Boyer Moore 알고리즘을 이용한 PDF 텍스트 검색기

기존 웹 스크래핑 Implementation
-> 텍스트의 길이가 짧아 가시적인 차이를 이끌어내기 어려움

PDF 문서에서 텍스트 패턴 탐색

Preprocessing

```
std::vector<int> preprocessBadCharHeuristic(const std::string& pattern) {
        int NO_OF_CHARS = 256;
        std::vector<int> badChar(NO_OF_CHARS,-1);
        for (int i = 0; i < pattern.size(); ++i)
                badChar[(int) pattern[i]] = i;
        return badChar;
}
```

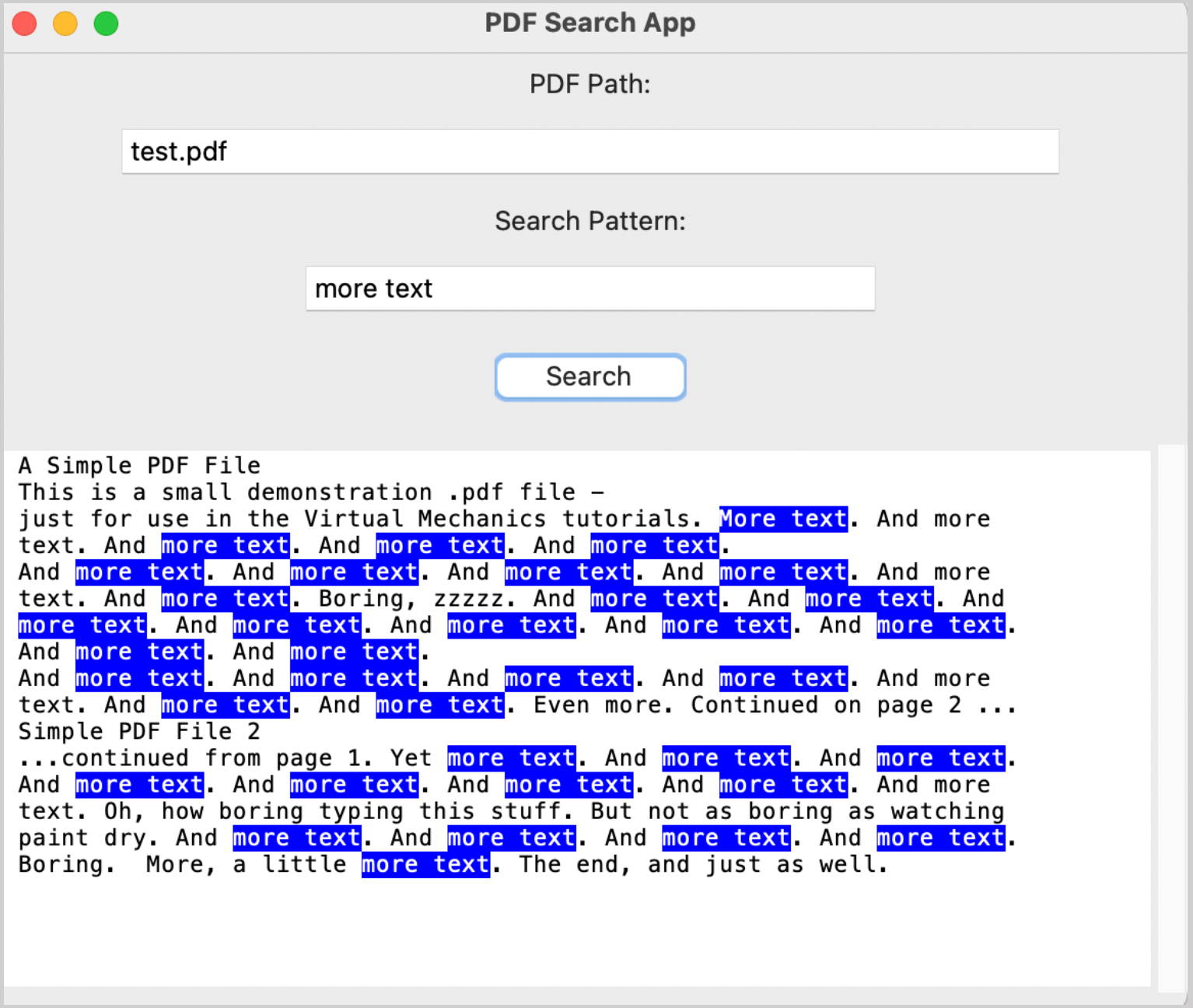
→ -1로 초기화

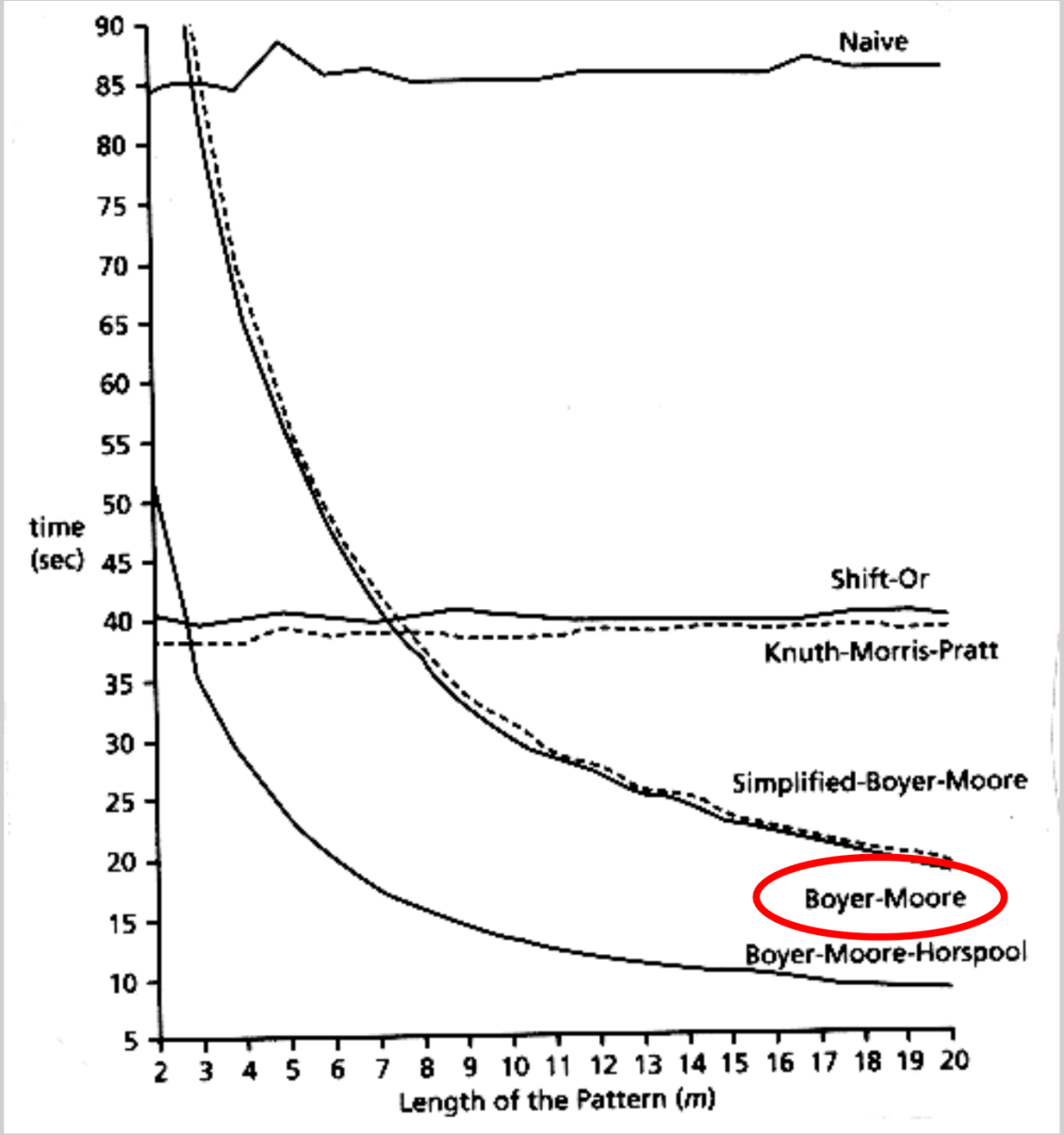
→ 알파벳 위치 정보
중복되는 문자에 대해선
가장 오른쪽에 있는 문자의
위치정보 저장

패턴 탐색 횟수와 텍스트 속 패턴의 인덱스 출력
-> 직관적이지 않음

```
Shiftcount: 153
Occurrences: [114, 146, 161, 176, 193, 208, 223, 238, 270, 300
, 630, 645, 660, 677, 692, 707, 722, 831, 846, 861, 876, 913]
```

Tkinter를 이용해 GUI 제작
일치한 패턴 하이라이트





패턴의 길이가 길어질수록
효율성 증가
-> 탐색 횟수로 효율 측정

The Project Gutenberg eBook of Pattern

Characters (with spaces) 290,328

패턴 길이 8

```
Shiftcount: 38123
```

패턴 길이 20

```
Shiftcount: 15182
```

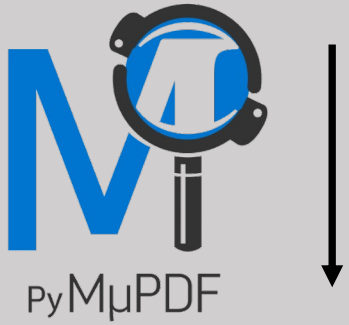
패턴 길이 40

```
Shiftcount: 7582
```

-> 패턴의 길이가 길수록 탐색 속도가 빨라짐

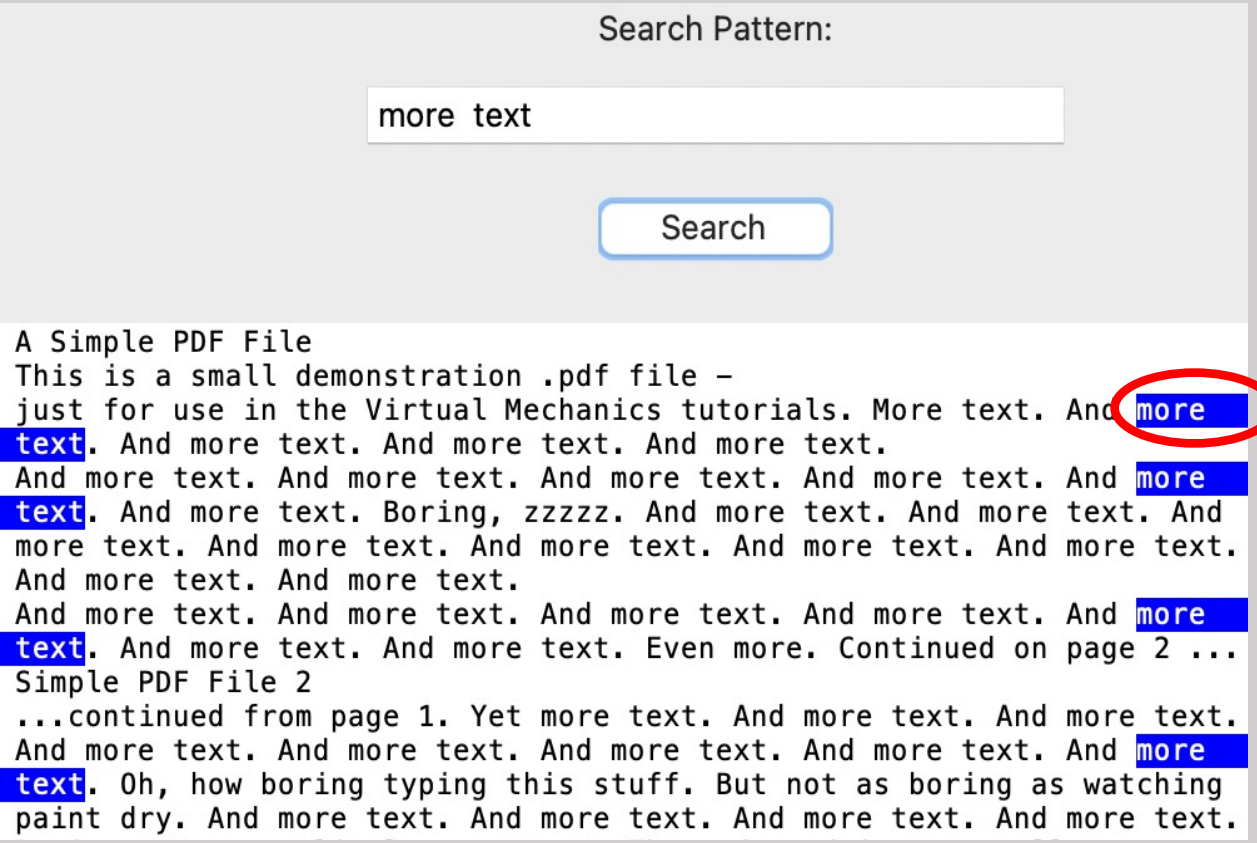
한계점 1. 부정확한 텍스트 추출

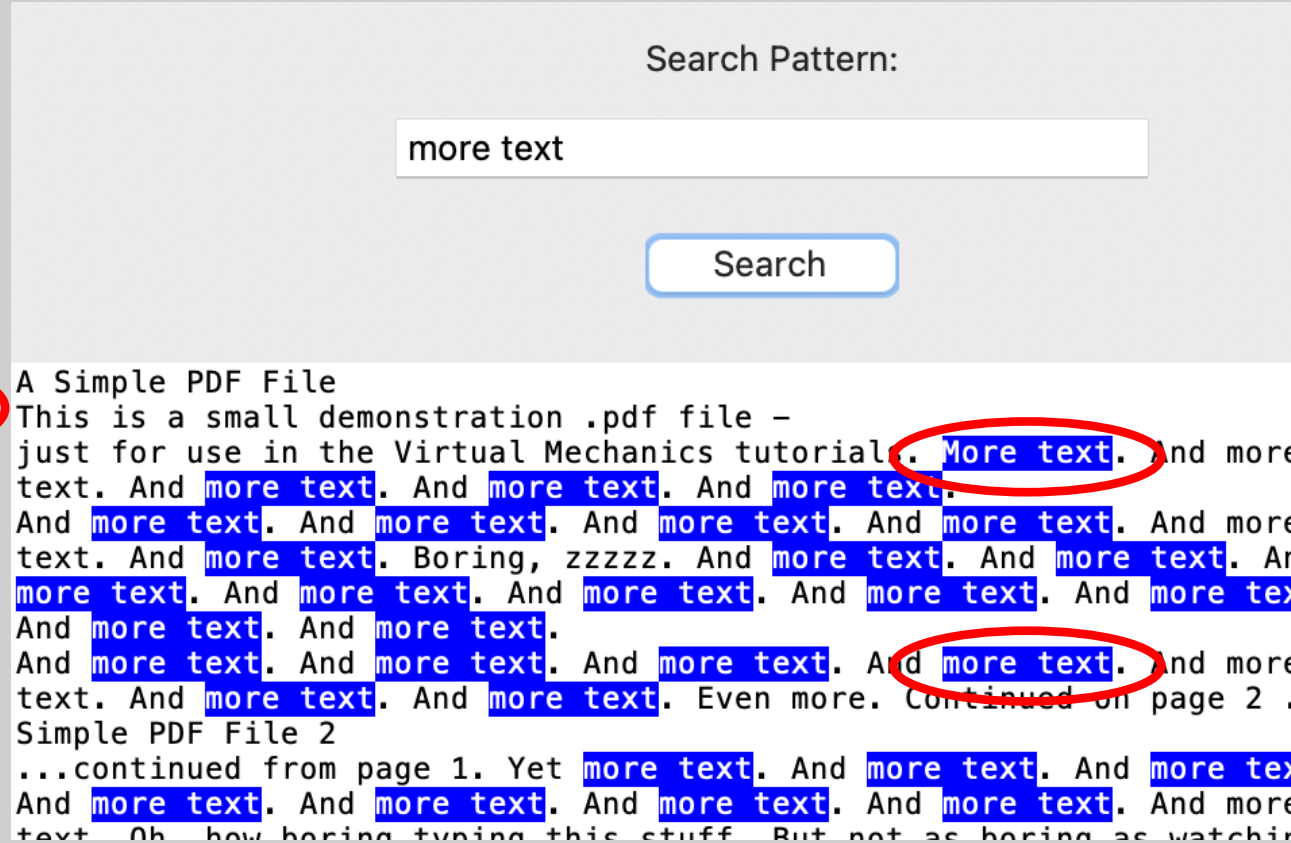
The Project Gutenberg eBook of Pattern




```
The Project Gutenberg eBook of Pa4ern
```

한계점 2. 텍스트 추출 후 개행문자 대응





감사합니다.