International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 4, Issue 12, December 2015 **ISSN 2319 - 4847**

A Survey and Analysis on String Matching Techniques and its Applications

Ritesh Kothari[1], Nishchol Mishra[2], Sanjeev Sharma[3] ,Ravindra Patel[4]

[1]M.Tech. Scholar, School of Information Technology, RGPV, Bhopal, M.P.

[2]Assistant Professor, School of Information Technology, RGPV, Bhopal, M.P.

[3]Associate Professor, School of Information Technology, RGPV, Bhopal, M.P.

[4]Associate Professor, Dept. of MCA, RGPV, Bhopal, M.P.

ABSTRACT

String matching is very popular as well as useful in many web related applications. String matching is related to information retrieval and helpful for performing search in search engines, social networking and other online applications as well as in software applications. This paper, deals with survey and analysis of state of art, exact string matching algorithms and approximate string matching algorithms with their applications including spell checking, plagiarism detection, intrusion detection and so on. An important objective of this survey is to describe that different string matching algorithms produce different results according to need and some of them are comparable as well as optimum in terms of efficiency. This paper also states that how different techniques, functions and approaches like preprocessing, character comparison, hash function, edit distance, n-gram and other, are used in string matching algorithms for making them efficient and optimum. Further, this study describes that why different string-matching algorithms that have been developed in last few years and how these are useful to achieve optimum results.

Keywords: *String matching, social networking, spell checking, pattern string, n-gram and edit distance.*

1. Introduction

String matching is an important technique which is widely used in information retrieval, text processing and in many other online, and software applications. String matching simply states that a pattern string should be matched to the text string. String matching is used to determine the position of a pattern string in a text string and to find all or nearest strings correspond to pattern string. String matching is a technique, which is applied in real life applications as well. As the necessities arise, the need for technology is come around. It is applied in string matching also. As we know that many people are now aware of world of computers, internet and information technology, and because a large amount of data are now available online, so everywhere in the field of online web related applications, when a person tries to find something from websites like social networking, search engines, ecommerce websites, and other softwares like dictionaries etc., string matching is very helpful. So there are two types of string matching techniques are used, one is exact string matching and another is approximate string matching. If a pattern string is exactly matched to the text string then it is called exact string matching and if a query string is not exactly matched to the text string but close to the text string, then it will be called approximate string matching. Here an approximate string matching means string matching allowing errors. [3] Approximate string matching is very beneficial to perform in online data searching. String matching is defined in other words also, let we have a pattern string P [1...m] and a text string T[1....n] where T>=P, so here string matching algorithm will perform an action to find a query or source string P in the given text string T, which is a target string.

In the different application areas like intrusion detection system, bioinformatics applications, plagiarism detection, pattern identifying, document searching , text mining, video retrieval, spell checking, data cleaning and so on, both the string matching algorithms are applied. Different functions, concepts and approaches are applied to both types of the algorithms to achieve optimum and efficient results.

String matching is important for this generation because most of the information is available online and millions of people are using internet so searching any word in the search engines is very convenient now with the help of the string matching techniques.

We can say that social networking field has become now a very useful for this modern technology. In the field of the social networking, string matching is playing an important role. Approximate string matching techniques are used for making a string available to the user with wrong spelling typed word because remembering the correct spelling of a particular word is very difficult for human mind. If we talk about the social networking world, concept of string matching is very much useful because a lot of people have an account on social networking sites like Facebook, Twitter etc. and if they perform a searching for a person's name for finding his/her friends on these websites then this searching

should be fast as well as throughput should be high. According to the social media, 2014 there were total 1,280,000,000 monthly active users on the Facebook. [1][2] Like as literature in the field of computer science a large amount of data (called big data) are being generated and stored. That's why string-matching algorithms should be efficient, while the speed and storage capacity of computers are increasing rapidly. [4]
So many algorithms have been developing for some years according to the requirements because string searching should be fast as well as in quick time. Most of the string matching algorithm is hybrid algorithms. Hybrid algorithms are created by merging two or more than two parent algorithms.

2. Literature Work

This section covers a brief description of exact string matching algorithms and approximate string matching algorithms with their functions, approaches, concepts, different behavior and application areas. There is a competition among several algorithms for getting optimum results.

2.1 Exact String Matching

Exact string matching is a technique in which a pattern string is found in text string or database (containing strings) in an exact manner. String matching comprises in retrieving one or more (generally all) patterns of a string in a text. [4]. Let we have a text string T and a pattern string P, and we have to find all the occurrence of P in T.

T = socialnetworkingsitesnetworking
P = work
After appling algorithms we find the results as:

P€T= socialnet***work***ingsites***work***ing
So pattern string ***work*** is found twice in a text given.

Exact string matching algorithms deal with character comparisons approach, searching time complexity, preprocessing phase, hash function etc.
Here we are performing a study on exact string matching algorithms with the help of 'Handbook of Exact String Matching Algorithms'. We analyzed that many exact string-matching algorithms have been developed with their working functionalities and some of them are summarized here with their important characteristics.

2.1.1 Brute Force Algorithm [4]

Working functionalities and features - Brute force algorithm is one of the important string matching algorithms, which is useful in some hybrid string matching algorithms also.

1. It does not perform any preprocessing phase.
2. Pattern is found in this way, that if characters are matched then no shifting is done else always shift the window by exactly one position to the right.
3. This algorithm does character comparison in any order for matching a pattern and expected no of character comparison is 2n.
4. Brute force algorithm requires O (mxn) searching time complexity.

2.1.2 Karp- Rabin Algorithm [4]

Working functionalities and features- Karp-Rabin algorithm is also useful string matching algorithm came with new strategies like hash function.

1. This algorithm adds a new technique in itself, called hash function. Hash function helps in the better performance of string searching because.
2. This algorithm implements a preprocessing phase in a constant space, with a time complexity of O (m).
3. This algorithm compares character of pattern to text, in any order and expected no of character comparison is O (m+n).
4. This algorithm performs the search in O (mxn) searching time complexity.

2.1.3 Morris-Pratt Algorithm-[4]

Working functionalities and features- Morris Pratt algorithm is also a useful string-matching algorithm which follows the functionalities of Brute Force algorithm, but the difference is that here no of comparisons are less and performance is more.

1. Morris-Pratt algorithm performs a preprocessing phase in a O (m) space complexity, with a time complexity of O(m).
2. Here the character comparisons are performed from left towards right direction.
3. During the comparison expected no of text Character comparison is 2n-1.
4. This algorithm performs the search in O (m+n) searching time complexity.

2.1.4 Knuth-Morris-Pratt Algorithm - [4]

Working functionalities and features – As this algorithm's name sounds, it follows the concepts of Moris-Pratt algorithm, but no of character comparisons are saved, because shifts length is improved here and due to this searching speed is increased.

1. Here preprocessing phase is performed in O (m) space and time complexity.
2. This algorithm compares character of pattern to text in left to right order.
3. During the matching of pattern to text expected no of character comparison is 2n-1.
4. This algorithm performs the search in O (m+n) searching time complexity.

2.1.5 Boyer-Moore Algorithm-[4]

Working functionalities and features – Boyer-Moore string matching algorithm is very convenient and helpful for other string matching algorithms, which are described ahead as well as different applications programs like editors etc. We can say that BM algorithm is base for many other string-matching algorithms. This algorithm introduced two functions, which are good suffix shift and bad character shift.

1. Boyer-Moore algorithm takes O (m+σ) time and space complexity to perform a preprocessing phase.
2. Boyer Moore string matching algorithm performs the shifting from right to left order.
3. During the matching of pattern's character to text character expected average no of comparison is 3n.
4. Searching is completed in an average O (m*n) time complexity and the best performance is O (n/m).

2.1.6 Horspool Algorithm-[4]

Working functionalities and features -

Horspool algorithm follows some property of Boyer Moore algorithm and there have been different researched on this algorithm. Horspool algorithm is used in different applications as intrusion detection.

1. Horspool algorithm is a simple variation of Boyer Moore algorithm and its working is quite well.
2. Horspool algorithm uses only the bad character shift property for comparing characters of pattern to text. Here Horspool gave an idea to use bad character shift of right most letter of pattern window, to perform the shifts in Boyer Moore algorithm.
3. Horspool algorithm implements preprocessing phase in O (m+σ) time and O(σ) space complexity.
4. Horspool algorithm performs no of expected character comparison in between 1/ σ and 2/(σ+1).
5. The searching phase is completed in O (m*n) average time.

2.1.7 Quick Search Algorithm-[4]

Working functionalities and features - Quick search algorithm is an efficient string matching algorithm, which produces results faster than Boyer Moore string matching algorithm. Likewise, Horspool string matching algorithm, Quick Search also follows working functionalities of Boyer Moore algorithm.

1. Quick Search string matching algorithm is also an improving variation of the Boyer Moore algorithm.
2. Quick Search algorithm also uses only the bad character shift property of BM algorithm for performing a match.
3. Quick Search string matching algorithm is also easy to implement form practical point of view.
4. Quick Search string-matching algorithm does its preprocessing phase in O (m+σ) time and O (σ) space complexity.
5. Here the searching is performed in O (m*n) time complexity.

2.1.8 Zhu-Takaoka Algorithm [4]

Working functionalities and features – Zhu-Takaoka also uses the working functionality of BM algorithm for producing the efficient and optimum results for string matching in between pattern and text. Here Zhu-Takaoka exact string matching algorithm is described with its following features.

1. As previously studied string matching algorithms, Zhu-Takaoka is also a simple and easy variant of BM string matching algorithm.
2. Zhu-Takaoka algorithm uses bad character shift to performing shifts and bad character shift is computed here with two consecutive text characters. This algorithm uses good suffix table also for shifts.
3. The character comparison is performed here from right to left direction.
4. Zhu-Takaoka algorithm completes its preprocessing phase in O(m+ σ2) time and space complexity.
5. Zhu-Takaoka completes its searching in O (m*n) time complexity.

2.1.9 Raita Algorithm [4]

Working functionalities and features – Raita has developed a string-matching algorithm in which a comparison is made between pattern and text in a specific order, for saving the comparisons.

1. Raita states that, first compare the last character of pattern to the right most text character in the window then the first character of pattern to left most character of text, if matching is found then match the middle one and continue for others form second one.

2. Raite algorithm performs the shifts using in string matching as Horspool algorithm.
3. Raita string matching algorithm implements its preprocessing phase in O (m+ σ) time and O(σ) space complexity.
4. Raita string matching algorithm does a searching phase in O(mxn) time complexity.

2.1.10 Skip Search Algorithm [4]

Working functionalities and features-, Skip search algorithm is a string-matching algorithm, which uses a concept of buckets.

1. Skip search algorithm uses a new one logic, of buckets. In this Skip search algorithm uses the buckets of positions for every character of the alphabet.
2. Skip search algorithm completes its preprocessing phase in O (m+ σ) time and space complexity.
3. Skip search algorithm needs O (n) expected text character comparisons for performing string matching.
4. Skip search algorithm completes its searching phase in O (m+n) time complexity.

Table 1: Analysis of exact string matching algorithms.

NAME OF ALGORITHM	NO OF CHARACTER COMPARISION/ORDER	PRE-PROCESSING TIME COMPLEXITY	SEARCHING TIME COMPLEXITY
Brute force	any order	No preprocessing phase	O (mxn).
Karp- Rabin	O (m+n).	O(m)	O(mxn).
Morris-Pratt	left to right	O(m)	O(m+n)
Knuth-Morris-Pratt	left to right	O(m)	O(m+n)
Boyer-Moore	Right to Left	O(m+|Σ|)	O(mn)
Horspool	1/ σ to 2/(σ+1)	O(m+σ)	O(m*n)
Quick Search	Any Order	O(m+σ)	O(m*n)
Zhu-Takaoka	right to left	O(m+ σ2)	O(m*n)
Raita	specific Order	O(m+ σ)	O(mxn)
Skip Search algorithm	O(n)	O(m+ σ)	O(m+n)

From the study of above mentioned exact string matching algorithms it is observed that different string matching algorithms have different searching and space-time complexity as well as different no of character comparisons for performing a search efficient. Here it is also observed that every string-matching algorithm has a single motive of reducing the no of comparisons in between pattern and text string, to improve the performance of string search better in accordance with the speed.

2.2 Approximate String Matching

An approximate string matching is a technique in which a query string finds similar strings. In other words, an approximate string matching finds all the patterns in the text or strings database with at most k differences, that's why approximate string matching is also called a string matching with k differences. [7] The reason behind developing approximate string matching algorithm is that ,a large amount of database (with a millions of words) are available and to spell correctly for all the words is not so easy for human mind.

Approximate string matching algorithms are used in spell checking, query suggestion in search engines, similar DNA searching in large databases [8], retrieving musical passages similar to a sample. [9]

There are some important concepts, which are used in the developing of the approximate string matching algorithms in different application areas.

1. Edit distance- Edit distance is a similarity measurement process, which states that how much two strings are similar to each other. This similarity is measured by three operations 1) insertion 2) deletion 3) substitution.

2. N- gram- An n- gram is defined as substrings of a string. Here n belongs to its base. Therefore, an n-gram of a string is a contiguous n base of the string. [6]

3. Inverted list- Inverted list is a collection of n-grams for every query as well as names ids containing that n-gram. [6]

We are characterizing four types of mechanisms, which are helpful in developing the approximate string matching algorithms.

1) Algorithms based on Dynamic Programming*:* - It is an earlier mechanism. Almost theoretical results of worst case algorithms came from here and a couple of them are competitive in use. The big success in this category are O (kn) worst case algorithm and O (kn/ $\sqrt{\sigma}$) average-case algorithms. [3]
2) Algorithms Based on Automata*:* - This is also an ancient mechanism. This technique gives the best algorithms in the worst cases, which has a time complexity of O (n). [3]
3) Algorithms based on bit-parallelism: - This method is not as old as above methods. Algorithms belonging to this class work on concept of parallelism of the computers, when they perform on bits. In this category, another algorithm works in parallel using bits. This works better for small size string and good enough for any error type, occurred during searching. [3].
4) Algorithm based on filtering: - This technique came in 1990 and till now, it is in a working state. It simply filters the strings, which are not relevant to a query string. An optimal algorithm with an average time complexity of O(n(k+logm)/m) is like an achievement in this category. [3]

We did a little study of different approximate string matching algorithms. Some of them are briefly explained here-

2.2.1 Similar String Search by q-gram Matching- A general filtering algorithm [8] –

The main issue behind different experiments for approximate string matching is the frequency count of q-gram. For making a search efficient, there is a need to reduce the size of inverted lists corresponding to the query string so that cost of frequency count of q-gram could be reduced.

From the study of [8], we found an algorithm, which is created for an approximate string matching. It is known as 'Similar String Search by q-gram Matching' a general filtering algorithm. This algorithm comprises problem statements, q-gram theory, edit distance and top k similar strings corresponding to a query strings.

Problem Statement: The problem is describing about retrieving all the similar strings 'm' belong to N corresponding to a query string 'p'. These strings (m) should be at a distance ≤ e. In general it is defined as that find all the similar strings m □ *N* such that *ed p, m) ≤ e*. Where ed is edit distance between query string and similar strings 'm'.

q-Grams: This algorithm uses principal of q-gram also, because this is used in retrieval of Top-k similar strings corresponding to a query string.

Top-k Approximate String Queries: This part of algorithm is very important because after applying of q-gram and edit distance technique top-k similar strings are now filtered with the help of inverted indexes.

The essential issue in this 'general q-gram based query algorithm' is that the cost of counting the frequency of the q-grams is high, and thus efficient technique is preferred. Therefore, other different algorithms have been developed to reduce the time and increase the throughput.

2.2.2 Branch and Bound Algorithm [8]

From the above algorithm it is found that it cannot reduce the cost of frequency of q-grams in query processing so much because the size of inverted list may be large. So a new and different algorithm has been created for an efficient result in terms of reducing n-grams frequency counts so that searching could be in a quick time. This algorithm has given a name "Branch and bound algorithm". It works on branch and bound concept and it uses two techniques one is 'count filtering' and another is 'length filtering'.

Count filtering is used with length filtering. Count filtering as it sounds it is used to count the no of q-grams share by strings with a small edit distance. Length filtering is a process to fix an edit distance between strings maximum up to e. The query processing has following steps-

Initialization- Set the frequency threshold T top−k=1, and the length difference ld=0.

Branch- Extract the corresponding inverted lists in which any string S has||S|− |P|| =ld

Bound - Rank the candidate strings and obtain the temporal edit distance threshold Ttop−k

So here branch and bound algorithm is able to reduce the size of inverted list by using two new strategies count and length filtering and that's why the cost of frequency count of q-gram is also reduced.

2.2.3 Adaptive q-gram selection algorithm [8]

There is an improved approximate string-matching algorithm, which is applied in finding the top-k similar strings with respect to a query string. As its name is Adaptive q-gram selection algorithm, it does an operation in selection of q gram. Both the algorithms we studied above had a static value of q-gram. Therefore, there was a problem of creating larger size of inverted list. In this algorithm author concentrates on the q-gram selection. He proposed a q-gram dictionary, which will be chosen at each iteration. Here the study says that if there is a query having a large size comparatively then the inverted list's size may be small and then the frequency count of q-gram can be reduced. In

addition, the motive is reducing the cost of frequency counting of selected q-grams. Therefore, that search could be effective as well as quickly.

So for achieving the above results the Adaptive q-gram algorithm is developed. Here, there are three important strategies are applied in the algorithm known as count filtering, length filtering, and adaptive q-gram selection.

Branch and Bound algorithm was working on two strategies. However, in Adaptive q-gram selection while q-gram selection is performed (by creating q-gram dictionary) so the performance is increased. Although there are some limitations, also as this algorithm cannot create all possible q-gram dictionaries because of lack of resources as well as for a very large value of query string performance is not so well.

Analysis of performance- The study says that among three algorithms, it is analyzed that 'Branch and Bound' algorithm is performing very well than the first one because of using the dynamic length filtering and count filtering strategies. If we compare Branch and Bound Algorithm and Adaptive q-gram selection algorithm then, AQ is better than BB string matching algorithm, because this can reduce the cost of frequency count of q-gram by using the concept of q-gram dictionary. [8]

3. Some application areas of String Matching

3.1 Intrusion Detection

As security is an important need in a real life, likewise in the field of networking it is also required. Various softwares like firewall, antivirus, internet security are available for monitoring, detecting and removing the viruses or other malicious activities. Intrusion detection system is one of them. An Intrusion detection system is important in the field of modern networking technology to detect the malicious activities. So in the working of IDS String matching algorithms are playing an important role to perform the searching for these malicious activities. In the searching phase, approximate 70% of IDS processing time is consumed. The big problem in this present scenario is that a number of users are working on the internet so the networking traffic is being increased and because of that, there is a need to reduce the searching time. For reducing the processing time, we need to work fast. So many string-matching algorithms have been proposed and one of them is Hash - Boyer-Moore -Horspool String Matching Algorithm. In this algorithm, enhancing BMH algorithm is used by adding the hash function. Therefore, method is called Hash Boyer-Moore Horspool (HBMH). Advantage of using hash function is that it reduces the number of character comparisons perform by BMH algorithm in each comparing step. Therefore, due to the HBMH algorithm performs better in the working of IDS. [5]

3.2 Plagiarism detection

String matching is an important tool in plagiarism detection. Plagiarism is defined as replication of data. This can be occur in different fields like literature, research work, paintings, music and so on.[10] Hence plagiarism detection has emerged as an important technique in the field of computer science and information technology. Plagiarism detection is a process of detecting the replicated data in the form of words, sentences, pictures and so on. There are different plagiarism detection techniques available. One of them is string-matching techniques. There are many approaches to detect plagiarism using string matching. Grammar based plagiarism technique is one of them which is used in plagiarism detection. This approach is applied in finding the duplication in the documents by using the grammar architecture. This is applied in the documents for getting a similarity, with the help of string matching algorithms. The grammar- based technique is enough for creating a plagiarism in documents with the copy of similar documents as it is, but if there are any changes in the words or sentences then this methods does not perform so well. Therefore, we can say that there is a limitation in this method for plagiarism detection. [11]

3.3 Spell-Checking

Spell checking is very important application regarding string matching. To apply the spelling check there is a framework called spell checker. This is an important tool, which helps to people in retrieving relevant information online, and we know that remembering correct spelling of words is difficult for human minds that are why too spell checking is needful. Approximate string matching is used for spell checking and different approximate string matching algorithms are applied in this field. Here some logics are used to implement these algorithms like n-gram, edit distance, inverted list.

3.4 String matching in next generation sequencing

Bioinformatics is an important application of string matching algorithms, which is, used to solve the biological problems, like DNA. DNA means Next-generation sequencing makes short reads. The short reads pairs are generally a collection of short sequences of normally less than 200 DNA bases. There is a need to compare these short reads pairs and then a matching is performed to these short reads pairs on references sequences. Since these short reads, pairs may be varied from one person to another or we can say that DNA sequences may differ to each person depends on reference sequences. So here to find the similarity in DNA sequence on the basis of reference sequence, there is approximate string matching applied, and this is known as aligning the reads pairs across the reference sequence.[6]

3.5 String matching in dictionary

A dictionary is a collection of vocabulary. One has to find the words in the dictionary. Here we are talking about a dictionary on which a searched is performed online as well as this can be any software or desktop application is well. To getting a word in the dictionary, an approximate string matching technique is very useful. There have some approximate string matching algorithms been developed, which help to retrieve the words for which a person is performing a search even he does not know the exact spelling of that particular word. Ricardo-Baeza-Yates Gonzalo Navarro developed an algorithm for online searching in vocabulary which is a fast approximate string matching in a dictionary. This is useful in web applications where multiple users are available to retrieving the vocabulary.[12]

4. Conclusion

We conclude here that string matching is really a most important and essential factor from computer science and information technology's point of view. Because from the study it is observed that string, matching algorithms are covering most important issue in the field of networking, information retrieval and many more. String matching algorithms are beneficial to solve the problems of real life as well. As we have seen that it is used in spell checking, plagiarism detection, DNA sequences, dictionary etc. We analyzed here that several exact and approximate string-matching algorithms have different functions, properties and time complexities and so on, because all of them wants to get an optimum results. For making string search fast and efficient several string matching algorithms have been proposed and still in progress.

References

[1]. http://blog.digitalinsights.in/social-media-users-2014-stats-numbers/05205287.html.
[2]. https://en.wikipedia.org/wiki/List_of_social_networking_websites.
[3]. "A guided tour to approximate string matching", Gonjalo Navarro.
[4]. "Handbook of Exact String-Matching Algorithms", Christian Charras and Thierry Lecroq.
[5]. "Hash - Boyer-Moore - Horspool String Matching Algorithm for Intrusion Detection System", Awsan Abdulrahman Hasan and Nur Aini Abdul Rashid School of Computer Sciences, University Sciences Malaysia, Penang Malaysia
[6]. "Name spell-check framework for social networks", Burak YILDIZ, Fatih EMEKCİ.
[7]. "An Improved Algorithm for Approximate String Matching", Zvi Ga1i1 Kunsoo Park
[8]. "Fast Algorithms for Top-k Approximate String Matching", Zhenglu Yang Jianjun Yu Masaru Kitsuregawa. Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)
[9]. "Indexing Methods for Approximate String Matching", GonzaloNavarro RicardoBaeza-Yates ErkkiSutinen JormaTarhio.
[10]."A new process model for text string searching", Beebe NL, Dietrich G. Research Advances in Digital Forensics III. Norwell, Springer 2007. p. 73–85.
[11]."Plagiarism Detection-Different Methods and Their Analysis: Review", S.A.Hiremath M.S.Otari
[12]."Fast approximate string matching in a Dictionary", Recardo-Baeza-Yates Gonzalo Navarro.