted email and prevent those messages from getting to a user's inbox. All spam filters use the concept of string matching to identify and discard the spam. Spam filter searches suspected signature patterns in the content of email by applying string matching. All content based filters are worked on string matching. [3, 7, 13] Spam filter basic structure is shown in Figure 12.

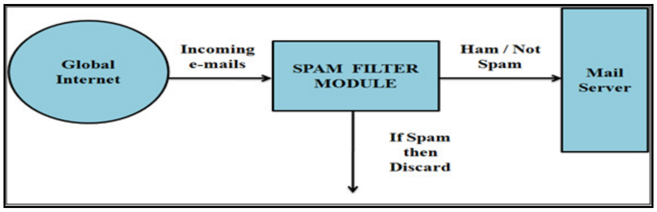


Figure 12: String Matching Methodology of Spam Filters

### C. *Intrusion Detection System:*

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities, Intrusion prevention systems (IPS), also known as intrusion detection and prevention systems (IDPS), are network security appliances that monitor network and / or system activities for malicious activity. In Intrusion Detection System [4] incoming data packets that contain intrusion related keywords are found by applying string matching strategy. All the malicious code is stored in the database and every incoming data is compared with stored data. If match found then alarm is generated. It is based on exact string matching algorithms where we have to capture each and every intruded packet and they must be detected. The Intrusion detection system modal is shown in Figure 13.

### D. *Search Engines:*

Search engines are program that search the documents for specified keywords on the World Wide Web and returns a list of the documents where the keywords were found. . Most of the data are available on internet in the form of textual data. Due to the large quantity of uncategorized text data, it becomes really difficult to search a particular content. Web search engines help us to solve this problem by organizing the required text / data as efficiently as possible. To categorize these data string matching algorithms are used. Categorization is done on the basis of search keywords. [6, 13] Figure 14 shows the basic model of Search Engine.

### E. *DNA Sequencing / Bioinformatics:*

DNA sequencing is the process of determining the precise order of nucleotides within a DNA molecule. It includes any method or technology that is used to determine the order of the four bases—adenine, guanine, cytosine, and thymine in a strand of DNA. [5]

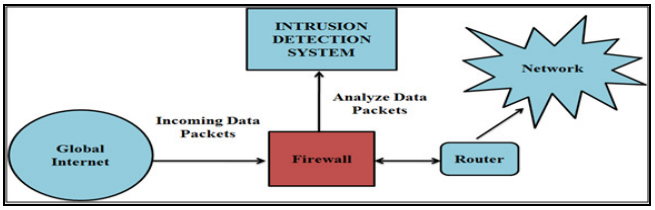


Figure 13: String Matching Methodology of Intrusion Detection

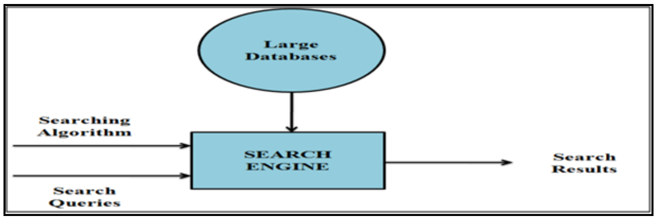


Figure 14: String Matching Methodology of Search Engine

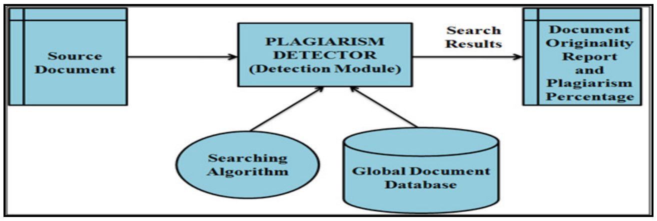

Source Document
PLAGIARISM DETECTOR (Detection Module)
Search Results
Document Originality Report and Plagiarism Percentage
Searching Algorithm
Global Document Database

Figure 15: String Matching Methodology of Plagiarism Detector

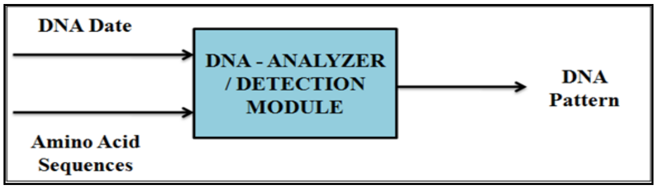


Figure 16: String Matching Methodology of DNA Sequencer

Bioinformatics is the application of information technology and computer science to biological problems, in perspective to the issues involving genetic sequences and in order to find the DNA patterns, string matching module and DNA analyzer both works with collaboration for finding the occurrence of the pattern set. [7, 13] Above figure 16, showing the Bioinformatics DNA Sequencing Module.

## IV. CONCLUSION

Due to influence of string matching in realm of computer science, it contributes essential effort for practical applications. Approximate string matching demand is more than Exact String Matching. Since last decades several String Matching Algorithms were introduced, each one may have improvement and enhancement possibility. Exact string matching has practical applications like Intrusion Detection, Search Engine and Spam Filters etc.; whereas approximate string matching has applications like Spell Checker, Plagiarism Detection and DNA Sequencing etc. To improve searching lots of algorithm has already been suggested, and observed that Multiple Pattern Algorithms have more practical applications comparing with Single Pattern Algorithms. An efficient and innovative searching algorithm becomes the key research areas in perspective of current real world scenario.

## REFERENCES

[1] Thomas H Corman, Charles E. Leiserson, Oonald L. Rivest and Clifforf Stein, “Introduction to Algorithms – String Matching”, IEEE Edition, 2nd Edition, PP No. 906-907.

[2] Leena Salmela, J. Tarhio and J. Kytojoki “*MultiPattern String Matching with Very Large Pattern Sets*”, ACM Journal Algorithmic, Volume 11, 2006.

[3] Nimisha Singla, Deepak Garg, “String Matching Algorithms and their Applicability in Various Applications” IJSCE, ISSN 2231-2307 Vol I, PP No. 6, January 2012.

[4] Simone Faro and Thierry Lecroq, “The exact online string matching problem: A review of the most recent results” ACM computing surveys Vol .V, PP .N, Article A, January 2011.

[5] Gonzalo Navarro, “A Guided Tour to Approximate String”, ACM Computing Surveys, Vol 33 No. 1, PP No. 31-88, March 2001.

[6] Christian Charras and Thierry Lecroq, “Handbook of Exact String Matching Algorithms”, Published in King’s college publication, Feb 2004.

[7] Alberto Apostolico and ZviGalil,” Pattern Matching Algorithms” Published in Oxford University Press, USA, 1st edition, May 29, 1997.

[8] Morris J.H., Pratt V.R., 1970, “A Linear Pattern-Matching Algorithm”, Technical Report40, University of California, Berkeley 1970.

[9] Donald Knuth; James H. Morris, Jr, Vaughaz Pratt (1977). "Fast Pattern Matching in Strings". SIAM Journal on Computing 6 (2): 323–350. Doi: 10.1137/0206024.

[10] BOYER, R. S. AND MOORE, J. S,”A fast string searching algorithm”, Communication of ACM 20, Vol. 10, pp. 762–772, 1977.

[11] Alfred v. Aho and Margaret J. Corasick,”Efficient String Matching: An aid to Bibliographic Search” communication of ACM, vol. 18, june 1975.

[12] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001-09-01). "The Rabin–Karp algorithm". Introduction to Algorithms (2nd ed.). Cambridge, Massachusetts: MIT Press. pp. 911–916.

[13] V. Saikrishna, A. Rasool, N. Khare, “String Matching and its Applications in Diversified Fields”, IJCSI Jan 2012, Volume 9- PP No 1.

### *AUTHORS PROFILE*

**Prof. Kapil Kumar Soni**
Assistant Professor,
Department of Computer Science & Engineering,
Truba Institute of Engineering and Information Technology, Bhopal.