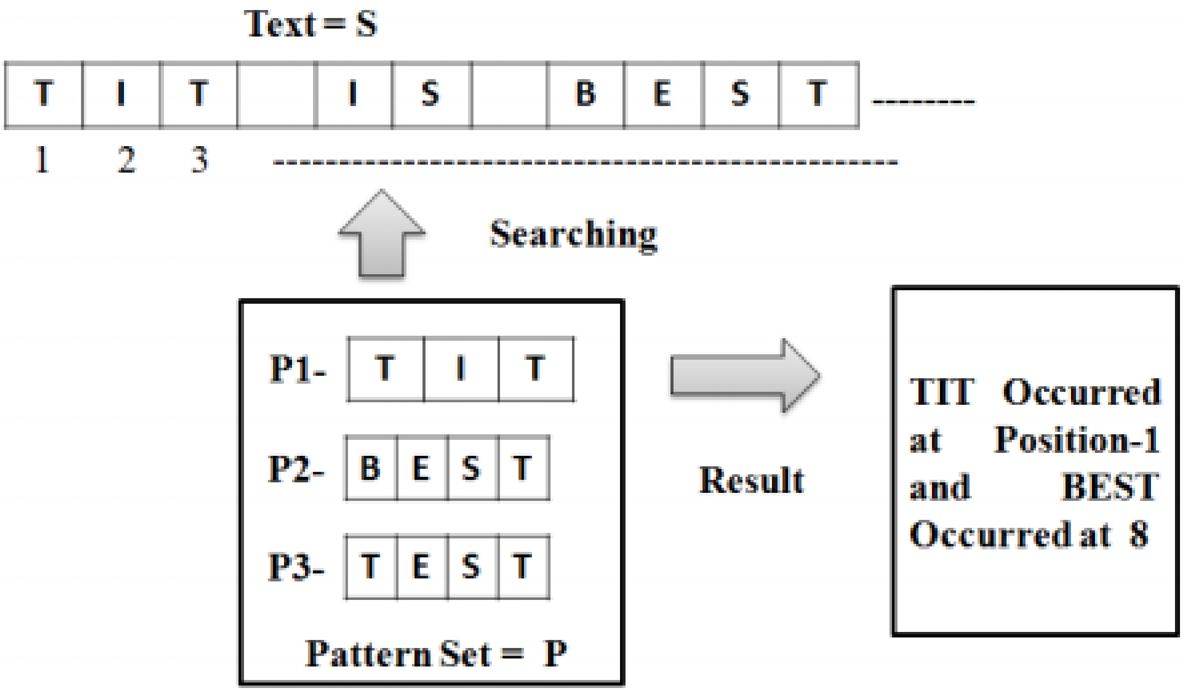
FPT UNIVERSITY

DATA STRUCTURES AND ALGORITHMSI

Giảng viên: Tiến sĩ Bùi Thanh Hùng
Trưởng Lab Khoa học Phân tích dữ liệu và Trí tuệ nhân tạo
Giám đốc chương trình Hệ thống thông tin
Đại học Thủ Dầu Một
Đại học FPT
Email: hungbt3@fe.edu.vn
Website: https://sites.google.com/site/hungthanhbui1980/


Text = S
T I T I S B E S T --------
1 2 3 --
Searching
P1- T I T
P2- B E S T
P3- T E S T
Pattern Set = P
Result
TIT Occurred at Position-1 and BEST Occurred at 8

String Matching

Exact String Matching Problem
Approximate String Matching Problem
Multiple String Matching Problem

Exact String Matching Problem

Exact String Matching

1. Brute Force Algorithm	Search in O(n.m)
2. Searching with automation	O(n)
3. Rabin Karp Algorithm	O(n.m)
4. Shift OR Algorithm	O(n)
5. Morris Pratt Algorithm	O(n+m)
6. Knuth- Morris Pratt Algorithm	O(n+m)
7. Colussi Algorithm	O(n)
8. Forward DAWG Matching algorithm	O(n)
9. Boyer Moore Algorithm	0(n.m)
10.Quick Search algorithm	O(n.m)

Brute Force Algorithm

- No preprocessing phase.
- Constant extra space needed.
- Always shifts the window by exactly 1 position to the right.
- Comparisons can be done in any order.
- Searching phase in O (m×n) time complexity.
- 2n expected text character comparisons.

Searching with automation

- Builds the minimal Deterministic Finite Automaton recognizing the language $\sum{}^*x$.
- Extra space in $O(m \times \sigma)$ if the automaton is stored in a direct access table.
- Preprocessing phase in $O(m \times \sigma)$ time complexity.
- Searching phase in $O(n)$ time complexity.

Rabin Karp Algorithm

- Uses an hashing function.
- Preprocessing phase in O(m) time complexity and constant space.
- Searching phase in O(m× n) time complexity.
- O(m+n) expected running time.

Shift OR Algorithm

- Uses bitwise techniques.
- Efficient if the pattern length is no longer than the memory word size of the machine.
- Preprocessing phase in O(m+σ) time and space complexity.
- Searching phase in O(n) time complexity(independent from the alphabet size and the pattern length).
- Adapts easily to approximate string matching.

Morris Pratt Algorithm

- Performs the comparisons from left to right.
- Preprocessing phase in O(m) space and time complexity.
- Searching phase in O(m+n) time complexity independent from the alphabet size.
- Performs at most 2n -1 text character comparisons during the searching phase.
- Delay bounded by m.

Knuth- Morris Pratt Algorithm

- Performs the comparisons from left to right.
- Preprocessing phase in O(m) space and time complexity.
- Searching phase in O(m+n) time complexity independent from the alphabet size.
- Performs at most 2n -1 text character comparisons during the searching phase.
- Delay bounded by logΦ (m) where Φ is the golden ratio(1+√5)/2.

Colussi Algorithm

- Refinement of the Knuth Morris Pratt algorithm.
- Partitions the set of pattern positions into two disjoint subsets. The positions in the first set are scanned from left to right and when no mismatch occurs the positions of the second subset are scanned from right to left.
- Preprocessing phase in O(m) time and space complexity.
- Searching phase in O(n) time complexity.
- Performs 3/2n text character comparisons in the worst case.

Forward DAWG Matching algorithm

- Uses the suffix automaton of x.
- O(n) worst case time complexity.
- Performs exactly n text character inspections.

Boyer Moore Algorithm

- Performs the comparisons from right to left.
- Preprocessing phase in $O(m+\sigma)$ time and space complexity.
- Searching phase in $O(m \times n)$ time complexity.
- n text character comparisons in the worst case when searching for non periodic pattern.
- $O(n/m)$ best performance.

Quick Search algorithm

- Simplification of the Boyer Moore algorithm.
- Uses only the bad character shift.
- Easy to implement.
- Preprocessing phase in $O(m+\sigma)$ time and $O(\sigma)$ space complexity.
- Searching phase in $O(m \times n)$ time Complexity.
- Very fast in practice for short patterns and large alphabets.

Approximate String Matching Problem

- Approximate string matching is a recurrent problem in computer Science which is applied in text searching, computational biology, pattern recognition and signal processing applications.
- The problem can be stated as follows: **For a length n and pattern of length m , we are supposed to find all the occurrences of pattern in the text whose edit distance to the pattern is at most K.**
- The edit distance between two strings is defined as minimum number of character insertion, deletion and replacements needed to make them equal.

- Dynamic Programming is a method to solve approximate string matching problem. For a Text of length n and pattern of length m ,the dynamic programming method returns a time complexity of O(nm).

- Bit Parallelism results in faster approximate string matching algorithm like the the fastest nonfiltering algorithms in practice are the $O(kn[m/w])$ algorithm of Wu & Manber,
- The $O([km/w]n)$ algorithm of Baeza-Yates & Navarro
- The $O([m/w]n)$ algorithm of Myers, where m is the pattern length, n is the text length, k is the error threshold and w is the computer word size.

- The motivation for approximate string matching comes from low quality of text, heterogeneousness of databases, spelling errors in the pattern or text, searching for foreign names and searching with uncertainty

Multiple String Matching Problem

- we are given a text T = t1t2 : : : tn and want to search simultaneously for a set of strings P = {p1 , p2 , … pr } Where pi =pi 1 pi 2………… pi mi is a string of length mi, for i=1…..r

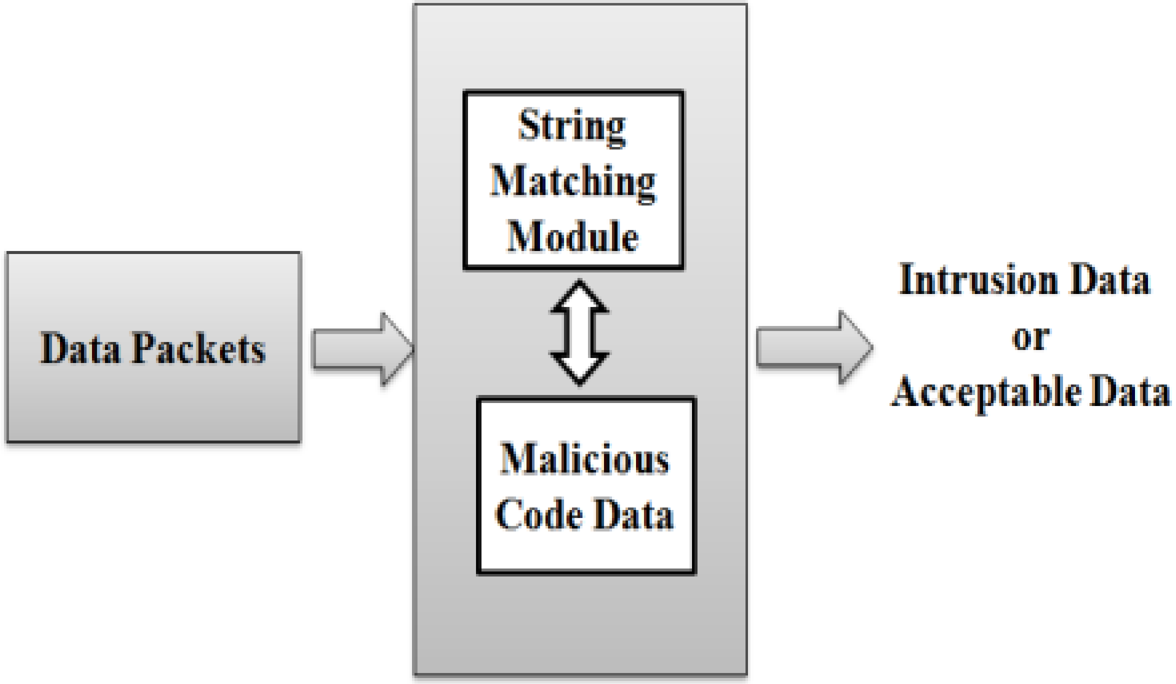
- Aho-Corasick string matching algorithm
- Rabin Karp String matching Algorithm
- Commentz-Walter algorithm

Applications of String Matching

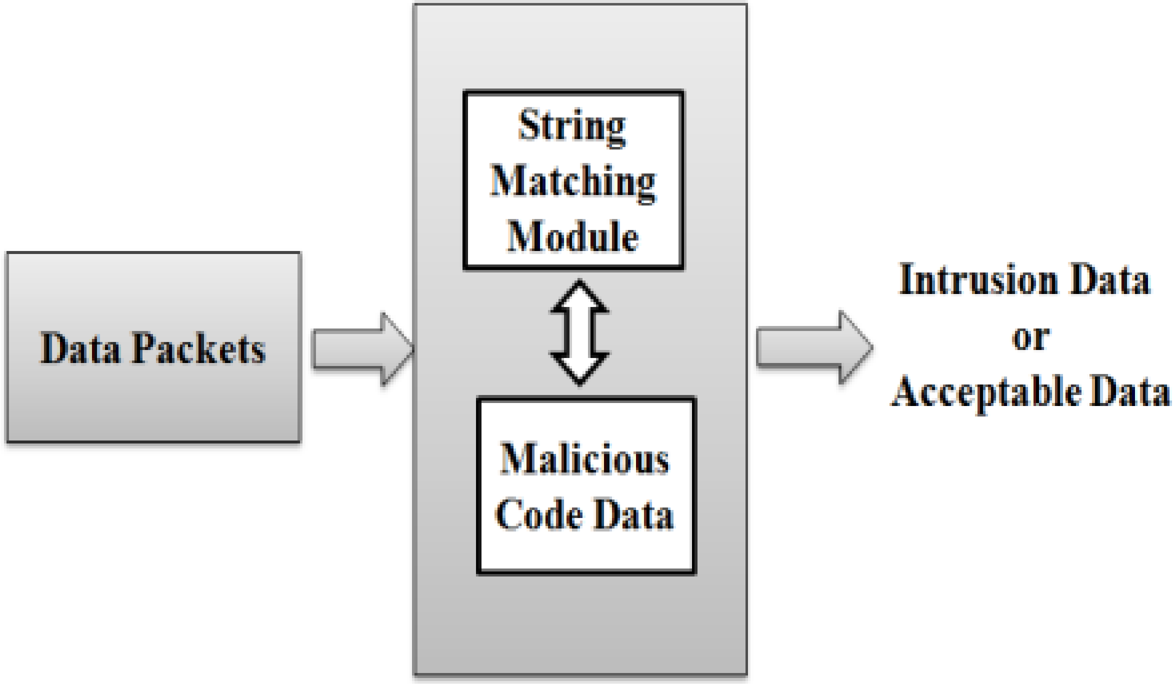
Applications of String Matching

- Intrusion detection.
- String matching in detecting plagiarism.
- String matching in bioinformatics
- String matching in Digital Forensics
- Text Mining Research
- String matching Based Video Retrieval

Intrusion detection

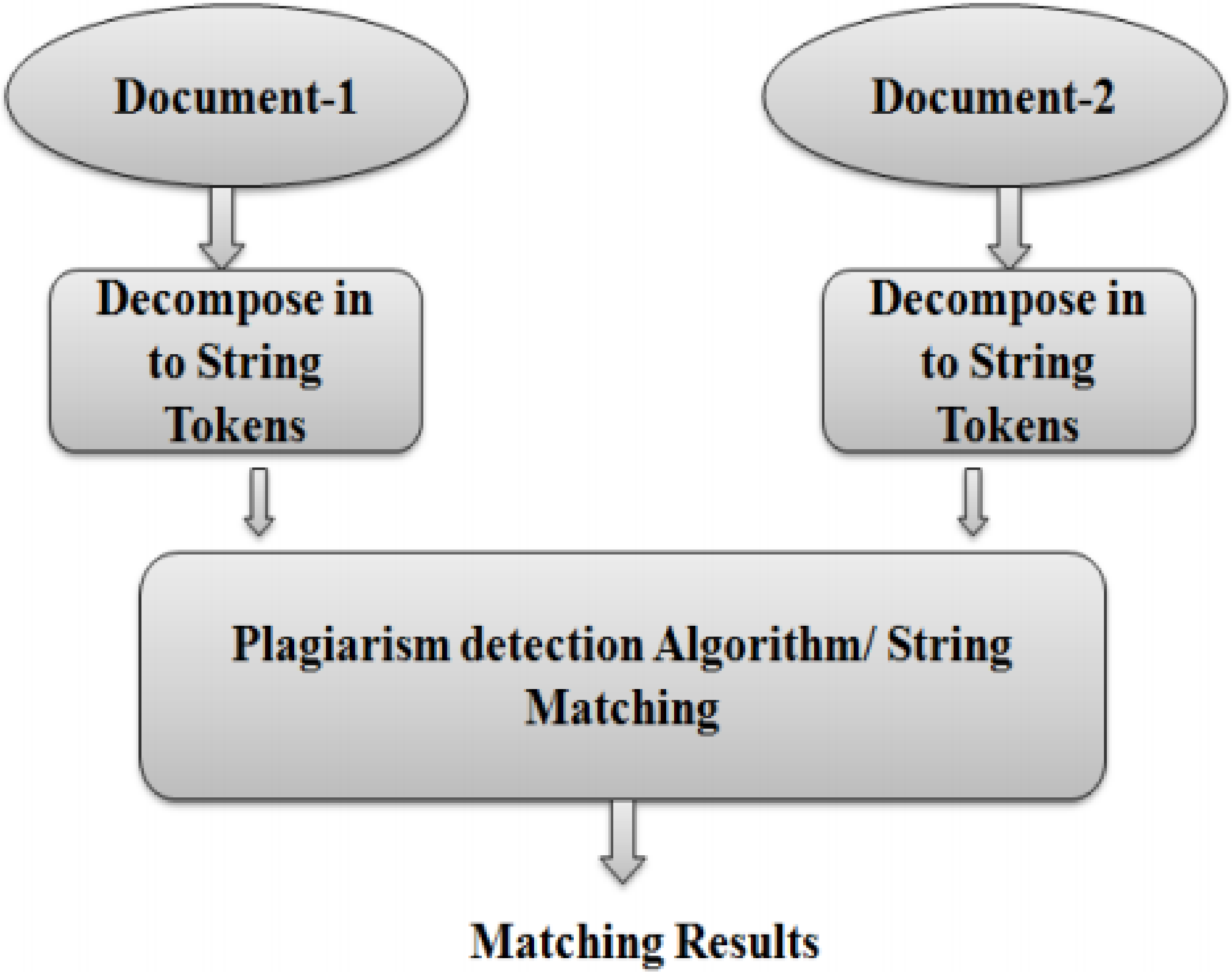


Intrusion detection



Intrusion detection is a set of techniques and methods that are used to detect suspicious activity both at the network and host level. Intrusion detection systems fall into two basic categories: signature-based intrusion detection systems and anomaly detection systems.

Plagiarism Detection



Document-1

Document-2

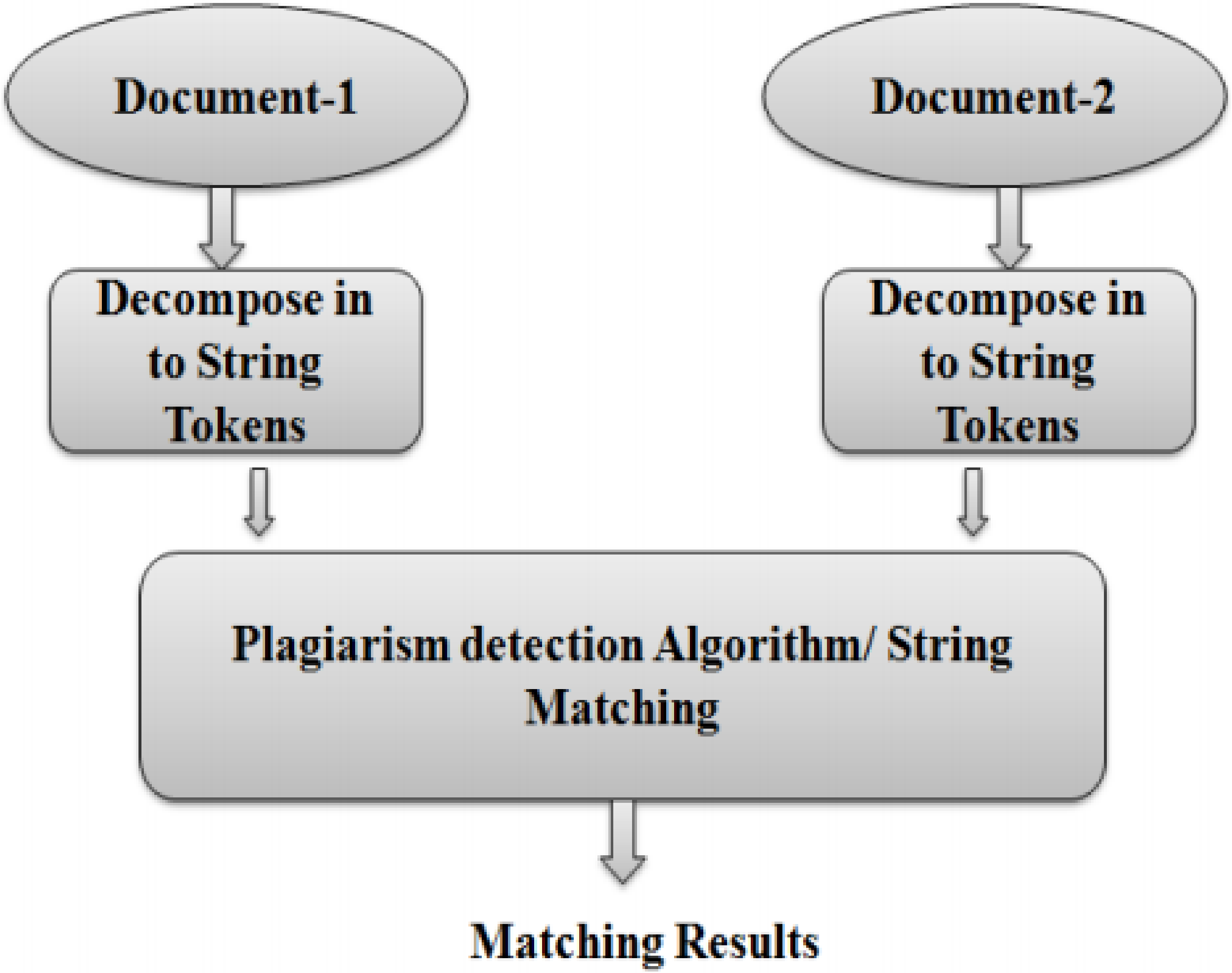
Decompose in to String Tokens

Decompose in to String Tokens

Plagiarism detection Algorithm/ String Matching

Matching Results

Plagiarism Detection



Document-1

Document-2

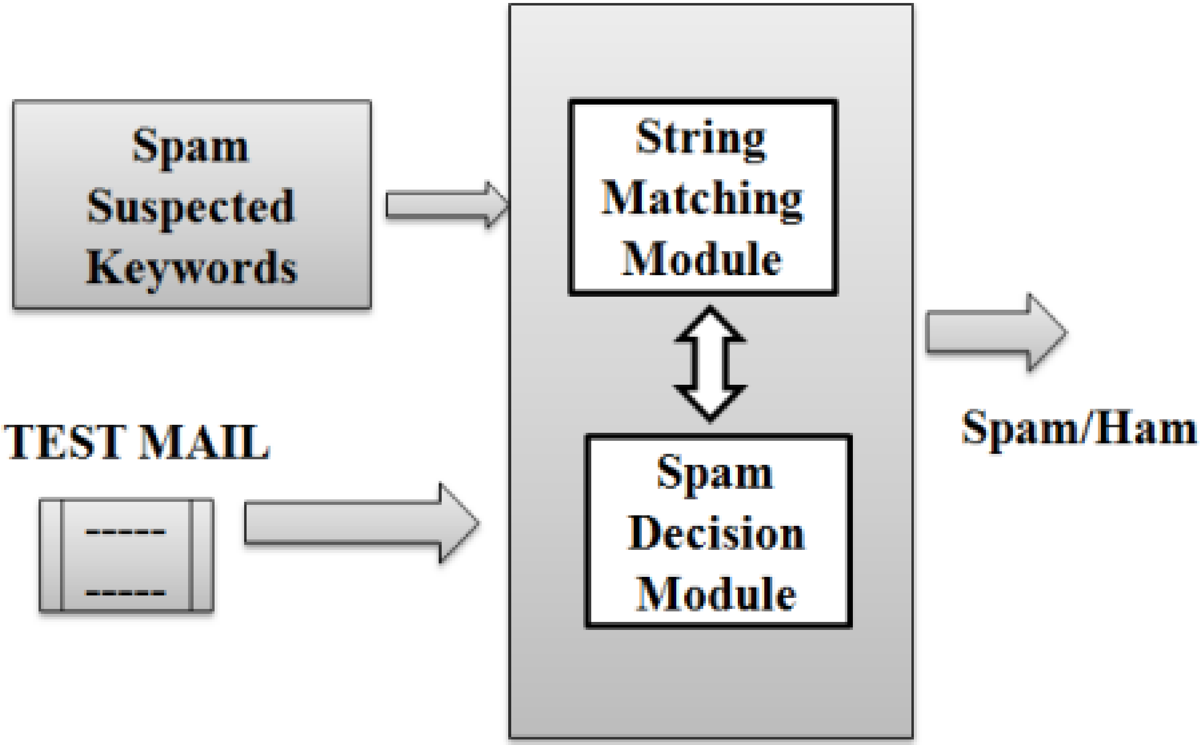
Decompose in to String Tokens

Decompose in to String Tokens

Plagiarism detection Algorithm/ String Matching

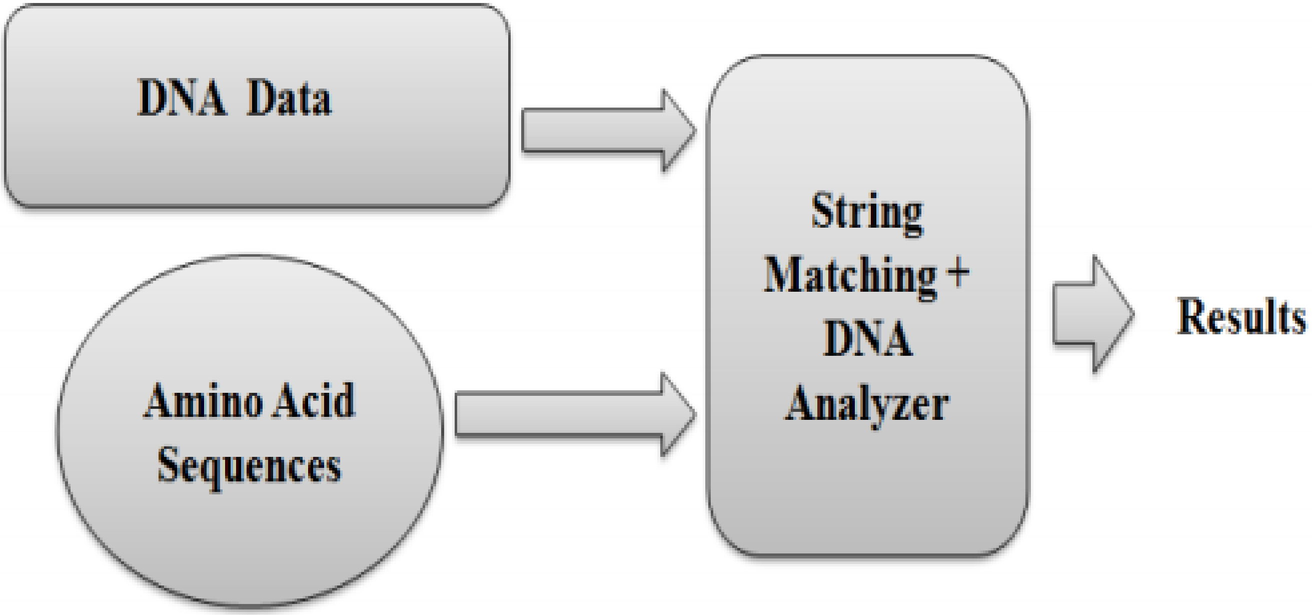
Matching Results

Spam filter

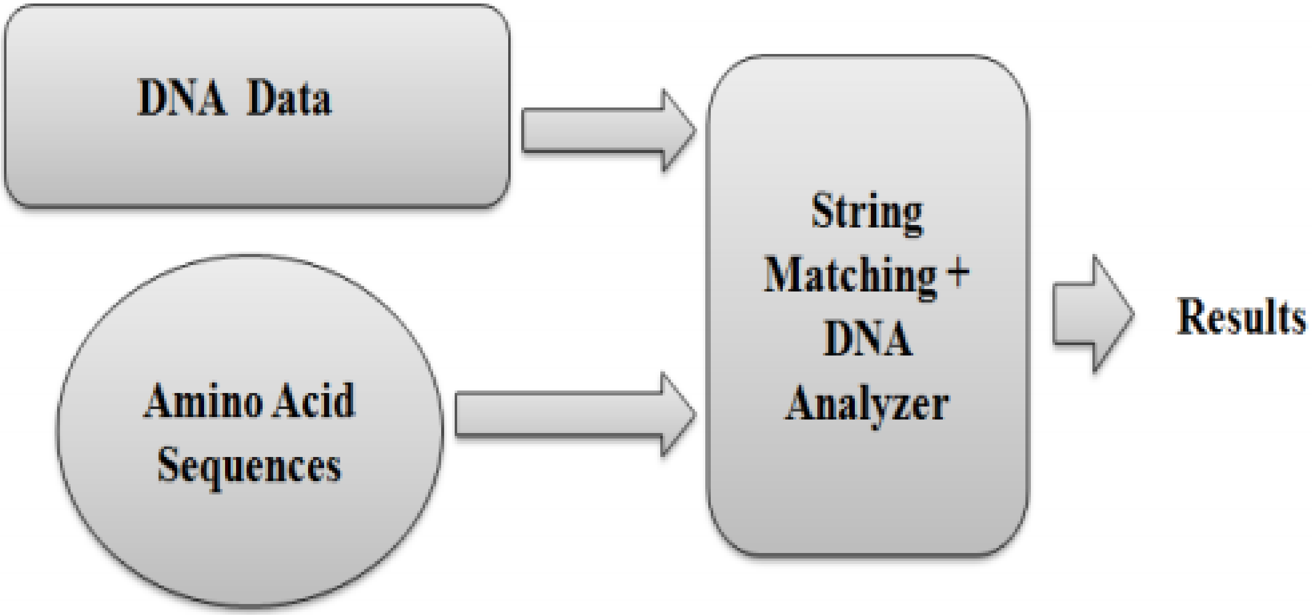

Spam Suspected Keywords
TEST MAIL

String Matching Module
Spam Decision Module
Spam/Ham

DNA Sequencing Module

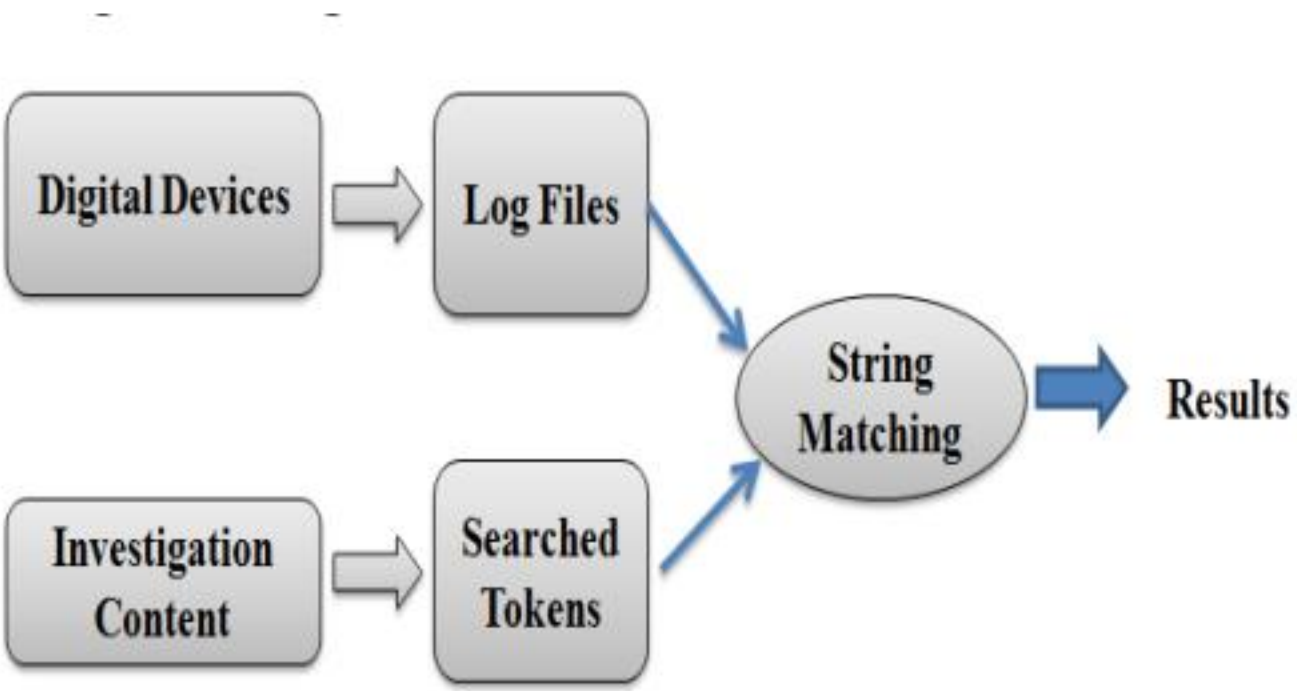


DNA Sequencing Module



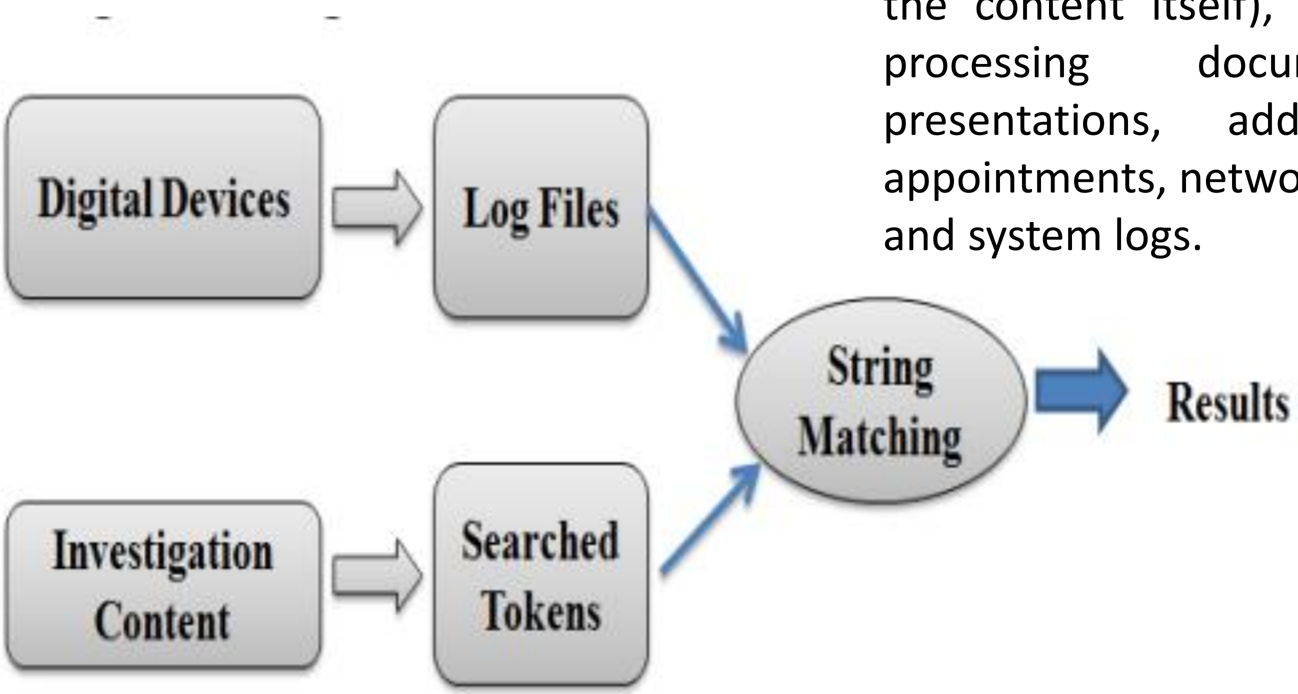
Bioinformatics is the application of information technology and computer science to biological problems, in particular to issues involving genetic sequences. String algorithms are centrally important in bioinformatics for dealing with sequence information.

Digital Forensics



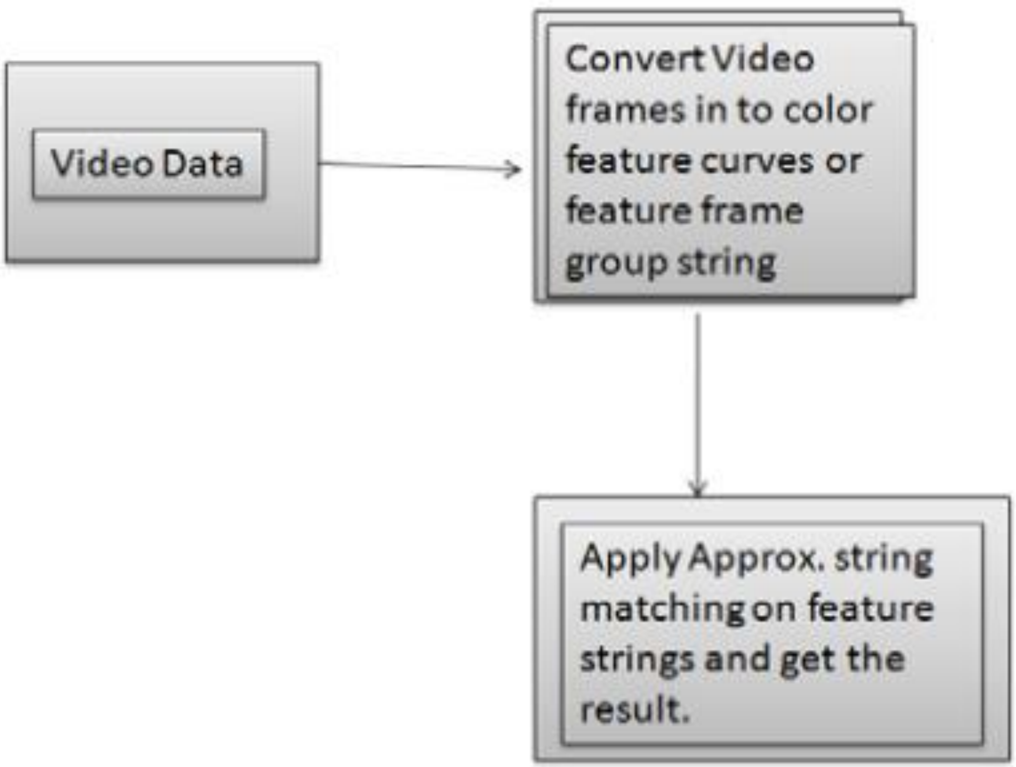
Digital Forensics

email, Internet browsing history (both logs and the content itself), instant messaging, word processing documents, spreadsheets, presentations, address books, calendar appointments, network activity logs,
and system logs.



Textual evidence is important to the vast majority of digital investigations.
Digital forensic text string searches are designed to search every byte of the digital evidence, at the physical level, to locate specific text strings of interest to the investigation.

String matching Based Video Retrieval

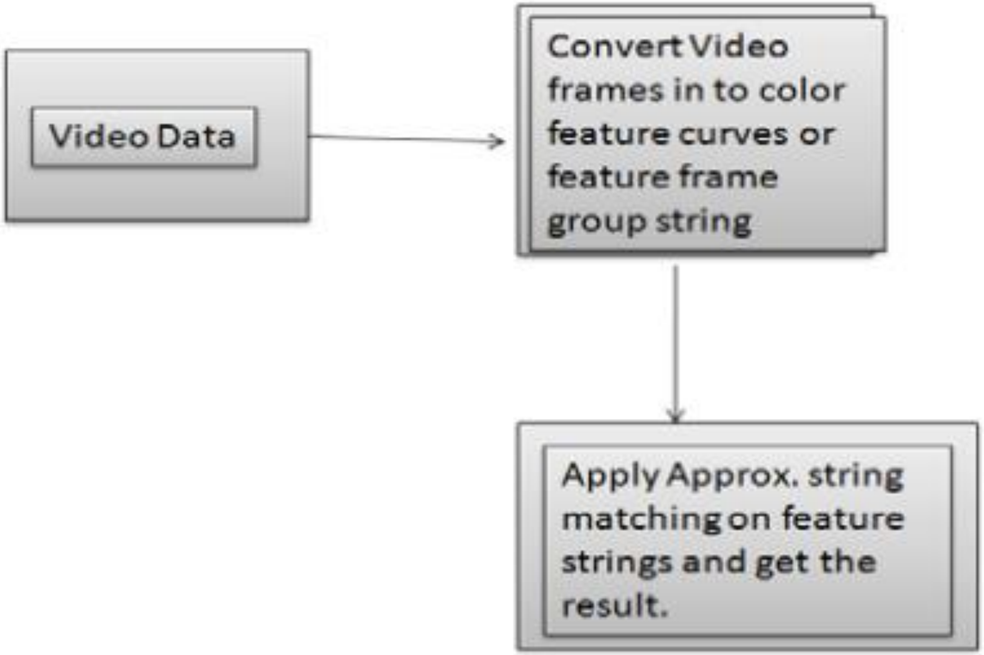


Video Data

Convert Video frames in to color feature curves or feature frame group string

Apply Approx. string matching on feature strings and get the result.

String matching Based Video Retrieval


Video Data

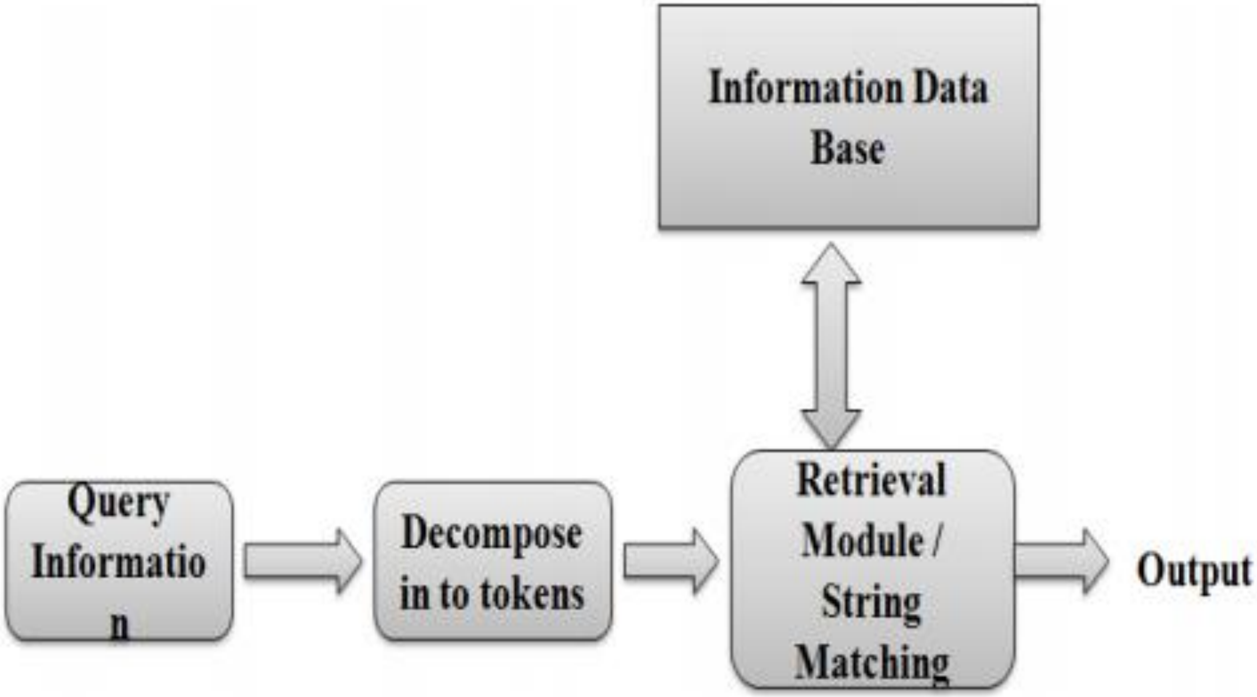
Convert Video frames in to color feature curves or feature frame group string

Apply Approx. string matching on feature strings and get the result.

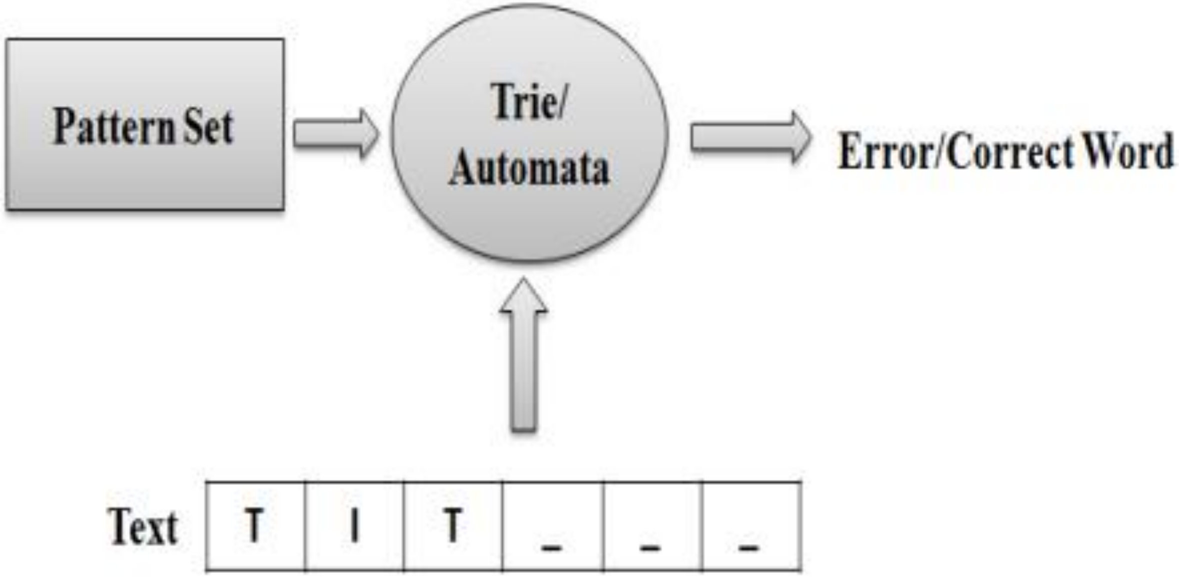
String matching can be effectively used to retrieve fast video as it uses the content based video retrieval in contrast with the traditional video retrieval which was slow and time consuming. String based video retrieval method first converts the unstructured video into a curve and marks the feature string of it. Approximate string matching is then used to retrieve video quickly.

1. **Information extraction**: identifies conceptual relationships, using known syntactic patterns and rules within a language.
2. **Topic tracking**: facilitates automated information filtering, wherein user interest profiles are defined and fine-tuned based on what documents users read.
3. **Content summarization**: abstracts and condenses document content.
4. **Information visualization**: represents textual data graphically (e.g. hierarchical concept maps, social networks, timeline representations).
5. **Question answering**: automatically extracts key concepts from a submitted question, and Subsequently extracts relevant text from its data store to answer the question(s).
6. **Concept linkage**: identifies conceptual relationships between documents based on Transitive relationships between words/concepts in the documents.
7. **Text categorization/classification**: automatically and probabilistically assigns text documents into predefined thematic categories, using only the textual content (i.e. no metadata).
8. **Text clustering**: automatically identifies thematic categories and then automatically assigns text documents to those categories, using only textual content (i.e. no metadata).

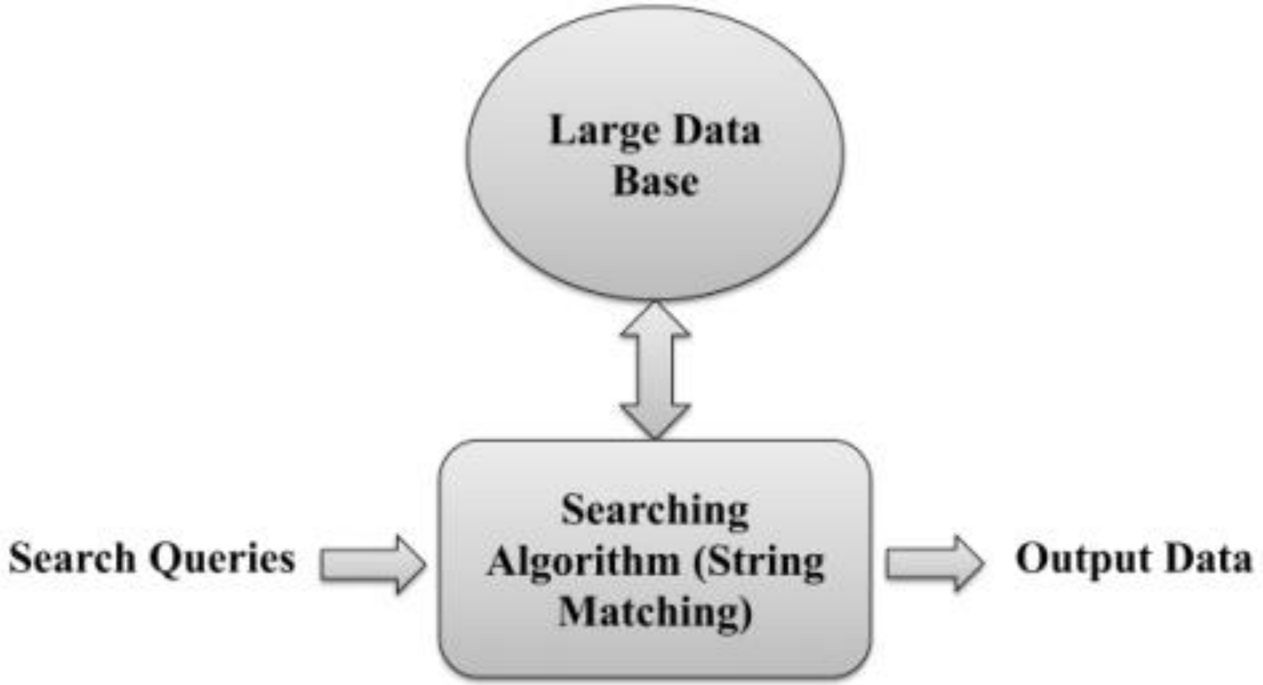
Information Retrieval Modal



Spell Checker



Search Engine Module



- Prefix function có tác dụng gì trong giải thuật KMP?
- Vai trò của String matching trong Text Processing?
- Tìm hiểu về Natural Language Processing

- Cài đặt 2 giải thuật KMP và Brute-Force