情報檢索

Information Retrieval

제4장 문서 검색

2012. 10. 08

가천대학교 IT대학

컴퓨터미디어융합학과

목 차

Why Information(document) Retrieval?

Information retrieval is the same as

Document retrieval.

There are many Documents even in the computer era.

아는 것이 힘이다.

Paperless society → 문서의 디지털화 → 문서 검색

4.1 개요

Collection: 문서의 집합

Document: 문서. 자연어 문장의 나열

색인어: 의미를 가지는 키워드 또는 키워드 무리

질의: 문서를 찾기 위한 사용자의 요구

정보검색의 문제 = 질의의 문제

색인어 공간에서 표현되는 문서와 질의를 어떻게 연결하는가?

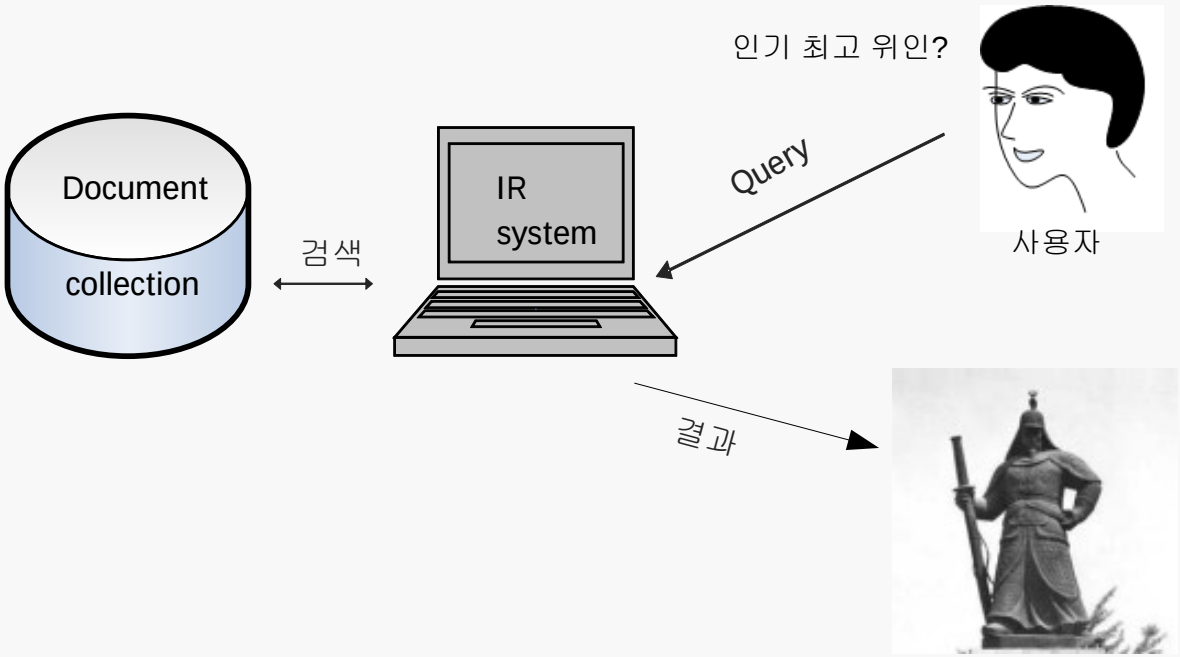
검색 결과에서 가장 적합한 문서를 어떻게 결정하는가?

문헌에서 색인어/검색어/주제어keyword로 사용되는 것은?

4.1 개요

Goal of IR

Goal: find documents relevant to an information need from a large document set



4.1 개요

Collection: 문서의 집합

Document: 문서. 자연어 문장의 나열

색인어: 의미를 가지는 키워드 또는 키워드 무리

질의: 문서를 찾기 위한 사용자의 요구

정보검색의 문제 = 질의의 문제

색인어 공간에서 표현되는 문서와 질의를 어떻게 연결하는가?

검색 결과에서 가장 적합한 문서를 어떻게 결정하는가?

문헌에서 색인어/검색어/주제어keyword로 사용되는 것은?

명사. Why? 의미 전달이 크기 때문에, 전처리?

4.1 개요

문서검색의 주요 주제

주 제	내 역
질의	질의와 문서를 무엇으로 어떻게 연결할 것인가?
	가장 적합한 문서인지를 어떻게 결정하는가?
검색 방법	사용자가 원하는 가장 관련된 문서를 어떻게 찾을 수 있는가?
색인	효율적인 검색이 가능한 색인을 어떻게 구축하는?
키워드	문서를 개념 기반으로 어떻게 검색하는가?

4.1 개요

Overview

IR is concerned with **indexing** and **retrieval** of textual documents.

Searching for pages on the WWW is the most recent "**killer application**".

Concerned firstly with retrieving relevant documents to a query.

Concerned secondly with retrieving from large sets of documents efficiently.

4.1 개요

Relevance

Relevance is a subjective judgment and may include:

- Being on the **proper subject**.
- Being **timely** (recent information)
- Being **authoritative** (from a trusted source).
- Satisfying the **goals of the user** and his/her intended use of the information (information need)

4.1 개요

Problems with keywords

- May not retrieve relevant documents that include synonymous terms.
  - “restaurant” vs. “café”
  - “PRC” vs. “China”
- May retrieve irrelevant documents that include ambiguous terms.
  - ‘bat’ (baseball vs. mammal)
  - ‘Apple’ (company vs. fruit)
  - ‘bit’ (unit of data vs. act of eating)

** 개념기반 검색의 필요성 대두

4.1 개요

IR Problem

First applications: in libraries (1950s)

ISBN: 0-201-12227-8

Author: Salton, Gerard

Title: Automatic text processing: the transformation, analysis, and retrieval of information by computer

Editor: Addison-Wesley

Date: 1989

Content: <Text>

external attributes and internal attribute (content)

Search by external attributes = Search in DB

IR: search by content

4.1 개요

Possible Approach

1. **String Matching** (linear search in documents)

   - Slow

   - Difficult to improve

2. **Indexing**

   - fast

   - Flexible to further improvement

4.1 개요

Main problems in IR

문제점	내 역
문서 색인	문서 내용을 색인으로 어떻게 잘 표현할 수 있는가?
질의 평가	질의를 문서와 비교할 수 있도록 어떻게 표현할 것인가?
시스템 평가	시스템이 얼마나 우수한가? 정확률: 검색된 문서가 관련이 있는 문서인가? 재현율: 관련 문서가 얼마나 검색되었는가?

4.1 개요

주요 문제점의 대책

해결 방식	대 책
문자열 처리	문서의 유사성을 문자열을 비교하여 결정. 문자열 처리 속도 향상 필요
색인 구축	색인을 구축하고 정비하는 부담을 개선
개념기반 검색	단어의 의미를 파악하여 사용자의 질의 의도에 적합하게 검색
지능형 검색	에이전트 등을 이용하여 사용성, 신뢰성, 정확성을 향상

4.1 개요

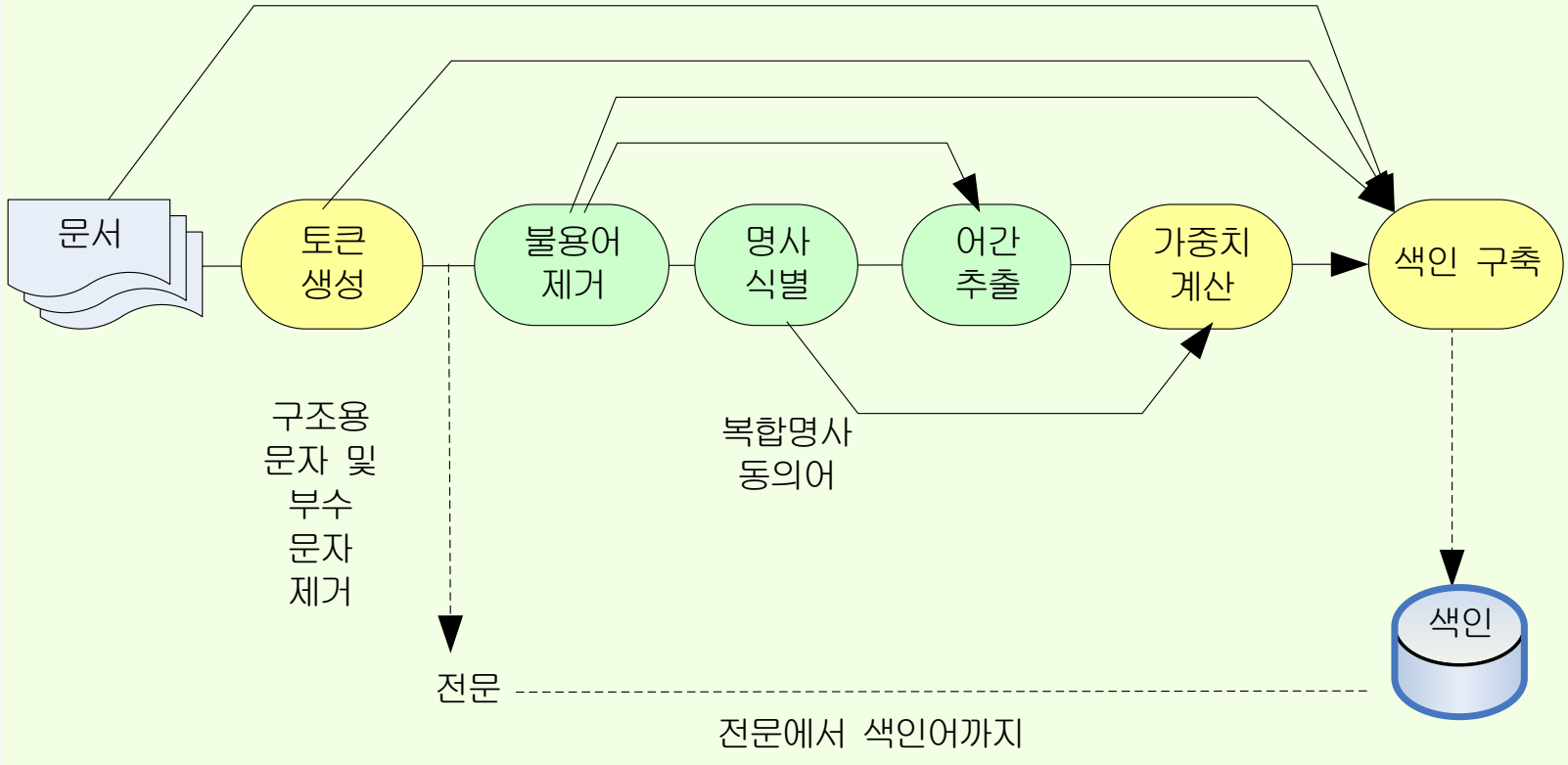
Intelligent IR

1) 사용되는 단어들의 의미를 고려한다.
2) 질의에 있는 단어들의 순서를 고려한다.
3) 직접 또는 간접적인 환류를 적용한다.
4) 원시 자료의 권위인 정확성과 신뢰성을 고려한다.
5) 에이전트 기법으로 분산처리 기능을 확대한다.

4.2 색인어 처리 텍스트 연산

색인어 추출과 색인 구축:

색인 구축을 위한 전처리 작업



4.2 색인어 처리

4.3.1 텍스트 연산: 색인 작성 절차

구분	작 업	내 역
1	토큰 생성	문서의 전체 문자열에서 단어를 식별.
2	불용어 제거	관사, 전치사, 동사 등의 의미 없는 단어 제거
3	복합명사, 동의어 식별	복합명사 분리, 시소러스를 이용하여 동의어 식별
4	어간 추출	접두사, 접미사, 파생어, 활용어 들을 감안하여 어간을 추출
5	가중치 계산	색인어의 빈도수와 가중치를 계산
6	색인 구축	용어와 가중치를 포함하여 역색인 구축

4.2 색인어 처리 텍스트 연산

불용어 1

통제되지 않는 어휘

일상적인 단어, 단어의 변형, 동의어의 사용

the, of, and, to, a, in : 인쇄된 텍스트의 20~30%

출현빈도가 높은 단어: 부정적 영향

단어 빈도 측정 : 빈도의 차이에 대한 비교 의미 감소

단어 자체의 의미가 없기 때문에 비생산적인 처리과정을 유발

해결 방안

불용어 목록 또는 negative dictionary 개발

일반적인 불용어 목록의 크기 : 250 ~ 300 단어

문헌 비교 및 검색 과정의 단순화 및 효율성 증대

4.2 색인어 처리 텍스트 연산

불용어 2

구에 대한 불용어 처리 방법

To be, or not to be”

불용어로 구성된 구를 고려하기 위한 특별한 기법 적용해야.

불용어 식별을 위한 자료 구조

이진트리

불용어 목록의 크기가 작고 일반적으로 알려져 있는 경우

해싱기법

각 불용어에 상호 구별될 수 있는 해쉬 함수 값 할당

Trie 구조

불용어에 대한 문자 단위 확인

Ex) the, then, to와 technology

4.2 색인어 처리 텍스트 연산

어간법 Stemming

하나의 단어가 다양한 형태로 파생될 때, 스테밍 알고리즘 이용

ex) computer, computers, computing, compute, computes, computed, computational, computationally, computable

스테밍 알고리즘

- 어근에 도달하기 위해 단어 끝부분을 반복적으로 제거

  Ex) Computationally의 경우

  : computational →computation → computa → comput

- 접두사의 제거의 어려움
  - 접두사인지, 단어의 일부인지 구분 모호
  - Ex) **im**possible v.s. **im**mediately

4.2 색인어 처리 텍스트 연산

어간법 문제점

단어 끝부분의 잘못된 제거

ex) ‘bed’에서 끝부분 ‘ed’

단어의 어근(stem) 자체가 변하는 경우

ex) knife→knives

문헌 전체에 대한 스테밍 작업의 부하

단어의 파생 및 변화 : 전체 문헌 두성 단어의 5~10%

스테밍을 위한 과도한 처리 부하 발생 가능성

해결 방안

사용자 질의 자체를 스테밍한 후 와일드 문자(*)로 대체 검색

4.2 색인어 처리

4.2.2 Thesaurus 관련어집

정의

- 용어 상호간의 관계를 정의한 사전
  동의어, 유사어, 광의어, 협의어, 관련어, 유사어, 반의어
  예: post a letter 와 mail a letter
- 용어들 사이의 관계 정보를 제공하는 어휘 도구.

  용어들의 계층 구조. 용어는 단어 또는 구.
- 용어들을 뜻의 관점에서 분류하여 체계화한 자료집

4.2 색인어 처리

4.2.2 Thesaurus 관련어집

목적

1) 동의어 등을 질의에서 사용할 때 단어의 변형으로 처리.
2) 문서를 작성하고 관리할 때 어휘를 통제하기 위해서 사용
   .
3) 질의 과정에서 질의의 검색 범위를 확대하기 위해서 사용
   .

4.2 색인어 처리 시소러스

Thesaurus 목적

- 큰 컬렉션으로부터 관련 문서를 효과적으로 검색하기 위해서

Thesaurus 실례

abstract data types

BT	data structure
TT	file organization
NT	data encapulation
	inheritance
	persistent objects
RT	OO programming
	type theory
CC	Jan. 1993
PT	data structure

See also: cross reference

NT: narrower term
BT: broader term
TT: top term
RT: related term
PT: prior term
UF: used for
CC: Classcification code
DI: Data of Input
USE: 대체어

INSPEC Thesaurus 2010

4.2 색인어 처리 시소러스

색인

문서를 표현(색인)하기 위해 가장 적절한 시소러스 용어를 선택

검색

질의 재 작성 과정에서 가장 적절한 용어를 선택 가능

충분하게 검색되지 않았다면 시소러스 관계를 다라 질의 확장

너무 많이 검색되면, 시소러스에서 더 구체적인 용어 선택

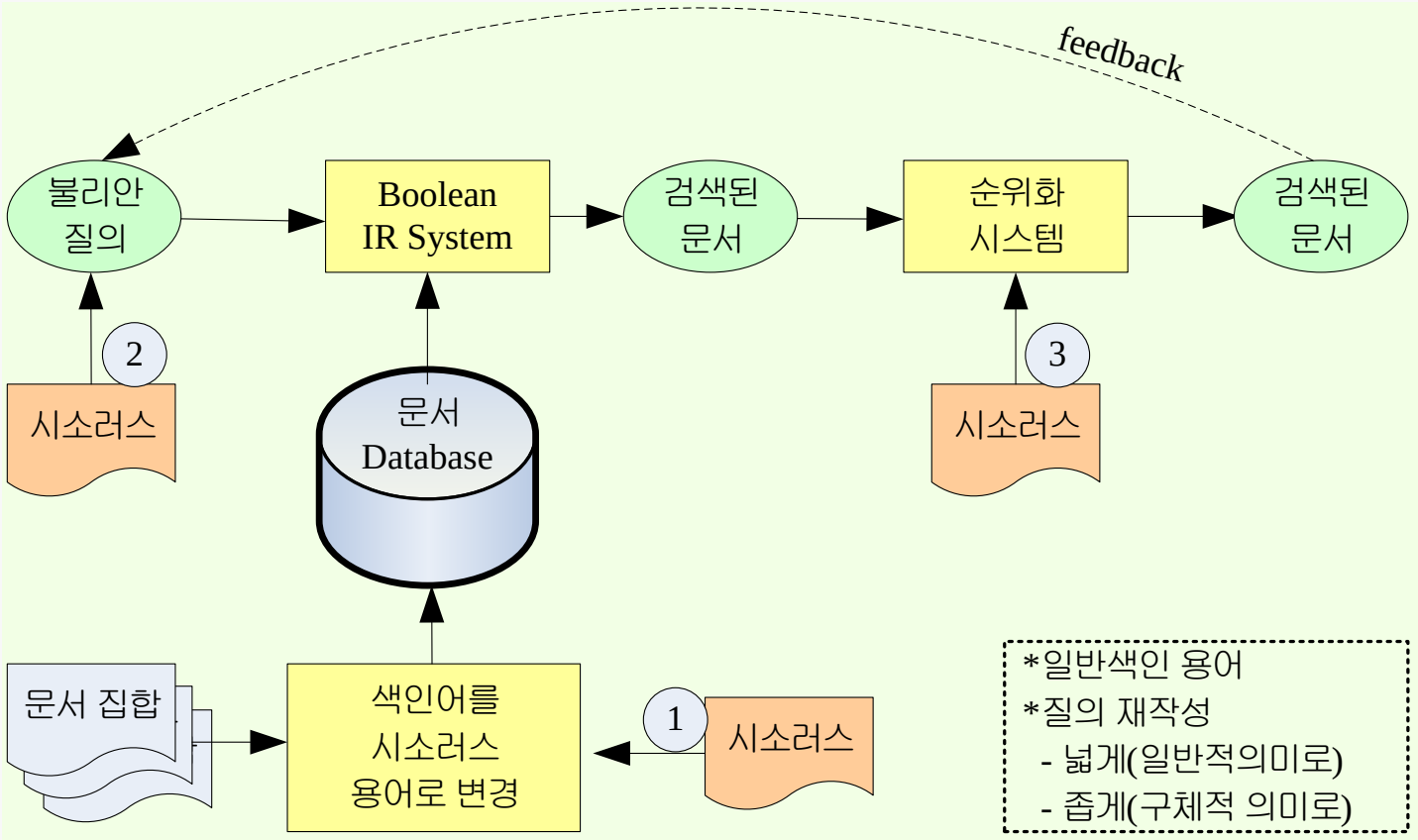
구축 방법

- 수동: 매우 개념적인 작업, 많은 지식과 노력 필요
- 자동: be challenging

4.2 색인어 처리 시소러스

Thesaurus의 역할

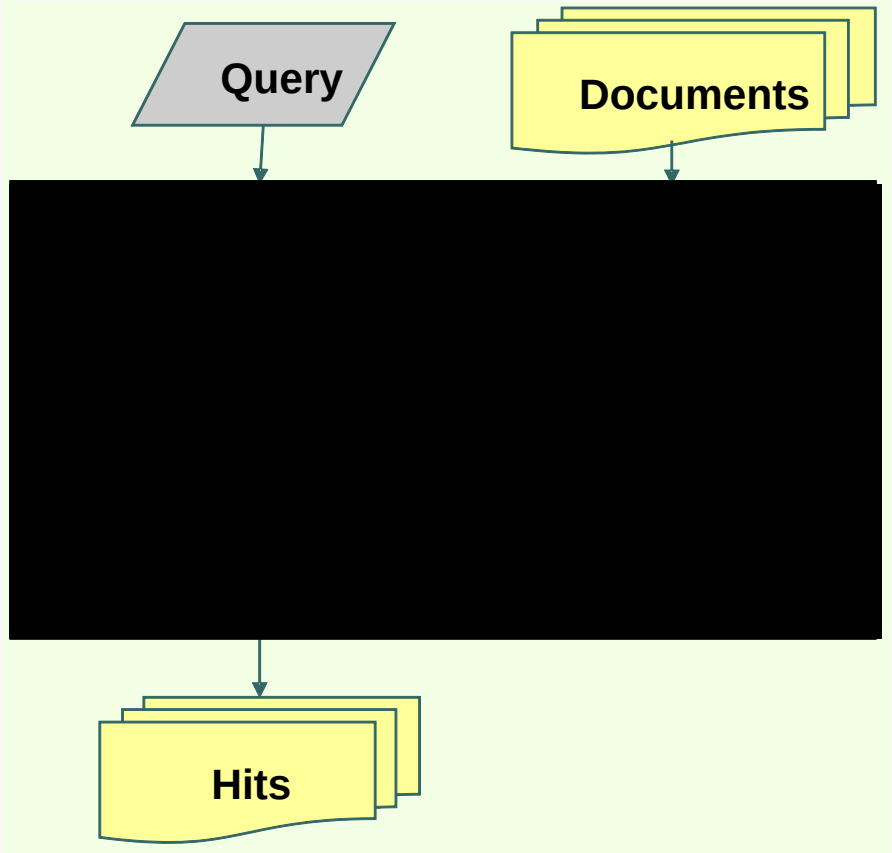
-



4.3 문서검색 System 구조

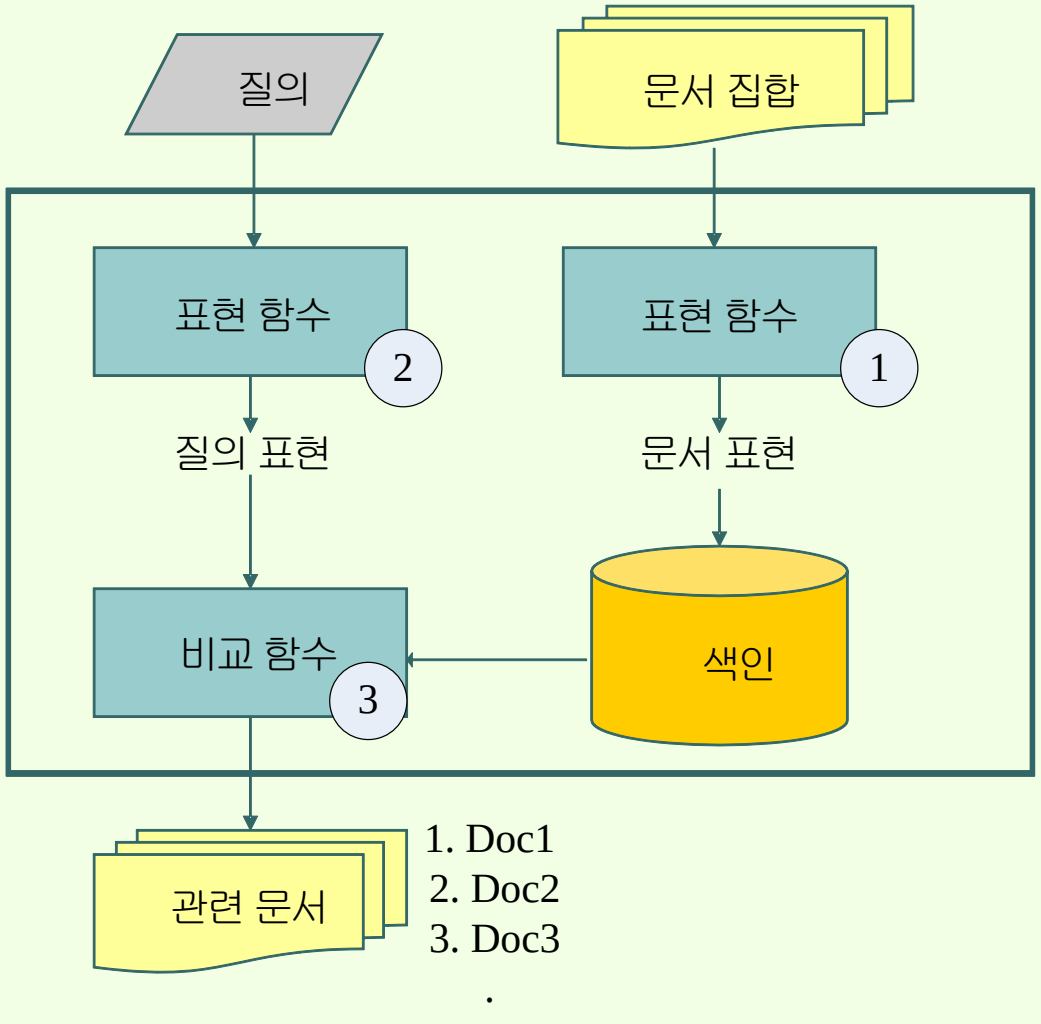
4.3.1 IR System 구조

Black Box



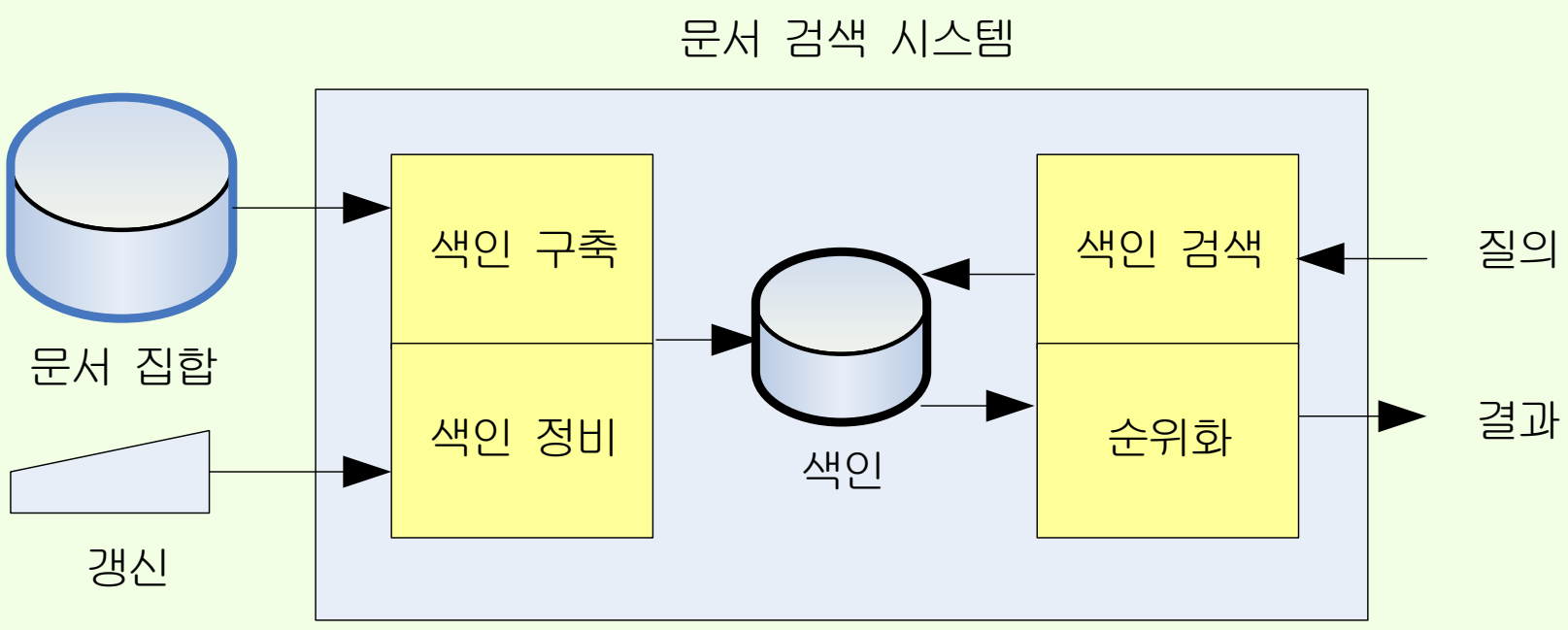
4.3 문서검색 System

4.3.1 IR System Inside



4.3 문서검색 System

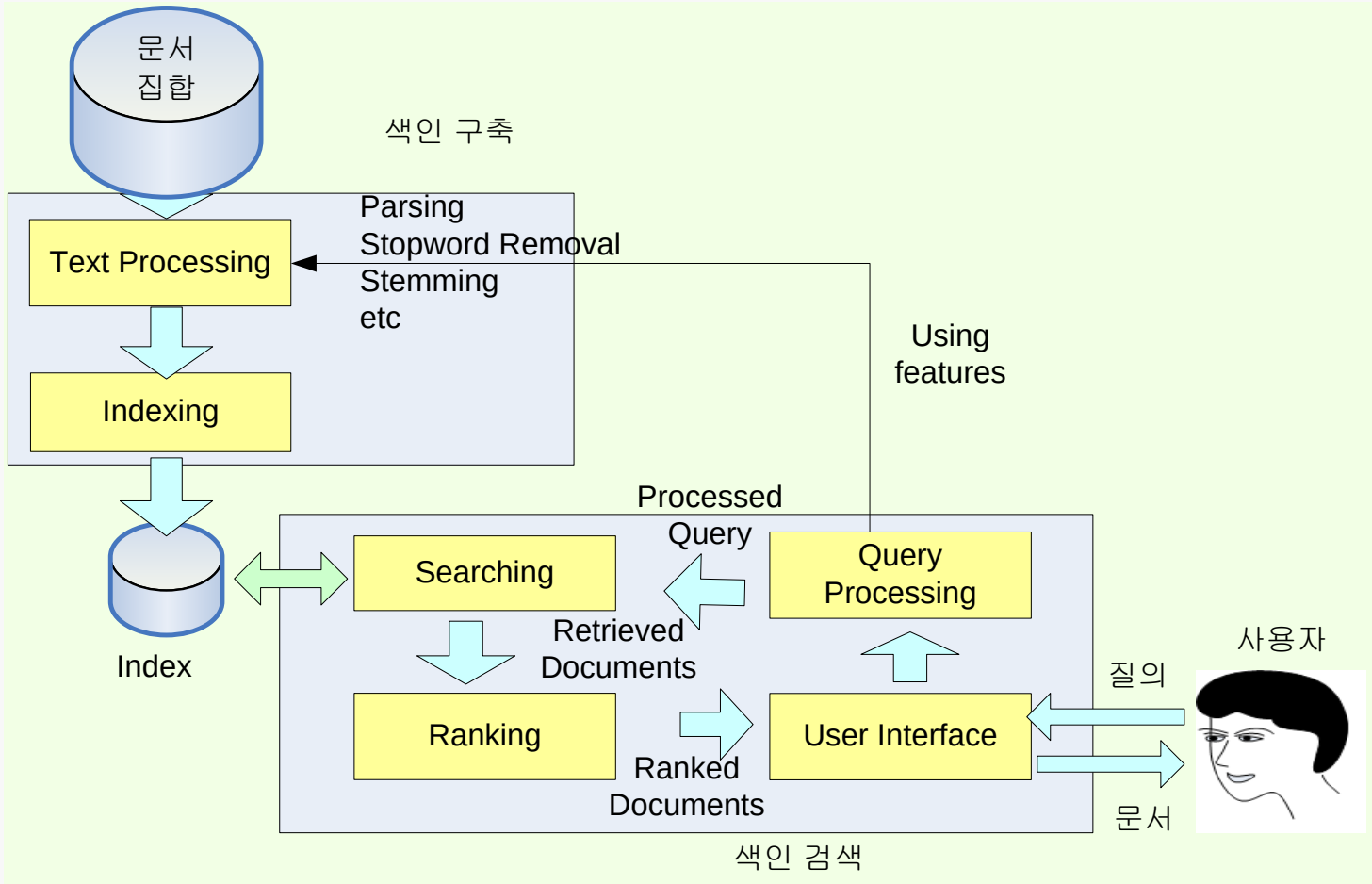
4.3.2 문서 검색 시스템



4.3 문서검색 System

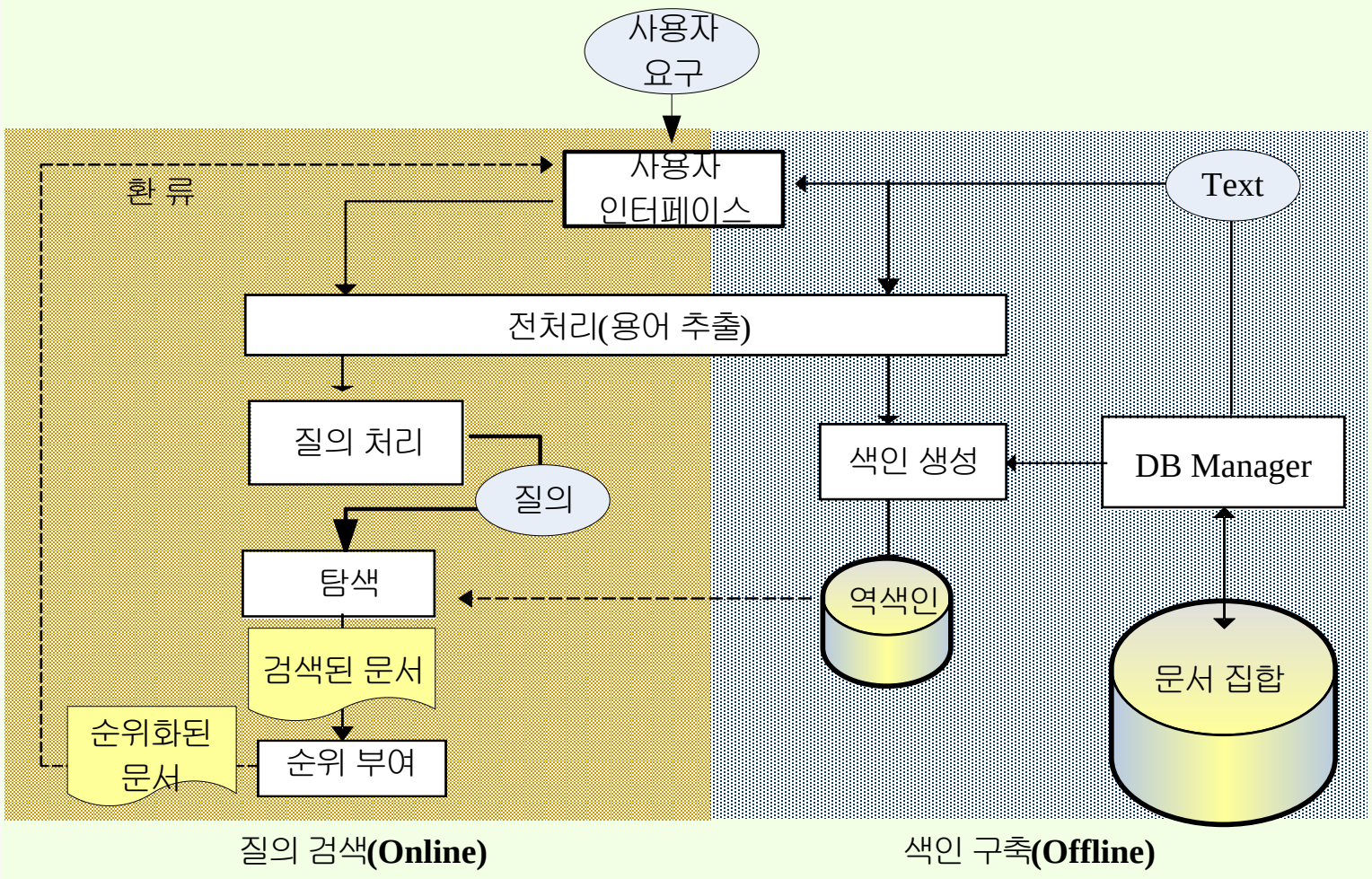
IR System 구조

1)



4.4 문자열 탐색 알고리즘

4.4.1 문자열 자료처리



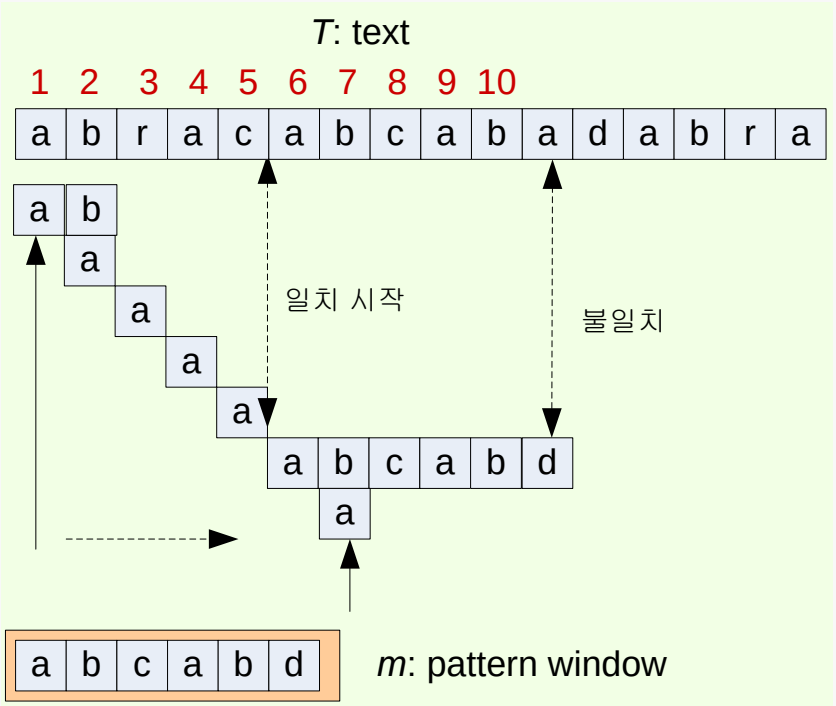
4.4 문자열 탐색 알고리즘

텍스트 탐색

정확한 문자열 정합 exact string matching

4.4.1 Naïve 알고리즘: Brutal Force

가장 간단한 문자열 탐색: 많은 알고리즘들의 기본형



4.4 문자열 탐색 알고리즘 Brutal Force Algorithm

text: 탐색 파일, 길이 n

pat: 탐색 창, 길이 m

효율: 최대 O(mn),
평균 O(n)

```
BFSearch(text, n, pat, m)  // n 길이의 text에서
char text[], pat[];              // m 길이의 pattern 찾기
int     n, m;
{
  int   i, j, k, lim;            //i: text 변수, j: pat 변수
  lim = n – m + 1;               // lim: 검색 마지막
  for (i=1; i<=lim; i++)  {      // Search
    k = i;                       //k: text의 재시작 위치
    for (j=1; j<=m && text[k] == pat[j]; j++) k++;
    if  (j > m) Report_match_at_position(i-j+1);
  }
```

4.4 문자열 탐색 알고리즘

4.4.2 Knuth-Morris-Pratt 알고리즘

패턴 *m*과 텍스트 *T*를 창문 안에서 비교하다가 불일치하면
창문 이동 위치를 효율적으로 결정하는 알고리즘.

Idea:

전처리 과정: *next[]*

테이블 생성… O(*m*)

텍스트에 일치되는 패턴의 접두사가 존재하고,
이를 이용해서 일치되지 않은 텍스트의 첫 위치를 추론.

4.4 문자열 탐색 알고리즘 4.4.2 Knuth-Morris-Pratt

```
index: 1 2 3 4 5 6 7 8 9 10
t:       a z f b d a b c a d f t
m:     a b c a b d
next[] 0 0 0 1 2 0
```

(a) 패턴과 next[]

```
a b c a b d ..... 접미사
a b c a b d ..... 접두사
```

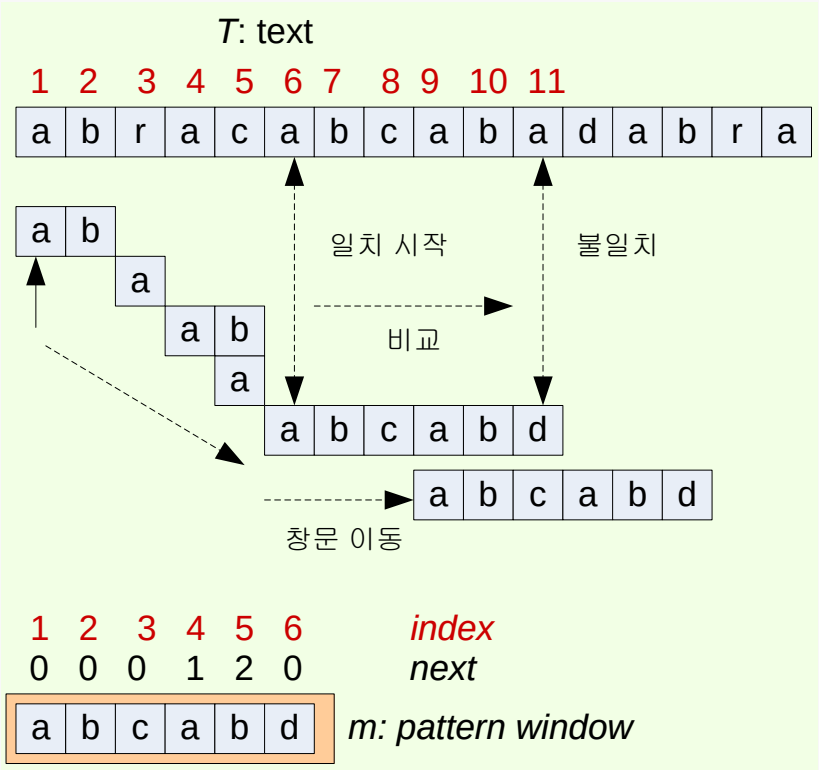
(b) 접두사와 접미사

4.4 문자열 탐색 알고리즘 4.4.2 Knuth-Morris-Pratt

- 두 번째 문자가 불일치하므로 이동하다가 6번째에서 일치
- 11번째 문자에서 불일치 발생

  이 때 일치한 b가 2이므로 2칸 앞으로 이동하여

  9번째부터 다시 비교



4.4 탐색 알고리즘

4.4.2 Knuth-Morris-Pratt 알고리즘

```
KMPSearch(text, n, pat, m)
char test[], pat[];
int   n, m;
{
   int j, k, resume;
   int next[MAX_PAT_SIZE];
   pat[m+1]  = CHAR_not_in_text;
   initnext(pat, m+1, next);
   resume = next[m+1];
   j = k = 1;
  do {          // Search
      if ( j==0 AND text[k] ==pat[j])
      { k++; j++; }
      else j = next[j];
      if ( j > m) {
         Report_match_at_position(k-m);
         j = resume;       }
   } while (k<=n);
  pat[m+1] = END_OF_STRING;
}
```

```
initnext(pat, m, next)  // initialize next[m]
char   pat[];
int      m, next[];
{
   int i = 1, j = next[1] = 0;
   do {
     if (j == 0 AND pat[i] == pat[j]) {
        i++;
        j++;
        if (pat[i] != pat[j] )
            next[i] = j;
        else
            next[i] = next[j];
      }
      else j = next[j];
   } while ( i <= m );
}
```

4.4 문자열 탐색 알고리즘

4.4.2 KMP 알고리즘

효율:

최대: O(*mn*)이지만 실제로 창문에서는 한 두번만 비교하게 되기 때문에 O(n)이 되기 쉽다.

같은 O(n)이라도 창문을 이용하여 이동하기 때문에 BF 알고리즘보다 효율이 우수하다.

4.4 문자열 탐색 알고리즘

4.4.3 Boyer-Moore 알고리즘

1977년 Boyer와 Moore가 발표.

윈도우 내부의 검사를 왼쪽부터 꺼꾸로 처리하는 알고리즘.

KMP의 next 테이블과 유사한 방법과 패턴의 접미사를 비교하여 윈도우를 가급적 많이 이동함으로써 검사 효율을 향상한다.

BM 알고리즘의 두 가지 핵심

1) 패턴의 오른쪽부터 문자를 비교하기 시작하여 왼쪽으로 탐색
2) 불일치 시에 오른쪽으로 이동할 길이를 패턴의 문자별로 미리 계산

4.4 문자열 탐색 알고리즘

창문 이동 방식

Bc: 나쁜 문자 방식과 Gc: 좋은 문자 방식

문자가 불일치하여 창문을 이동할 때 Bc와 Gc 중에서 큰 값 선택

Gc: 좋은 문자 방식

1) 두 문자가 같으면 왼쪽으로 이동한다.
2) 두 문자가 다르고 패턴과 일치하는 부분이 있으면 이동하여 일치시킨다.
3) 두 문자가 다르고 패턴과 일치하는 부분이 없으면 패턴 길이만큼 오른쪽으로 이동한다.

4.4 문자열 탐색 알고리즘 Boyer-Moore 알고리즘

Bc: 나쁜 문자 방식

1) 두 문자가 같으면 왼쪽으로 이동
2) 두 문자가 다르고 패턴이 없으면 패턴 길이 길이만큼 이동.
3) 두 문자가 다르지만 패턴에 있으면 두 문자를 일치시킨다.

패턴: STORE

```
A     S T A R     A S G E A S T H     S T O R E
S T O R E
          S T O R E
                S T O R E
                      S T O R E
                            S T O R E
```

4.4 문자열 탐색 알고리즘 Boyer-Moore 알고리즘

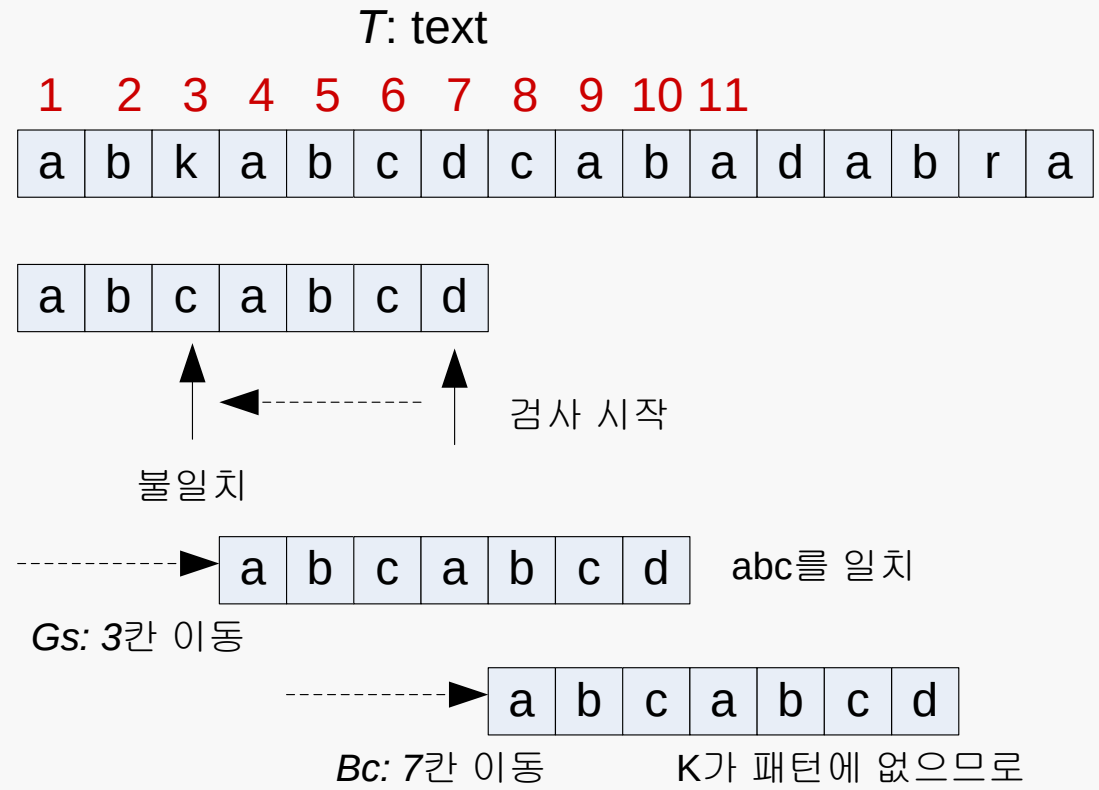
Gc: 좋은 문자 방식

1) 두 문자가 같으면 왼쪽으로 이동한다.
2) 두 문자가 다르고 패턴과 일치하는 부분이 있으면 이동하여 일치시킨다.
3) 두 문자가 다르고 패턴과 일치하는 부분이 없으면 패턴 길이만큼 오른쪽으로 이동한다.

```
a  b  c  b  b  a  b  b  b  b  a  a  b  c  a  b
a  b  c  a  b
      a  b  c  a  b
            a  b  c  a  b
                  a  b  c  a  b
                        a  b  c  a  b
```

4.4 문자열 탐색 알고리즘

4.4.3 Boyer-Moore 알고리즘



4.4 문자열 탐색 알고리즘 4.4.3 Boyer-Moore 알고리즘

```
BMSearch(text, n, pat, m)
char test[], pat[];
int   n, m;
{
   int j, k, skip;
   int dd[MAX_PAT_SIZE], d[MAX_PAT_SIZE];
   initd(pat, m, d);       //next
   initdd(pat, m, dd);   //next
   k = m; skip = dd[1]+1;
  while (k <= n) {                  // Search
      j = m;
      while (j>0 && text[k] == pat[j]) {
          j--;  k--;
       }
       if ( j==0 )  {
          Report_match_at_postion(k+1);
          k += skip;         }
      else k += skip;
   }
}
```

4.4 문자열 탐색 알고리즘

4.4.3 Boyer-Moore 알고리즘의 효율

전처리 시간: $O(m+c)$

탐색 시간: 평균 $O(n\log n/m)$, 최악에는 $O(mn)$

평균 복잡도는 $O(n)$이나 KMP보다 우수

4.5 요점 정리

문서검색의 목적:

문서 집합에서 원하는 정보와 관련이 있는 문서 찾기.

문서검색의 주요 주제

질의, 검색 방법, 색인, 키워드

관련성

적절한 주제, 최근성, 신뢰성

키워드의 문제점

이음 동의어, 동음 이의어

4.5 요점 정리

IRS의 주요 문제점

문서 색인, 질의 평가, 시스템 평가

IRS의 대책

문자열처리, 색인 구축, 개념기반 검색, 지능형 처리

색인 구축 절차

토큰, 불용어, 복합 명사/동의어, 어간, 가중치, 색인 생성

Thesaurus 목적/역할

큰 문서 집합에서 관련 문서를 쉽게 찾으려고,,,

단어 변형, 어휘 통제, 검색 범위 확대. BT, NT, RT, PT, TT

4.5 요점 정리

IRS의 구성 요소

문서 표현 기능, 질의 표현 기능, 비교 함수, 색인

색인 구축 기능, 색인 검색 기능

문자열 탐색 알고리즘의 효율

Naïve 알고리즘: 야수

KMP 알고리즘: 접미사를 이용한 창문 이동

BM 알고리즘: 오른쪽부터 비교하는 창문 이동

4.6 익힘 문제

1 문제를 선택하여 자신의 주장을 기술하시오.

BF, KMP, BM 알고리즘 중에서 하나를 선택하여 실제 자료를 검색하는 프로그램을 작성하시오.