김일성종합대학학보
정보과학
주체109(2020)년
제66권 제2호

조선어고속입력체계에서 TRIE나무를 리용한 합리적인 초성렬사전구축방법

방도일, 김경림, 조호건

세계적으로 매개 민족어의 형태론적 및 통계적특성에 대한 분석에 기초하여 본문편집체계에서의 고속성을 보장하기 위한 연구 및 개발사업이 진행되고있다.

선행연구[1, 2]에서는 건반재구성이나 련상기능에 의한 고속입력방법에 대하여 론의하였지만 초성렬사전구조에 대한 연구는 현재 진행되지 않았다.

론문에서는 조선어문자고속입력체계의 대규모단어정보사전구축에서 기본으로 되는 초성렬사전구조의 합리적인 설계와 구축방법을 제안하였다.

1. TRIE나무를 리용한 초성렬사전구조의 설계

초성렬위주의 조선어고속입력체계실현을 위한 사전구축에서 나서는 요구조건은 다음과 같다.

① 조선어글자를 이루는 요소들가운데서 초성만으로 열쇠모임을 구성하여야 한다.

② 매 열쇠마다에 현재경로까지의 초성렬에 해당한 후보단어들이 대응되도록 하여야 한다.

③ 매몰형장치의 자원에 영향을 주지 않도록 용량계산을 하여야 하며 그것을 위한 후보단어들의 개수를 설정하여야 한다.

조선어단어 w가 문자 S들의 모임

$$w = \{S_1, S_2, \cdots, S_n\}$$

이고 개별적인 문자

$$S_i = < L_i, V_i, [T_i] >$$

로 구성된다고 할 때

$$Z = \{L_1, L_2, \cdots, L_n\} \quad (1)$$

을 단어 w의 초성렬이라고 한다. 여기서 n은 단어 w의 길이, L_i, V_i, T_i는 문자의 초성, 중성, 종성이다.

실례로 《조선민주주의인민공화국》이라는 단어는 초성렬 《ㅈㅅㅁㅈㅈ》와 같은 형식으로 표현된다.

초성렬 《ㅈㅅ…》에 의한 단어구성을 그림 1에 보여주었다. TRIE구조의 열쇠를 초성렬형태로 표현하면 그림 1에서 보여준 실례는 열쇠모임

$$K=\{ㅈㅅㅁㅈ,\ ㅈㅅㅁㅅ,\ ㅈㅅㅎㅈ,\ ㅈㅅㅈㅇ\ \cdots\}$$

으로 구성할수 있다.

TRIE구조로 이것을 표현하면

$$K'=\{ㅈㅅㅁㅈ\#,\ ㅈㅅㅁㅅ\#,\ ㅈㅅㅎㅈ\#,\ ㅈㅅㅁ\#,\ ㅈㅅ\#\cdots\}$$

과 같이 표시된다.

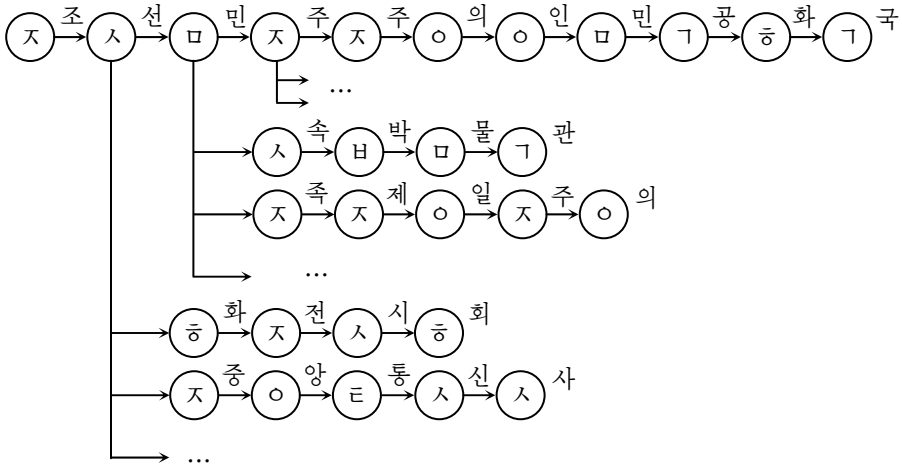

그림 1. 초성렬《ㅈㅅ…》에 의한 단어구성

초성렬열쇠모임 K'에 대한 TRIE구조를 그림 2에 보여주었다.

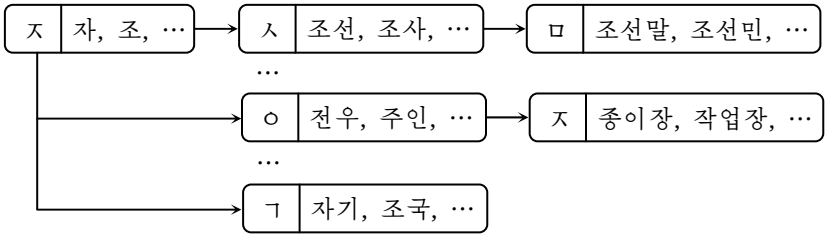

그림 2. 초성렬열쇠모임 K'에 대한 TRIE구조

그림 2에서 보여준것처럼 TRIE구조에서 열쇠와 단어사이에 이전의 1:1관계가 아니라 1:n관계가 성립한다는것을 알수 있다.

초성렬사전을 TRIE구조로 구성하려면 열쇠와 마디, 사전구조 등을 정의하여야 한다.

정의 1(초성렬사전의 열쇠)

단어 w의 초성렬을 Z, 그 단어의 길이를 n이라고 할 때

$$k = < n, Z > \tag{2}$$

를 초성렬사전의 열쇠라고 한다. 여기서 열쇠의 길이 n을 어떻게 설정하는가 하는데 따라 사전의 용량이 결정된다.

열쇠의 개수가 m일 때 일반적으로 사전의 용량 DS는 다음과 같이 계산된다.

$$DS = \sum_{i=1}^{m}(KD_i + KS_i) = \sum_{i=1}^{m} KD_i + \sum_{i=1}^{m} + KS_i \tag{3}$$

$$m = \sum_{j=1}^{n} KC_i \tag{4}$$

여기서 KD_i는 i번째 열쇠에 해당한 자료의 크기이고 KS_i는 i번째 열쇠의 크기이며 KC_i는 길이가 i인 열쇠의 개수이다. 이전의 사전들에서는 열쇠와 조선어단어가 1:1대응관계

를 가지기때문에 열쇠의 개수이자 곧 단어의 개수로 된다. 그러나 초성렬에 의한 사전에서는 단어들을 초성렬에 관하여 무리화하였으므로 열쇠의 개수가 이전의 사전들에 비하여 줄어들게 된다.

결국 초성렬사전의 용량은 그만큼 줄어들게 된다.

초성렬의 개수가 ms일 때 초성렬사전의 용량은 다음과 같다.

$$DS = \sum_{i=1}^{ms}\left(\sum_{p}^{ND_i} KD_p + KS_i\right) = \sum_{i=1}^{m} KD_i + \sum_{i=1}^{ms} KS_i \tag{5}$$

$$m = \sum_{j=1}^{ms} ND_j \tag{6}$$

여기서 $m \geq ms$이므로 초성렬의 사전은 $\sum_{i=ms}^{m} KS_i$만큼 줄어들게 된다.

초성렬사전에서 초성렬에 의하여 입력하여야 할 단어를 선택하는것이 목적이기때문에 열쇠와 목록을 결합하여 매 마디를 구성하는것으로 사전구조를 설계한다.

초성렬사전구조의 매 마디 Item은 다음과 같이 정의된다.

정의 2(초성렬사전의 마디)

$$\text{Item} = < k, \text{List}, P, C\text{Array} > \tag{7}$$

여기서 k는 열쇠이고 P는 부모마디식별자이며 CArrary는 자식마디식별자이다.

List는 열쇠 k에 따르는 단어들의 렬로서 매 단어들은 i번째로 출현가능한 단어 즉 빈도수에 따라 정렬된 단어들이다.

$$\text{List} = \{< w_i, \text{freq}_{wi} > | 1 \leq i \leq \text{count}\} \tag{8}$$

여기서 w_i는 i번째 단어이고 freq_{w_i}는 i번째 단어의 빈도수이며 count는 열쇠 k에 따르는 단어들의 개수이다. 이때 i번째 단어의 빈도수와 $i+1$번째 단어의 빈도수사이에는 반드시

$$\text{freq}_{wi} \geq \text{freq}_{w(i+1)} \tag{9}$$

과 같은 관계가 이루어져야 한다. 만일 같은 빈도수를 가지는 단어들인 경우 단어의 중요도를 계산하여 정렬한다.

대상으로 하는 문서 D_i가 속하는 분야의 코퍼스를 리용하여 어떤 단어 w_i가 D_i에서 출현하는 빈도 $t\text{Freq}(w_i)$와 그 분야에서의 평균출현빈도 $d\text{Freq}(w_i)$의 상대비률을 단어 w_i의 중요도로 표시한다. 즉

$$K(w_i) = \log\frac{t\text{Freq}(w_i)}{d\text{Freq}(w_i)} \tag{10}$$

이다.[2]

초성렬사전구조는 다음과 같이 정의된다.

정의 3(초성렬사전구조)

뿌리마디: <k, List, null, CArrary>

내부마디: <k, List, P, CArrary>

잎마디: <k, List, P, null>

$K\text{Dic} = \{\text{Item}_i\}$를 초성렬사전이라고 한다. 즉 초성렬 TRIE사전구조는 매 열쇠가 초성렬에 의해 구성되고 열쇠에 따르는 매 마디는 초성렬에 해당한 가능한 단어렬들로 구성된다. 이때 단어들은 빈도수와 중요도에 의하여 마디안에서 정렬된 구조이다.

2. 초성렬사전구조를 리용한 사전구축방법설계

조선어고속입력체계의 사전은 초성렬사전구조에 기초하며 조선어본문코퍼스를 리용하여 구축한다.

조선어본문자료기지를 *KTD*라고 하자. 입력이 *KTD*이고 출력이

$$K\,\mathrm{Dic} = <\mathrm{FindStruct}, \mathrm{WordInfo}>$$

이며 초기화조건이 $i=0, j=0, k=0, m=4$, $\mathrm{Buf}=\phi, \mathrm{Buf}\,1=\phi$, $\mathrm{Unit}=\phi$, $n=0$ 일 때 사전구축알고리듬은 다음과 같다.

걸음 1 *KTD*로부터 *i*번째 문장을 선택하여 EnterBuf에 넣는다.

걸음 2 EnterBuf의 문장을 공백단위로 분할하여 SpaceBuf에 넣는다.

n = length(SpaceBuf)

걸음 3 for $j=1; j<n$

if SpaceBuf[*j*]가 기호가 아니면

SpaceBuf[*j*]를 Word에 보관한다.

else

함수 Insert(Word)를 호출한다.

Word를 초기화한다.

end

걸음 4 $i=i+1$

걸음 5 if $i>$ length(*KTD*)이면 걸음 6으로 이행한다.

else $i=i+1$, 걸음 1로 이행한다.

걸음 6 DataBase의 매 요소에 대하여 빈도순위로 정렬을 진행한다.

걸음 7 실행을 완료한다.

함수 Insert(Word)는 단어 Word로부터 다음의 동작을 수행하는 함수이다.

$\mathrm{Word}=S_1S_2\cdots S_N$ 이라고 할 때 $S_i(i=\overline{1,N})$ 는 조선어글자이다.

걸음 1 단어 Word에서 글자 S_i 로부터 분리한 초성을 Z_i 라고 하자.

$$Z=Z_1Z_2\cdots Z_N$$

걸음 2 초성렬 *Z*로부터 열쇠 key를 구성한다.

걸음 3 *ID*=FindStruct(key)

만일 $ID=0$이면 걸음 4로, 아니면 걸음 5로 이행한다.

걸음 4 FindStruct에 새로운 열쇠 key를 삽입하고 WordInfo에 Word를 삽입한다.

걸음 5 WordInfo[*ID*]에서 Word를 탐색한다.

만일 성공이면 해당한 빈도수를 1만큼 증가시킨다.

실패하면 Word를 WordInfo[*ID*]에 삽입하고 그 빈도수를 1로 한다.

사용자가 편집을 진행할 때 초기사전이 주기억에 적재된 다음부터 사전갱신을 여러 단계를 걸쳐 진행한다.

입력문자렬을 Word라고 할 때 사전구축알고리듬은 다음과 같다.

걸음 1 DataBase를 불러들인다.

걸음 2 입력문자렬 Word를 파라메터로 하여 함수 Find(Word)를 호출하여 후보단어들을 검색한다.

후보단어가 존재하는 경우 걸음 3으로, 없는 경우 걸음 4로 이행한다.

걸음 3 검색한 후보단어들을 후보창문에 현시해준다.

걸음 4 사용자가 후보단어를 선택하였거나 단어를 입력한 다음 공백을 누르고 Insert Word를 호출하면 단어가 등록된다.

걸음 5 사용자가 편집을 계속하면 걸음 2로, 편집을 끝내면 걸음 6으로 이행한다.

걸음 6 주기억상의 사전자료를 DataBase에 추가한다. 이때 요소에 대하여 빈도순위로 정렬을 진행한다.

걸음 7 실행을 완료한다.

함수 Insert(Word)는 우에서 정의한 함수이다.

함수 Find(Word)는 입력문자렬 Word로부터 다음과 같은 동작을 수행하는 함수이다.

$\text{Word} = S_1 S_2 \cdots S_N$ 이라고 할 때 $S_i (i = \overline{1, N})$ 는 조선어글자이다.

걸음 1 입력문자렬 Word의 글자 S_i 로부터 분리한 초성을 Z_i 라고 하자.

$$Z = Z_1 Z_2 \cdots Z_N$$

걸음 2 초성렬 Z로부터 열쇠 key를 구성한다.

걸음 3 $ID = \text{FindStruct(key)}$

만일 $ID = 0$이면 실행을 끝내고 $ID > 0$이면 여기서 얻어지는 후보단어들을 입력문자렬 Word와 부합되는 후보단어들로 돌려준다.

사전구축알고리듬을 그림 3에 보여주었다.

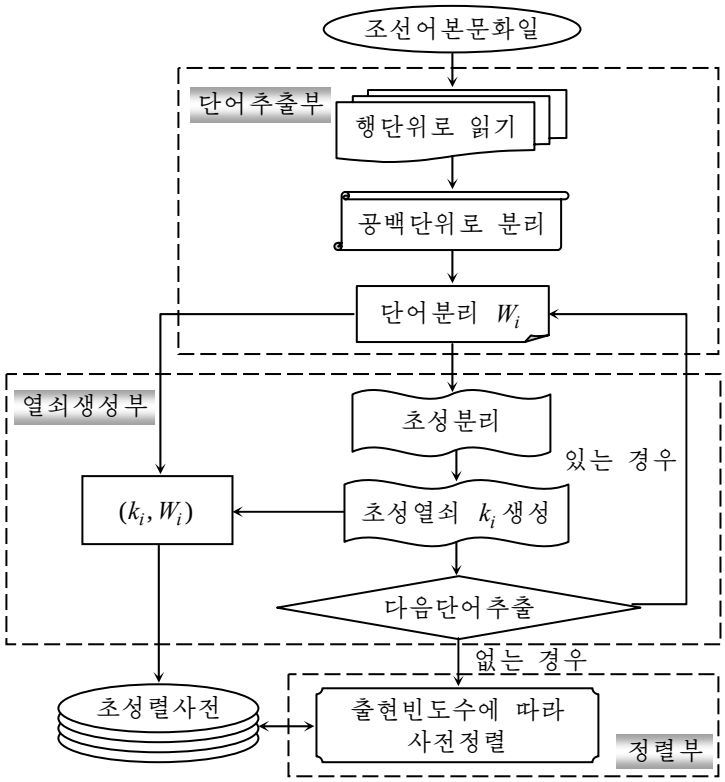

그림 3. 사전구축알고리듬

3. 성 능 평 가

조선어고속입력체계의 초성렬사전구축성능평가를 위해 다음과 같은 실험을 통하여 그 효과성을 평가하였다.

사전구축에 리용되는 본문정보를 표 1에 보여주었다.

표 1. 사전구축에 리용되는 본문정보

구 분	주 제	단어/개
사전본문 1	일상용어	1 078
사전본문 2	소설(입말체위주)	12 313
사전본문 3	종합	13 114

표 1에서 구성한 본문정보를 리용하여 초성렬사전을 구축하면 그 결과를 얻을수 있다. 사전용량의 크기분석자료를 표 2에 보여주었다.

표 2. 사전용량의 크기분석자료

사전자료	단어/개	사전용량/KB		사전용량줄임률/%
		선행방법	제안한 방법	
사전 1	1 078	116	111	95.69
사전 2	12 313	523	249	47.61
사전 3	13 114	538	255	47.39

표 2에서 보여준것처럼 론문에서 제안한 자료구조를 리용하면 10 000개이상의 단어를 가진 사전일 때 용량을 50%까지 줄일수 있다는것을 알수 있다.

맺 는 말

조선어고속입력체계에서 TRIE구조를 리용하여 합리적인 초성렬사전구조를 설계하고 사전구축방법을 실현하였으며 이전의 사전에 비해 용량을 훨씬 줄일수 있다는것을 실험을 통하여 그 효과성을 검증하였다.

참 고 문 헌

[1] Sun-Yuan Hsieh; IEEE Transactions on Computers, 61, 5, 726, 2012.

[2] Naoki Yoshinaga, Masaru Kitsuregawa; Journal of Information Processing, 20, 1, 119, 2012.

주체109(2020)년 2월 5일 원고접수

Study on the Rational Construction Method of Consonant Sequence Dictionary Using TRIE Structure in Speedup of Korean IME

Pang To Il, *Kim Kyong Rim* and *Jo Ho Gon*

In this paper, we proposed the dictionary of consonant sequence using TRIE structure in speedup of Korean IME and implemented the algorism to construct the dictionary.

Keywords: Korean input, dictionary of consonant sequence, TRIE structure