지문법과 서열정렬법을 결합한 다단계 정렬 방법의 문서 유사도 비교

서종규, 옥창석, 조환규
부산대학교 컴퓨터공학과
e-mail : maniasjk, csock, hgcho@pusan.ac.kr

A method for comparing documents using fingerprinting and sequence alignment.

Jongkyu Seo, Chang-Seok Ock, Hwan-Gue Cho
Dept. of Computer Engineering, Pusan National University

요 약

문서유사도를 비교하는 방법은 지문법과 서열 정렬법이 널리 알려져 있다. 지문법은 계산속도가 빠른 대신 정확도가 떨어지며, 서열정렬법은 계산속도가 느린 대신 정확도가 높다. 다단계 정렬은 두 방법의 비중을 조절하여 문서 유사도를 비교할 수 있는 방법으로, 각 방법의 장점을 얻으면서 단점을 보완하도록 고안되었다[1]. 이 논문에서는 다단계 정렬방법에 대해 설명하고, 다단계정렬 방법에서 발생 가능한 단편화 문제를 제거하여 정확도를 향상시키는 방법에 대해 소개한다.

1. 서론

태권도 국가대표 선수의 논문표절부터 대통령 후보의 논문표절 논란 등 표절에 대한 사회적 관심이 높아지고 있다. 많은 수의 문서에서 비슷한 내용의 문서를 일일 찾는 것은 엄청난 시간과 노력을 필요로 하기 때문에 자동화된 문서 비교 방법이 어려가지 제안되었다. 그 중에서 널리 사용 되는 방법으로 지문법(fingerprinting method)과 서열정렬법(sequence algorithm)이 있다. 지문법은 문서를 구성하는 '지문'을 추출 후 비교하는 방법으로 빠르게 계산이 가능한 반면, 문서의 부분적인 유사도는 알 수 없다는 단점이 있다. 서열정렬법은 두 문자열의 일치, 삭제, 삽입을 통해 유사한 구간을 찾는 방법으로, 짧은 구간의 유사도 비교에 매우 효율적이지만 $O(m \cdot n)$이라는 긴 계산시간이 필요하다.

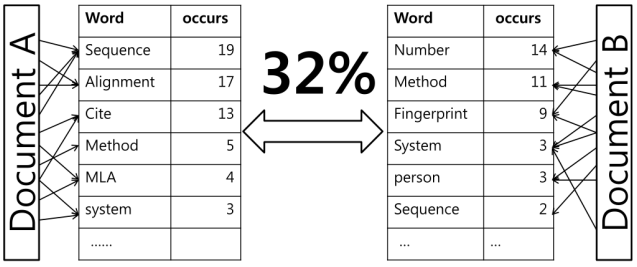
다단계정렬은 지문법과 서열정렬법을 조합한 새로운 방법으로, 사용자가 필요에 따라 지문법과 서열정렬법의 비중을 조절하여 정확도와 계산시간을 선택할 수 있도록 고안된 방법이다[1].

이 논문에서는 다단계 정렬에 대한 소개와 유사 구간의 단편화 제거를 통해 다단계정렬 방법의 성능을 향상시키는 방법에 대하여 설명하고, 실험적으로 다단계정렬방법의 유사도 검출 능력과, 단편화 제거를 통한 성능 향상을 보인다.

2. 관련연구

가) 지문법

지문법은 말 그대로 지문을 비교하여 일치 여부를 판단하는 방법이다. Rabin's algorithm[2]은 지문법을 문자열 비교에 사용한 알고리즘중 하나로, 문자열을 하나의 숫자로 나타내어 적은 수의 비교로 문자열의 일치 여부를 판단한다. 이 방법을 확장하여 문서에 등장하는 단어의 빈도를 벡터로 나타내어 문서 사이의 유사도를 계산하는 방법이 있다. (그림 1)은 단어의 출현 빈도를 이용한 지문법의 계산 방법을 나타낸다.



(그림 1) 단어 출현 빈도를 이용한 문서 유사도 비교 방법. 벡터 유사도 비교 함수를 이용하여 두 지문의 유사도를 계산한다.

지문법은 입력 문서의 길이가 m, n일 때 시간 복잡도가 $O(m+n)$으로 선형 시간에 빠르게 계산 가능하다는 장점이 있다.

나) 서열정렬방법

서열정렬방법은 DNA에서 유사 염기서열을 찾아내는데 사용되는 방법으로[3], 문자열의 일치 또는 불일치에 따라 문자열을 배치(align) 하여 최대 유사 문자열을 찾아내는 방법이다. Computer science에서는 일반적으로 동적 계획법을 이용하여 서열정렬법을 구현한다. (그림 2)는 서열정렬법을 이용한 유사 문자열 탐색 방법을 나타낸다.

INPUT	T	H	I	S	I	S	S	E	N	T	E		N	C	E	F	O	R	I	N		P	U	T
TARGET			I	T	I	S	S			T	R	I	N	G		F	O	R	C	O	M	P	A	R	E
Result				I		I	S	S		T		N		F	O	R				P				

(그림 2) 서열 정렬법은 이용한 최대 유사 문자열 탐색. 그림에서는 “THIS IS SENTENCE FOR INPUT”과 ”IT IS STRING FOR COMPARE“ 두 문장을 비교하였다.

서열 정렬법은 시간 복잡도가 $O(m \bullet n)$으로 지문법에 비해 계산 시간이 오래 걸리지만, 지문법보다 유사구간을 정확하게 찾아 준다는 장점이 있다.

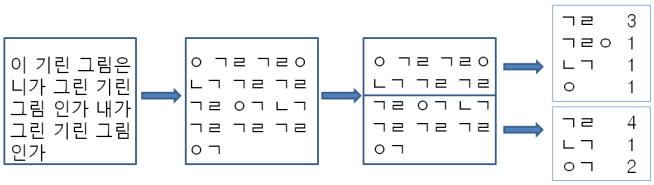
3. 다단계 정렬

앞서 설명한 유사도 비교방법중 하나만 이용할 경우 사용자는 지문법의 계산 속도와 서열정렬법의 정확도 중에서 하나 택해야 한다. 이에 반해 다단계 정렬방법은 위의 두 가지 방법을 조합하여 지문법과 서열정렬법의 비중을 사용자가 선택 할 수 있도록 고안되었다.

다단계정렬은 크게 세 단계로 나눌 수 있다.

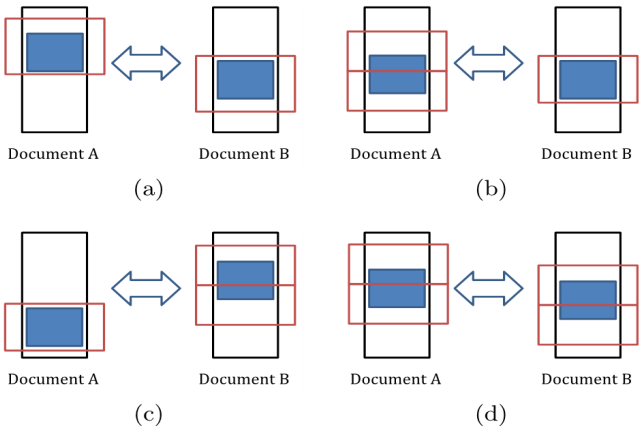
가) 전처리 과정

전처리 과정은 두 입력 문서를 일정한 크기의 로 나누고 각 블록으로부터 지문을 추출 하는 과정이다. 우선 두 입력 문서에서 한글 초성을 추출한다. 이 과정은 초성이 가지는 높은 정보량[4]과 낮은 원본 추출 가능성[5]을 이용하여, 원본을 공개하지 않고 문서 유사도를 비교 할 수 있도록 하여, 발생 가능한 저작권 문제를 방지해 준다. 그 후 두 문서는 일정한 크기의 여러 블록으로 나누어지고, 각 블록의 구성 단어를 이용하여 벡터 형태의 “지문”을 생성한다. 이때 한글단어의 90%는 3글자 이하 이며, 통계적으로 그 뒷부분은 변화가 심한 어미에 해당하므로 단어의 첫 3음절을 이용하여 벡터를 생성한다. 한글에서 (그림 3)은 전처리 과정을 간단하게 나타낸 것이다.



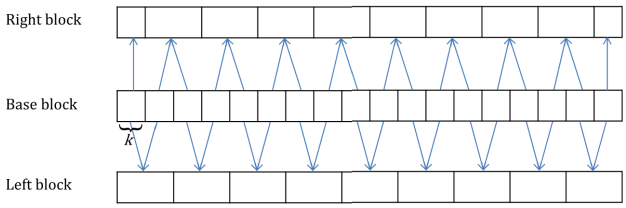
(그림 3) 전처리 과정을 나타낸 그림. 이 과정에서는 입력 문서의 초성 추출, 블록화, 지문 추출의 단계를 거치게 된다.

문서를 여러 블록으로 나누는 단계에서 실제로 유사한 구간이 단편화되는 현상이 발생할 수 있다. (그림 4)가 이러한 현상을 설명하는 것으로, 푸른 사각형은 유사구간을 나타내며 붉은 사각형은 한 블록을 나타낸다. (그림 4-a)는 유사구간이 블록내에 있어 단편화 되지 않은 상태로 가장 이상적인 결과를 내준다. 반면 (그림 4-b)와 그림(그림 4-c)는 한 문서에서 유사구간이 두 개의 블록에 나누어 졌으며, (그림 4-d)는 두 문서 모두 유사구간이 두 블록으로 나누어져 단편화로 인한 오차가 발생하게 된다.



(그림 4) 유사구간의 단편화가 발생할 수 있는 4 가지 경우. (a)는 단편화가 없지만 (b)와 (c)는 한 문서에서, (d)는 두 문서 모두 단편화가 발생하여 오차가 생길 수 있는 경우이다.

이때 한 블록 b_i에 속한 단편화 된 유사구간은 인접한 블록 b_{i-1} 또는 b_{i+1} 중의 하나로 유사구간이 이어진다는 특징이 있기 때문에, 인접한 두 블록의 지문 벡터를 합하는 방법으로 단편화 문제를 제거할 수 있다. 단편화된 유사구간은 왼쪽 또는 오른쪽의 블록으로 이어지기 때문에 각각 대한 모든 경우를 포함하기 위해 (그림 5)과 같이 두 개의 벡터 집합을 생성한다.



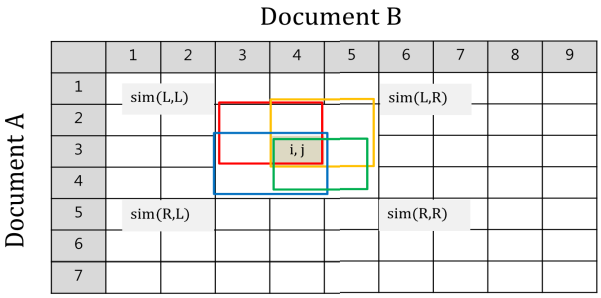
(그림 5) 단편화 제거를 위하여 인접블록을 합한 벡터를 만드는 과정

단편화된 유사구간은 반드시 두 벡터 집합중 하나에서 다른 유사구간과 결합하여 높은 유사도를 나타내기 때문에 이 방법을 이용하여 단편화 문제를 상당부분 해결할 수 있다.

나) 유사도테이블생성

전처리 과정에서 생성된 두 개의 벡터 집합을 이용하여 유사도 표를 작성한다. 문서의 크기가 m,n 이고 블록의 크기가 k라면 각 문서는 $m/k,n/k$ 개의 블록을 가진다. 유사도 표 T에서 하나의 셀 $T[i,j](0<i<m/k,\ 0<j<n/k)$ 는 첫 번째 문서의 b_i와 두 번째 문서의 b_j 사이의 유사도를 계산한 결과이다. 벡터의 유사도 계산에는 널리 알려진 cosine similarity measure를 사용한다.

이때 b_i와 b_j는 전처리 과정에서 생성한 두 개의 문서 집합에 각각 포함된다. 따라서 $T[i,j]$를 계산하기 위해서는 총 네 번의 계산이 필요하게 된다. (그림 6)은 유사도 표에서 $T[i,j]$를 계산하는데 필요한 네 번의 블록 비교를 보여준다.



(그림 6) 유사도 표에서 $T[i,j]$를 계산하는데 필요한 네 번의 블록 비교.

(그림 6)에서 L과 R은 b_i또는 b_j가 결합된 통합 벡터를 의미하며 $sim(A,B)$ 두 벡터의 유사도를 계산하는 함수이다. 만약 $T[i,j]$에서 유사구간이 단편화 되었다면, 이 네 번의 비교 중에서 반드시 한번은 단편화가 최소가 되는 경우가 발생하므로, 아래의 식과 같이 $T[i,j]$는 네 가지 경우 중 최대값을 취하게 된다.

$$T[i,j]=\max\begin{cases} sim(L_A(\lfloor i/2 \rfloor), L_B(\lfloor j/2 \rfloor)) \\ sim(L_A(\lfloor i/2 \rfloor),\ R_B(\lfloor j/2 \rfloor)) \\ sim(R_A(\lfloor i/2 \rfloor), L_B(\lfloor j/2 \rfloor)) \\ sim(R_A(\lfloor i/2 \rfloor), R_B(\lfloor j/2 \rfloor)) \end{cases}$$

다) 최대 유사구간 탐색

작성된 유사도 표에서 서열 정렬법을 이용하여 가장 유사한 구간을 탐색한다. 아래의 식을 이용하여 동적계획법으로 유사구간 표 H를 작성할 수 있다. 식에서 d는 서열정렬법에서의 gap penalty이다.

$$H[i,j]=\max\begin{cases} 0 \\ H[i-1,j-1]+T[i,j] \\ H[i-1,j]+d \\ H[i,j-1]+d \end{cases}$$

최종적으로 작성된 유사구간 표를 추적하면 최대 유사구간을 찾을 수 있다.

4. 다단계 정렬의 유사도 검출 성능 실험

가) 실험 데이터 구성

실험을 위하여 서로 다른 문서에 임의로 유사구간을 추가하여 주었다. T_o는 원본 문서를 이며 T_1에서 T_9는 T_0의 일부가 포함된 비교 대상 문서이다.

<표 2> 실험에 사용된 문서의 크기 및 유사구간

문서	크기(KB)	유사구간 개수	유사구간 크기(단어)
T_0	4000	0	0
T_1	10	1	100
T_2	20	2	100
T_3	30	3	100
T_4	40	4	100
T_5	50	5	100
T_6	100	1	200
T_7	500	2	400
T_8	1000	3	600
T_9	2000	4	800

나) 실험 방법 및 결과

앞의 과정을 살펴보면 전처리 과정에서 문서를 분할할 때의 블록 크기에 의해 지문법과 서열 정렬법의 비중이 달라지는 것을 알 수 있다. 두 방법의 비중을 달리하며 성능을 측정해 보는데, 측정 기준으로는 다음 두 방법을 이용한다.

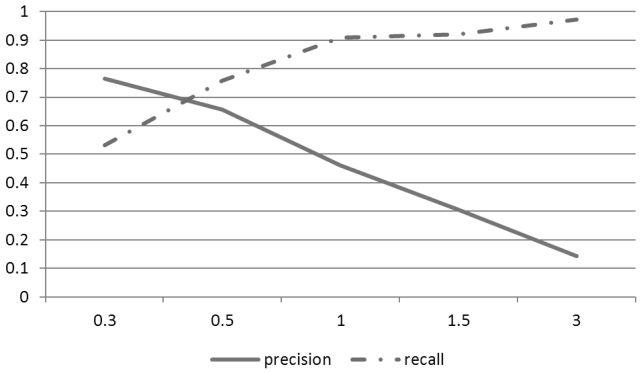
$$precision=\frac{|\{검출된유사구간\}\cap\{실제유사구간\}|}{|\{검출된유사구간\}|}$$

$$recall = \frac{|\{\text{검출된유사구간}\} \cap \{\text{실제유사구간}\}|}{|\{\text{실제유사구간}\}|}$$

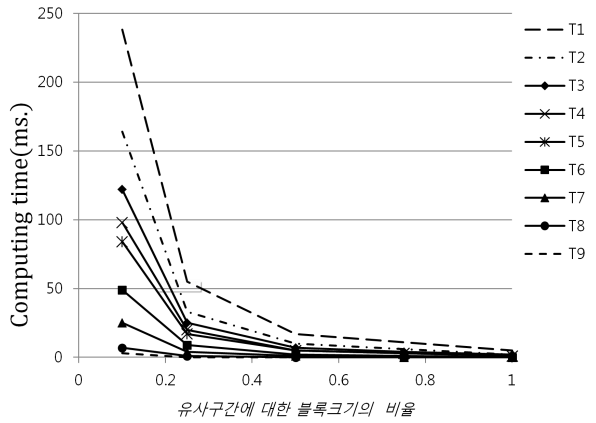
precision은 다단계정렬로 검출된 유사구간 중에서 실제로 유사한 구간의 비율을 나타내며, recall은 실로 유사한 구간 중에 다단계정렬로 검출된 구간의 비율을 나타낸다.

<표 3> 지문법과 서열정렬법의 비중에 따른 $precision(p)$과 $recall(r)$의 변화. 표에서 비율은 유사구간에 대한 블록의 크기이며, 비율이 클 수록 지문법에 가까워진다.

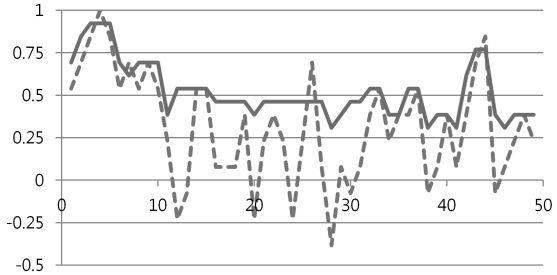
비율	0.3		0.5		1.0		1.5		3	
	p	r	p	r	p	r	p	r	p	r
T_1	0.98	0.88	0.87	0.87	0.66	0.95	0.67	1.00	0.33	1.00
T_2	0.83	0.44	0.77	0.87	0.49	0.98	0.33	1.00	0.33	1.00
T_3	0.56	0.31	0.50	0.44	0.30	0.60	0.24	0.73	0.00	NA
T_4	0.72	0.22	0.50	0.75	0.45	0.90	0.32	0.95	0.00	NA
T_5	0.62	0.21	0.40	0.46	0.16	0.96	0.00	NA	0.00	NA
T_6	1.00	0.90	0.99	0.99	0.70	1.00	0.67	1.00	0.33	1.00
T_7	0.78	0.75	0.77	0.87	0.55	0.89	0.51	0.85	0.30	0.89
T_8	0.84	0.66	0.60	0.81	0.52	0.92	0.00	NA	0.00	NA
T_9	0.56	0.41	0.52	0.77	0.32	0.97	0.00	NA	0.00	NA



(그림 7) <표 3>의 각 비율에서 모든 문서의 precision과 recall의 평균을 계산하여 나타낸 그래프



(그림 8) 비중을 달리하며 계산시간(ms) 을 측정하여 나타낸 그래프



(그림 9) 단편화 제거 전과 후의 유사도비교. 점선이 단편화 제거 전이며, 실선이 단편화 제거 후의 그래프이다

5. 결론

다단계 정렬은 블록의 크기를 조절하여 지문법과 서열정렬법의 비중을 선택할 수 있다. 지문법의 비중이 많아지면 탐색시간은 짧아지지만 precision이 감소하며, 반대로 서열 정렬법의 비중이 많아지면 탐색시간이 길어지는 대신 precision이 증가한다. 또한 이 논문에서 소개한 단편화 제거 기법을 이용하면 계산시간의 손해 없이 많은 수의 단편화를 제거하여 훨씬 정확한 유사도를 계산할 수 있다. TurnItIn[6]이나 memeChecher[7]와 같은 다른 문서 비교 시스템과는 달리, 다단계 정렬은 초성 추출을 통한 원문보호가 가능하며, 사용자의 필요에 따라 속도와 정확도 사이의 비중을 선택할 수 있다는 장점이 있다.

감사의 글

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2012-0005518)

참고문헌

[1] 박선영, 조환규, "버로우즈-휠러 변환과 다단계 정렬을 이용한 초고속 한글문서 탐색", 부산대학교 공학석사 학위논문, 2012.

[2] Michael O. Rabin "Fingerprinting by Random Polynomials" CRCT, Harvard Univ. 1981

[3] Mount DM "Bioinformatics: Sequence and Genome Analysis" Cold Spring Harbor Laboratory Press, 2004

[4] 이재홍, 오상현 "한글 음절의 초성, 중성, 종성 단위의 발생확률, 엔트로피 및 평균상호정보량" 전자공학회논문지 27(9):1299-1307

[5] 김성환, 박선영, 조환규 "한글 초성을 이용한 원문보호 탐색기법" KIPS 춘계학술대회 18(1):386-389

[6] TurnItIn http://www.turnitin.com/ , 2012

[7] MmemeChecker http://www.memechecker.com/ , 2012