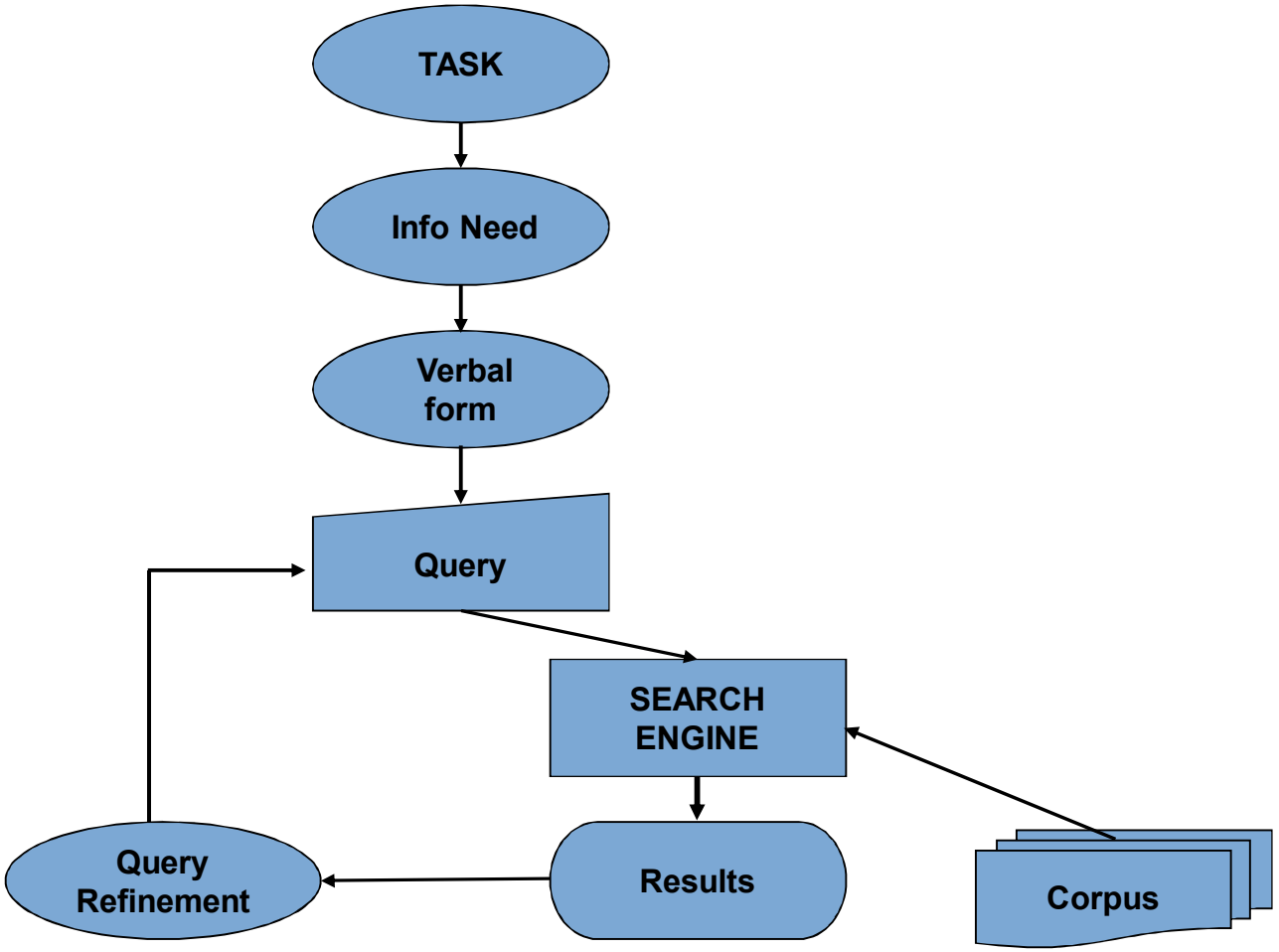
Search Engine (Basic concept)

The classic search model



Inverted index (역색인) construction

Documents to be indexed

Friends, Romans, countrymen.

Tokenizer

Token stream

Friends Romans Countrymen

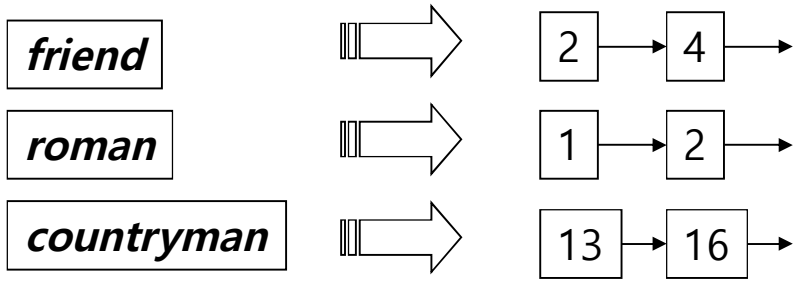
Linguistic modules

Modified tokens

friend roman countryman

Indexer

Inverted index



Indexer steps: Token sequence

- Sequence of (Modified token, Document ID) pairs.

Doc 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

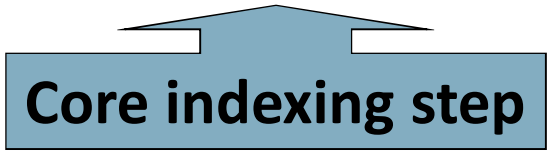
Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Indexer steps: Sort

- Sort by terms
  - And then docID



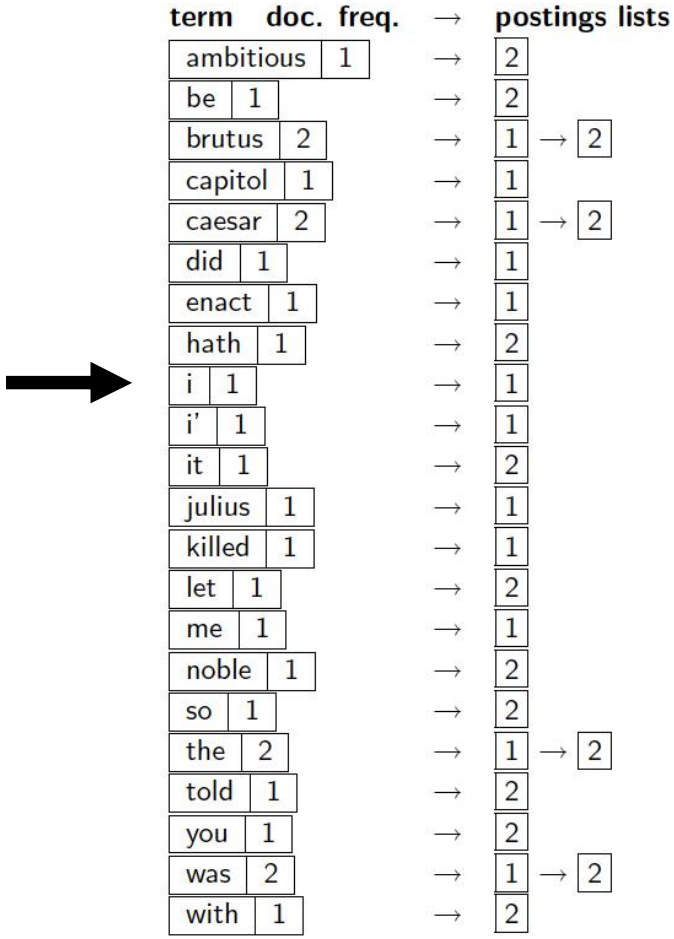
Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Indexer steps: Dictionary & Postings

- 동일 문서에서 같은 용어가 여러 번 출현하면 하나로 합침
- 사전과 포스팅으로 분리
- 사전에 문서 빈도 (Doc. Frequency)를 저장

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2



Implementation

index.c

- 문서집합을 역색인(inverted index) 구조로 변환한다.

- 입력 : 문서 집합 (r52.txt)
- 출력 : 사전 엔트리 목록 (dic.txt)
  역색인 파일 2종 (header.idx, posting.idx)

r52.txt

- Source: Reuters-21578 data set

- 한 라인에 하나의 문서가 저장되어 있음
  - 9100 라인(문서)
- Modified token 형태로 모두 변환됨
  - tokenized, 소문자로 정규화, stopword 제거, 기호 제거)

- DocId는 첫 줄부터 1, 2, 3, …으로 간주

dic.txt

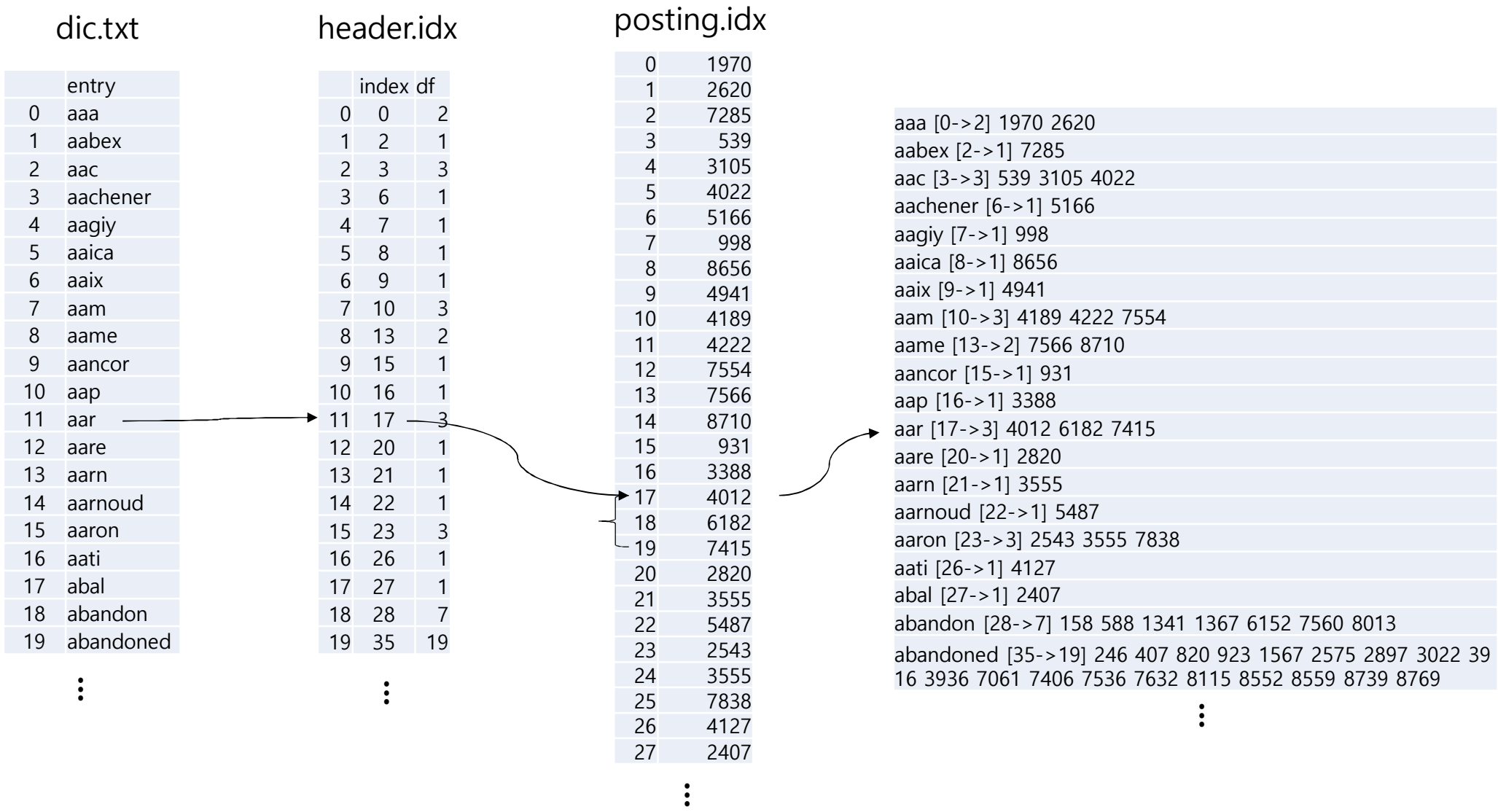
- 사전(dictionary)에 저장될 엔트리 목록
  - 25,611개
  - 정렬되어야 함
  - qsort 사용
    - 구조체 배열 정렬
    - Compare 함수 : 정렬 기준 1순위: 토큰(문자열), 2순위: DocId(정수)
- 예)
  - aaa
  - aabex
  - aac
  - aachener
  - aagiy
  - aaica
  - aaix

header.idx, posting.idx

- header.idx
  - 사전의 각 엔트리들의 시작 위치(index)와 문서빈도(doc. Freq.)를 저장
  - 구조체 배열

```
typedef struct {
        int index; // starting position in posting.idx
        int df; // document frequency
} tHEADER;
```

- posting.idx
  - 문서 포스팅 정보 저장
  - 정수(int) 배열


dic.txt
header.idx
posting.idx
entry
0 aaa
1 aabex
2 aac
3 aachener
4 aagiy
5 aaica
6 aaix
7 aam
8 aame
9 aancor
10 aap
11 aar
12 aare
13 aarn
14 aarnoud
15 aaron
16 aati
17 abal
18 abandon
19 abandoned
index df
0 0 2
1 2 1
2 3 3
3 6 1
4 7 1
5 8 1
6 9 1
7 10 3
8 13 2
9 15 1
10 16 1
11 17 3
12 20 1
13 21 1
14 22 1
15 23 3
16 26 1
17 27 1
18 28 7
19 35 19
0 1970
1 2620
2 7285
3 539
4 3105
5 4022
6 5166
7 998
8 8656
9 4941
10 4189
11 4222
12 7554
13 7566
14 8710
15 931
16 3388
17 4012
18 6182
19 7415
20 2820
21 3555
22 5487
23 2543
24 3555
25 7838
26 4127
27 2407
aaa [0->2] 1970 2620
aabex [2->1] 7285
aac [3->3] 539 3105 4022
aachener [6->1] 5166
aagiy [7->1] 998
aaica [8->1] 8656
aaix [9->1] 4941
aam [10->3] 4189 4222 7554
aame [13->2] 7566 8710
aancor [15->1] 931
aap [16->1] 3388
aar [17->3] 4012 6182 7415
aare [20->1] 2820
aarn [21->1] 3555
aarnoud [22->1] 5487
aaron [23->3] 2543 3555 7838
aati [26->1] 4127
abal [27->1] 2407
abandon [28->7] 158 588 1341 1367 6152 7560 8013
abandoned [35->19] 246 407 820 923 1567 2575 2897 3022 3916 3936 7061 7406 7536 7632 8115 8552 8559 8739 8769

search.c

- 입력된 사용자 query를 포함하는 문서를 출력하는 프로그램
  - Query
    - exact match : 문자열
    - wildcard search : *을 포함하는 문자열
    - 불린(Boolean) 연산자 : & (and), | (or)
  - 사전 엔트리 목록 → trie 저장
  - 사용자 query → trie에서 탐색 → (역색인 구조에서) 문서 검색 → 결과 출력

프로그래밍을 위한 tip

- 구조체 정렬
  - qsort(tokens, num_tokens, sizeof(tTokenDoc), _compare);
- 파일 쓰기
  - fwrite(header, sizeof(tHEADER), num, fp);
  - fwrite(posting, sizeof(int), num, fp);
- 파일 읽기
  - fread(header, sizeof(tHEADER), num, fp);
- 동적메모리 할당/재할당
  - malloc, realloc
- 파일 위치 설정 및 확인
  - fseek, ftell

프로그래밍을 위한 tip

- trie 구조체 변경
  - 엔트리 문자열 → 엔트리 인덱스
    - 초기값은 -1로 설정

```
typedef struct trieNode {
        int     index; // 0, 1, 2, ...
        struct trieNode *subtrees[MAX_DEGREE];
} TRIE;
```

- Makefile 사용
  - 복수의 소스코드 컴파일