論文 76-13-5-5

한글정보처리에서 단음절의 자동식별

(Automatic Discriminating of Monosyllable in Korean Characters)

李 桂 根*·南 宮 在 贊*

(Lee, Joo K. *and* Nam Kung, Jae C.)

요 약

24개의 기본요소를 가지고 2~7개 요소로서 한 문자를 구성하는 한글 data의 연속입력으로부터 space code없이 단음절을 자동식별하는 한 system을 제안한다.

3천여자의 한글을 30종의 form으로 form화하고, 그들 form에 대한 7개의 form 특징과 문자구간을 검출하여 단음절을 식별한다. 그 결과 연속입력되는 한글 data의 처리에서 음절분리용 space code를 사용했을 때와 비교하여 컴퓨터의 기억용량이 약 25% 절감되고 처리속도가 약 30% 향상된다.

ABSTRACT

A system that can discriminate monosyllables automatically from sequential input of Korean character's data without space codes is proposed.

Korean characters are synthesized by two to seven elements out of twenty four basic elements.

Three thousands Korean characters are formalized into thirty character forms discriminates monosyllable automatically by detecting seven form features and character length.

In this result, this system, compared with the input method with space codes which have been used to separate each syllable, can save about 25% of the memory capacity of computer and improves about 30% of the processing speed of Korean characters.

1. 서 론

컴퓨터를 도구로 한 오늘날의 사회는 정보사회라고들 말한다. 정보사회의 주 목적은 우리의 주변에서 일어나는 모든 문제들을 신속하고 합리적으로 처리하여 효율의 극대화에 있다고 보겠다.

그것이 점차로 인간의 지적인 영역에까지 접근하여 간다는 것은 주지의 사실이다. 여기서 지목되는 것은 컴퓨터를 기본도구로 하는 각종 정보처리 장치의 핵심 매개체는 언어문자란 점이다.

따라서 우리는 단순한 표기요소로서의 문제에 대한 개념으로부터 눈을 돌리게 된다는 것은 필연적인 귀결이라 보겠다. 문서의 자동판독(pattern 인식), man-machine 대화에서의 Display 장치, 학습기계, 번역기계, 음성 pattern 및 Image pattern의 인식, 편집 및 자동인쇄, TV 신문 등 인간의 지적인 행동영역을 망라하는 아직은 이름만 가진 것을 포함해서 이러한 기계들은 확실히 미래를 지향하는 흥미있는 것이라 하겠다. 이러한 기운의 추세속에서 alphabet와 같이 간단하고 수가 적은 문자에 대한 자동인식 및 Display는 이미 연구가 완성되었다고(필기체는 제외) 보겠으며, 한자와 같이 복잡하고 그 수가 방대한 문자에 대해서는 많은 연구발표가 있으나(일부 실용화) 아직은 결정적인 제안은 나타나지 않은 것으로 알려져 있다. 이러한 장치들에서 첫제조건은 입력 Key의 수가 적어야 하

* 正會員 : 仁荷大學校 工科大學
Dept, of Electronic Engineers, In ha University.
接受日字 : 1976年 11月 8日

고 고속이며, 간단하고 소형의 것이라야 한다. 그러나 한자와 같은 수천종의 문제에서는 이율배반적인 문제가 된다. 한글에 있어서도 그 수가 방대하고 구조의 특의성으로 인하여 pattern인식, 고속 printer 및 Display 설계 등에서 지극히 어려운 난점을 안고 있다.

이러한 문제의 해결을 위하여 저자는 몇가지의 방식을 제안한 바 있다[2]~[3].

한글의 정보처리 과정에서 풀어쓰기로 연속 입력될 때 space code로서 단음절을 분리하는 방법과 자동식별의 두 가지 경우를 생각할 수 있다. 전자는 음절분리가 간단해지지만 정보와는 관계없는 space code를 삽입해야 하기 때문에 처리속도가 떨어진다.

후자의 경우는 처리속도는 향상되지만 음절식별이 복잡해진다. 본 연구에서는 5개의 쌍자음에 대한 공동적인 한개의 가상모음을 정의하고, 문자 form의 특징을 검출하여 일의적으로 결정지으므로써 간단한 system으로 글자의 길이가(element 수) 각각 다른 단음절을 자동 분리할 수 있다. 이것은 방대한 수의 문자를 몇개 군으로 formalize하고 또다시 그들 특징을 검출하여 그것으로서 결정짓는다는 점에서 실현의 가능성을 부여할 수 있는 이론적인 배경을 제시하는 것으로서 그 결과 컴퓨터의 기억용량을 약 25% 절약시키고 동시에 처리속도를 약 30% 향상시킬 수 있다.

마치 서구어에서 word space 없이 단어 및 일반정보를 처리하는 형태에 대응하는 것으로서 이러한 시도는 아직 발표를 볼 수 없다.

2. 문제점에 대한 고찰

일반적으로 정보처리 과정에서 문자수가 많으면 code가 길어지고 입력부분이 방대해지며 system이 대단히 비대해져서 단말장치로서 최종적인 것이 못된다. 한글은 다행히 기본문자를 가지고 있기 때문에 수천자의 문자를 24개만의 입력으로 가능하다. 그러나 구조적인 특이성으로 인해서 풀어쓰기로 입력되는 data로부터 원상태로 재현하기란 극히 어렵고 또 한 문자에 기본요소가 2~6개(7개는 현재 쓰지 않음)로서 구성되기 때문에 그 길이가 일정하지 않다. 따라서 처리과정에서 불요 정보를 문자마다 삽입하게 되므로 필연적으로 처리속도가 떨어지며, 컴퓨터 내부의 data 점유영역이 넓어지는 동시에, 기억용량의 증가를 가져오게 된다. 이 문제의 해결을 위한 간단한 system의 구성이 이 논문의 point이다.

일반적으로 풀어쓰기로 연속입력되는 한글의 정보처리 과정을 정보이론의 관점에 비추어 볼때 구조적으로는 Markov's과정을 잘 반영하는 특이한 성질이 있다. 즉 한글의 음질은 뒤따르는 모음으로서 선행하는 자이 어느 음절에 속하는가를 찾을 수 있다. (모든 외국어에서는 Markov's과정은 성립하지 않고 large number의 법측이 성립하는 ergrodic 과정이다)

앞서 저자는 한글의 구조적인 본질을 파악하기 위해서 기본자모를 두개의 기호로 표현되는 집합으로 나타내고 또 다시 그들 기호의 조합으로서 30종의 집합으로 form환 바 있다[1][2]. 즉 기본문자의 집합

$$G=\{g_i\},\ i=1,2,\cdots,24 \tag{1}$$

자음의 집합은

$$C=\{g_i|\ g\in G,\ g:\text{consenant}\}$$
$$=\{ㄱ, ㄴ, ㄷ, \cdots\cdots, ㅎ\} \tag{2}$$
$$i=1,2,3,\cdots,m\ \ m=14$$

모음의 집합은

$$V=\{g_i|\ g\in G,\ g=vowel\}$$
$$=\{ㅏ, ㅑ, ㅓ, \cdots\cdots, ㅣ\} \tag{3}$$
$$i=1,2,3,\cdots n,\ n=10$$

이라 할 때

$$C\cup V=G \tag{4}$$
$$C\cap V=\phi,$$
$$m+n=24$$

이 되어 집합 G는 Subset와 C, V로 합성 또는 분리된다. 이들 Subset의 합성으로서 모든 문자를 포함하는 조직적인 30종의 character form set를 제시하고, 그들 form으로부터 form특징을 검출하여 모든 한글을 일의적으로 결정짓는 새로운 방식의 character generator를 제안한 바 있다[1][2][4].

본 논문에서는 그들 30종의 character form set로부터 다시 system에 입력되는 flow 상태로 분해하여 그림 1에 표시하고 단음절의 분리 가능성을 검토하겠다.

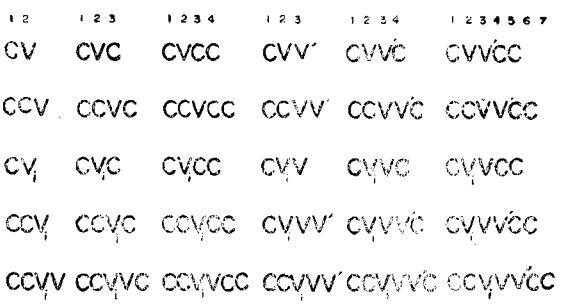

FIG. 1. The Character form.

그런데 이들 정보원에서 단음절을 한 단위로 임의로 선택하여 일렬로 연속 전달할 때 음절구간이 분리되지 않는다.

즉 그림 1의 정보원에서 "CVC" form의 정보가 반복 선택되어 "CVCCVC"로 연속 전달될 때 그것을 원문자로 재현하기 위해서는 form 구간을 분리해야 한다. 그러나, 그것이 "CV", "CCVC"로 분리되는가 "CVC", "CVC"로 분리되는가를 알 수 없다. 즉 세번째 자음 C가 앞문자에 속하는지 뒤문자에 속하는지 구별이 되지 않는다. 그것은 분리된 4개 form은 정보원에 있는 문자 form 들이기 때문이다.

또 "CVCCCVC"의 경우는 CVC, CCVC로 분리될 것인가 "CVCC", "CVC"로 분리될 것인가의 네번째 자음 C의 처리가 문제된다.

즉 위 두가지 "CVCCVC", "CVCCCVC"의 정보계열은 4개 form의 문자를 선택한 것이지만 8개 form의 문자로 분리되며 또 두 계열이 연속될 때 "CVCCVCC VCCCVC"에서는 16개 form으로 분리된다.

좀더 구체적인 예로서 [영향]이란 정보가 coding되어 철자법순으로 풀어서 system에 입력되는 상태는[ㅇㅕㅇㅎㅑㅇ]이 되어 첫번째 자음 [ㅇ]에서 세번째[ㅇ]까지 추적하면 뒤따르는 문자가 자음 [ㅎ]이므로 [ㅇ]는 철자법상 앞음절에 속한다는 것을 알 수 있다. 그러나 [덧신]의 경우 [ㄷㅓㅅㅅㅣㄴ]와 [바싹]의 경우 [ㅂㅏㅅㅅㅏㄱ]을 비교할 때 앞의 예에 따르면 [덧신]의 경우는 분리되지만 후자의 경우는 [밧삭]으로 되어 원 문자가 재현되지 않는다. 때문에 한글 TTY 등에서 음절분리용 space code를 문자마다 삽입하는 것이 보통이며, M300 TTY 등에서는 초성 쌍자음이 출현할때 space code를 넣는 등 예가 있다.

본 연구에서는 이러한 문제들이 동일한 자음 C의 반복사용에 원인된다는 점으로부터 쌍자음의 뒷자음을 가상모음 한개 parameter를 정의하여 Markov's과 정의 일반법칙을 만족시키고, form특징을 검출한다.

3. System의 구성

앞에서 언급한 30종의 character form에서 특징 Subset를 state로 하는 flow diagram은 그림 2에서 두 가지 형식으로 표현가능하다. 그림 2 (a)에서 초성 C_o, 중성은 특징별로 종횡 V_i, V_j으로 분리하고 종성을 C_k로 할때 정보의 연속 flow상태를 표현한다. 그런데 종성 C_k는 따로 존재하는 것이 아니고, 본질적으로는 같은 자음이므로 그림 2 (b)와 같이도 표현된다. 그런데 본 연구에서 거론되는 문제는 단음절의 식별에 있으므로 space S_o를 고려하면 그림 2 (c),(d)와 같이 된다.

특히 그림 2 (d)에서 기호표시 ||는 자음 C부터 모음 $V_{i,j}$를 거치지 않고는 S_o로 flow할 수 없다는 제한조건을 표시한다. 이상에서 검토된 제 연구를 기초로 하여 그것을 실현하는 system구성은 두 가지의 측면에서 검토 가능하다.

1) System 1 ; 본 방식의 주요 기능은 그림 3와 같이 Delay system, Syllable Detector, Pattern Selector 및 Basic Pattern Distributor로서 구성된다. Delay

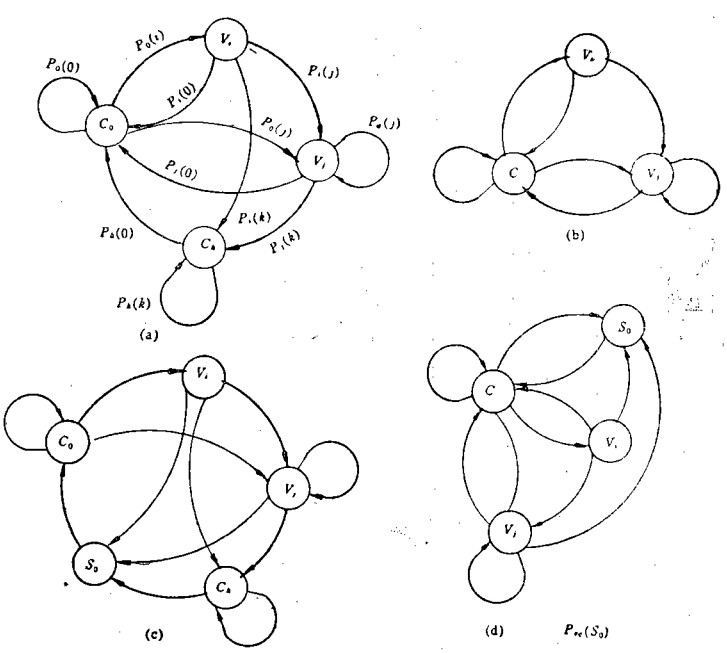

FIG 2. The flow graph of Korean Characters.

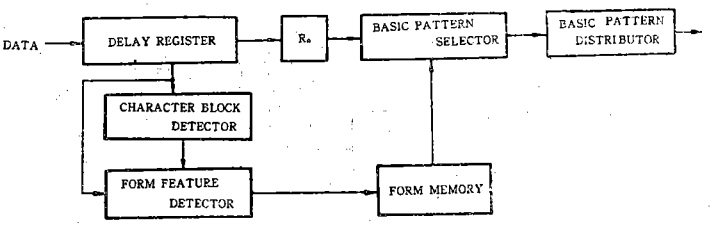

FIG 3. The System Block Diagram. (Ⅰ)

system은 7개의 Address를 가지며, syllable을 검색하는 동안 일시 지연 작용을 한다. Syllabe Detector는 음절 구간을 결정짓는 기능과 그 구간내의 문자 form의 특징을 검색하여 지금 어떤 form의 문자가 뚤어져 입력되는가를 결정짓는 기능을 가진다. 이것은 본 system의 중요 기능의 하나로서 대단히 복잡성을 가질 가능이 보이지만 대표적인 몇개의 form 특징으로서 결정짓기 때문에 간단하다. 그것을 다시 form memory에 일시 기억시켰다가 입력 data를 선택하여 Basic Pattern Distributor의 소정 Address에 기본 Pattern을 배정한다. 이때 한 문자를 형성하는 기본문자의 소정위치가 결정된다.

그런데, 연속입력 data로부터 음절의 시작부분과 접유구간을 관측하기 위하여 Parameter를 고려에 넣고 음절의 시작 부분을 지정하는 논리판정을 $N(ai)$라면

$$N(a_n)=C(a_n)\cdot V(a_{n+1})$$
$$n=1,2,3,\cdots 6 \qquad (5)$$

만약 그림 4 Delay Register의 최초의 a_1 번지에서 음절이 시작되었다면 다음 새음절은 그림 5에서와 같이 문자구성상 a_3번지에서 시작될 것이므로 $a_1, a_2, \sim a_6$의 5개 번지에서 30개 form의 form특징을 판별할 수 있다.

이때 어느 한 순간 한 음절이 선택되었다면 다른 번지의 정보는 음절검출 논리에서 부판정이 내려져야 하므로 a_n번지의 새 음절$N(a_n)$에 대응하는 논리부정 $I(a_i)$는 다음 관계로 표현할 수 있다.

$$I(a_n)=\sum_n N(a_n) \qquad (6)$$

$$3\leq n\leq 6$$

이들 논리함수로서 음절구간이 결정된다. 다음 초성 C_i, Ð parameter, 종횡모음을 V_j, V_i로 하고 $V_j'(|)$, 음절 특징을 $Y_1, Y_2, \cdots, Y_7$이라면 새음절 $N(a_1)=1$일 때

$$Y_1=C(a_1)$$
$$Y_2=D(a_2)$$
$$Y_3=\sum_{n=2}^{3} V_i(a_n)$$
$$Y_4=\sum_{n=2}^{3} V_j(a_n)+V_j(a_3)I(a_3) \qquad (7)$$
$$Y_5=\sum_{n=2}^{3} V_j'(a_n)+\sum_{n=3}^{4} I(a_n)\,V_j'(a_{n+1})$$
$$Y_6=\sum_{n=3}^{6} C(a_n)I(a_n)$$
$$Y_7=\sum_{n=3}^{5} C(a_n)\cdot C(a_{n+1})\cdot I(a_{n+1})$$

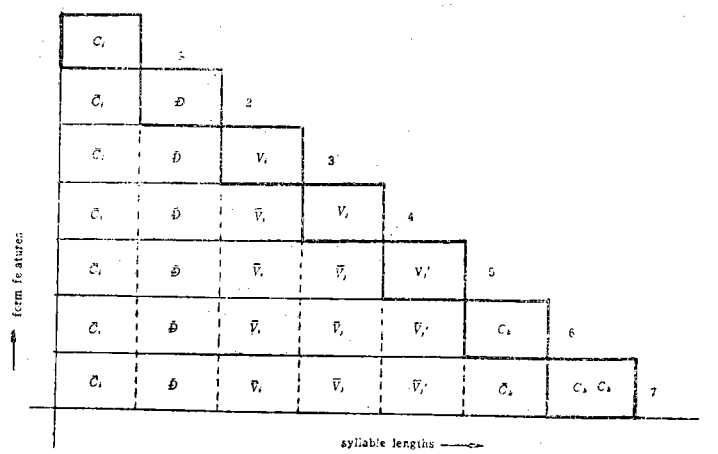

FIG 4. The Selected State of Basic Character having form feature.

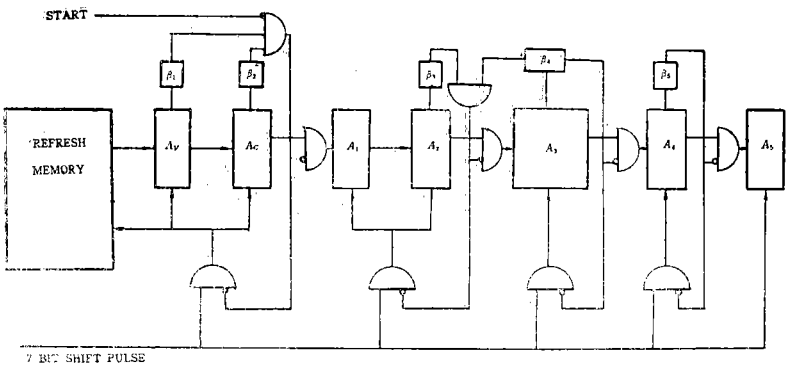

FIG 5. The System Block Diagram(Ⅱ)

(7)식으로부터 syllable Detector가 구성된다. 이들 form특징을 control 신호로 하여 입력 data를 그림 4에서와 같이 기본 pattern을 차례로 선택하여 Pattern Distributor의 고정 Address에 분배한다. 이때 고정 Address는 7개의 번지를 가지며 syllable을 한 단위로 분배한다.

이 방식은 컴퓨터의 terminal interface는 물론 teletypewriter 등에도 이용 가능하다.

2) System 2는 연속입력 data로부터 길이가 일정치 않은 단음절 구간을 분리하여 음절종류별로 일거에 처리하는 기본설계 사상은 System1에서와 같다. 그러나 이 방식은 음절의 시작부분만 식별하여 구간을 결정짓고, 각 음절 form특징의 검출은 기본 pattern Distributor의 주위에서 개별적으로 간단한 판정으로 결정짓는 것이 다르다. System이 전자보다 간단하나, 전자는 Self control인데 반해서 컴퓨가 일부 contral을 부담하는 경향이고, 처리속도가 약간 늦다. 이 System은 그림 5에서와 같이 입력 data를 두개의 Registor로서 구성되는 A_V, A_C에서 새음절의 form특징 C, V를 검출하여 data처리 진행을 Control하여 후속 data을 Address A_1, A_2에 차례로 전달한다. 이때 A_V, A_C의 내용을 B_1, B_2에서 검색하여 다음단에 정보의 전달여부를 결정한다. 이와 같이하여 입력 data는 $B_1 \sim B_5$에서 차례로 처리하여 최종 Address까지 소정위치에 분배한다. 물론 이때 음절구간에 들어있지 않은 Pattern들은 해당 위치에서 저지되며, 배당된 문자 element들은 한 문자 form을 단위로 하여 Distributor에서 검출된다. 이때 Address A_3는 $A_3 = \{V_i, V_j, V_j'\}$의 내용을 가진다.

4. 결과에 대한 총괄

본 연구는 한국어의 정보처리 과정에서 24개 기본문자에 대한 연속입력 data로부터 단음절을 자동 분리하여 모아져 나오기 위한 전처리 system을 구성하여 다음 같은 성과를 얻었다.

1) 문자마다 space code를 넣을 때에 비하여 문자당 평균기본문자의 수가 2.64[3]이므로 속도가 약 30% 향상되고, memory capacity가 약 25% 절감된다.

2) 종래의 받침등이 따로 필요없이 24개 기본문자만으로서 처리되므로 자판등 복잡한 문제들의 해결의 가능성이 있다.

3) ISO표준 code와 혼용해도 7 bit로 처리 가능하다 (모아져 나올 수 있다)

4) 풀어쓰기로 연속 입력될 때 다음절의 자동분리는 서구어의 단어분리와 흡사한데, 서구어의 단어자동분리는 아직 발표가 없다. 본 이론을 일반 정보처리에 확장된다면 주목될만 하지만 한국어에 국한된다는 것이 유감이다.

5) system 1은 teletypewriter 등에도 그대로 이용할 수 있다.

참 고 문 헌

1. Joo-Keun Lee; Recognition of Printed Korean characters, proc. ISC. oct. 1970.
2. Joo-Keun Lee; Korean Character Display by Varable combination Method Keio Engineering Reports. vol.26, No.10, 1973.
3. 李柱根, 崔興文 : 한국어의 單音節의 Entropy에 관한 연구 電子工學會誌 Vol.11, No.3, 1974—3.