자판 배열 특성을 이용한 Neuro-Fuzzy 한국어 철자 교정기의 구현*

정 한민, 이 근배, 이 종혁

포항공과대학 전자계산학과

An Implementation of Neuro-Fuzzy Korean Spelling Corrector Using Keyboard Arrangement Characteristics

Hanmin Jung, Geunbae Lee, Jong-Hyeok Lee

Computer Science & Engineering Department

Pohang Institute of Science & Technology

Tel: (0562)279-2254 Fax: (0562)279-2299

e-mail: gblee@vision.postech.ac.kr

요약

본 논문은 신경망과 퍼지 이론을 결합한 한국어 철자 교정기 KSCNN(Korean Spelling Corrector using Neural Network)에 대하여 기술한다. KSCNN은 퍼셉트론(perceptron) 학습을 이용한 연상 메모리(associative memory)로 구성되며 자판 배열 특성을 고려한 퍼지 멤버쉽 함수에 의해 신경망의 입력값을 정한다. 본 철자 교정기의 장점은 인지적인 방법으로 철자를 교정하기 때문에 기존의 VA나 BNA와는 달리 오류의 종류에 영향을 받지 않으며 교정된 철자나 후보자들에 대한 견인값(attraction value)을 측정하여 시스템의 신뢰도를 높일 수 있다는 데 있다. 또한, 본 논문은 실험을 통해서 퍼지 멤버쉽 함수에 의한 입력 노드의 활성화가 자판 배열 특성을 고려할 수 있기 때문에 시스템의 성능을 향상시킨다는 사실을 보여준다.

* 본 논문은 과학재단(KOSEF)의 연구과제 92-21-00-05 "필기체 한글을 위한 후처리 기법에 관한 연구"에 의한 것임.

I. 서론

컴퓨터의 사용이 확산되고 다루어야 할 정보량이 증대되면서 정보입력에 도움을 줄 수 있는 기술들이 필요하게 되었다. 철자 교정은 바로 그런 기술들중의 하나이며 정보의 자동 입력을 위한 광학 문자인식(OCR) 및 음성인식을 위한 후처리에 절대적으로 필요한 기술이다. 일반적으로 정보입력중에는 문서의 타이핑중에 발생하는 오류와 원문의 작성시에 발생하는 작성자의 무지에서 오는 오류등이 포함된다. 이러한 오류들을 처리하는 철자 교정기의 개발은 인간을 정보입력의 공포에서 해방시켜줄 수 있는 좋은 수단이 된다.

기존의 철자 교정 방법론으로는 지식의 확률적 표현에 기초한 상향식(bottom-up) 방법과 문맥적 지식의 구조적 표현에 기초한 하향식(top-down) 방법, 그리고 상향식과 하향식을 결합한 복합적(hybrid) 방법이 있다. 그러나, 이들 방법론에는 많은 제약들과 구현을 위한 많은 조건들이 필요하다 [4, 7]. 그리고, 무엇보다도 이들은 어떻게 인간이 단어를 기억, 재생하며 부분적인 정보로도 단어의 원래 형태를 예측할 수 있는 가능을 고려할 수 없다는 한계성을 지닌다. 이러한 기존의 방법론에서 발생하는 문제점들을 해결하기 위한 새로운 방안으로 본 논문에서는 퍼셉트론(perceptron) 학습을 적용한 연상 메모리 [1, 5]를 이용하며 타이핑중에 발생하는 오류들을 처리하기 위하여 자판 배열 특성을 고려한 퍼지 멤버쉽 함수에 의해 신경망의 입력값이 정해지는 한국어 철자 교정기 KSCNN(Korean Spelling Corrector using Neural Network)을 제안한다.

II장에서는 기존의 철자 교정 알고리즘의 개요와 문제점들을 살펴보며 III장에서는 한국어 철자 교정기 KSCNN의 구조를 알아본다. IV장은 KSCNN에서 구현한 학습 알고리즘과 퍼지 멤버쉽 함수 및 국부 최소(local minimum)의 해결 방안을, V장은 본 철자 교정기의 결과 분석과 구현 환경을 기술한다. VI장에서는 결론 및 본 철자 교정기의 개선 방향을 논의한다.

II. 기존의 철자 교정 방법론

기존의 철자 교정 방법론은 크게 상향식과 하향식으로 나눌 수 있다. 상향식 방법은 입력되는 문자열의 연속적인 확률적 가설을 개선하여 나아가는 과정으로서, 하나의 문자열 집합이 주어졌을 때 다른 한 문자가 나타날 전이 확률(transition probability)에 의한 통계적 방법을 이용하며 Markov 가설에 기초를 두고 있다. 이러한 상향식 방법의 대표적인 예가 Viterbi 알고리즘(VA)이다 [6]. VA의 방법론과 문제점을 간단히 살펴보기로 한다. 식 (1)은 1차 Markov 가설에 근거한 사후 확률(posteriori probability)을 나타낸다.

$$G1(X, Z) = \sum_{i=1}^{m+1} \log P(Xi|Zi) + \log P(Zi|Zi-1) \quad (1)$$

VA는 모든 가능한 G(X,Z)를 계산하지 않고 G(X,Z)를 최대로 하는 단어 Z를 발견하는 데 관심을 두고 있다. VA의 가장 큰 문제점은 얼마나 신뢰성있는 P(X|Z)와 전이 확률을 얻을 수 있느냐는 데에 있다. P(X|Z)는 실제로 옳은 단어인 Z가 X로 관찰될 확률을 가리키며 이를 얻기 위해서 많은 양의 데이타를 시스템에 넣고 수행시켜 보아야만 한다. 시스템의 특성과 데이타의 분포 상황에 따라 신뢰성이 크게 좌우된다는 문제점이 있다.

하향식 방법은 입력 문자열이 나타날 기대값에 의해 문자를 결정하며 사전, 구문 구조, 의미의 명시적 또는 묵시적 표현을 이용한다. 대표적인 하향식 알고리즘인 binary n-gram 알고리즘(BNA)에 대해서 살펴보기로 한다.

$$B^{m}_{ij}(\alpha,\beta) \begin{cases} = 1 & \text{if character } \alpha, \beta \text{ occur in position i, j of some words in Dm} \\ = 0 & \text{otherwise} \end{cases} \qquad (2)$$

식 (2)는 단어의 길이가 m으로 구성된 종속 사전 Dm을 위한 위치를 지정한 (positional) binary di-gram을 나타낸다. 단어 X = X1 X2 X3 ... Xm에 대해 모든 위치를 지정한 binary di-gram을 논리적 곱셈(logical product)하여 그 결과가 1이 되지 않으면 잘못된 철자를 가진 단어로 판단한다. Binary n-gram 알고리즘은 사전의 크기가 커질수록 오류의 발견과 교정에 대한 처리 능력이 떨어지며 오류가 두 군데 이상에서 발생하면 처리하는 데에 많은 어려움이 있다는 문제점들이 있다.

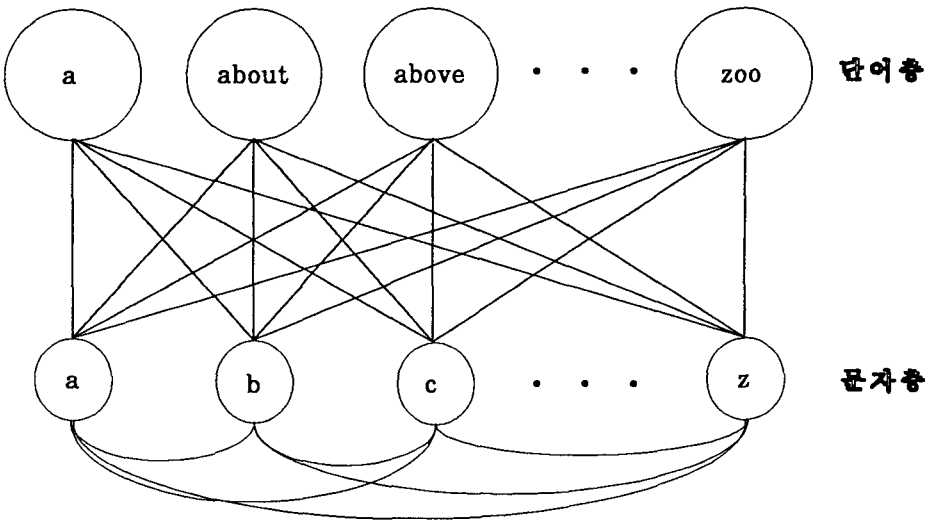

그림 1. 영어 단어를 위한 연상 메모리 모델

이러한 기존 방법론의 문제점들을 해결하기 위해 새로운 연상 메모리 모델이 [2, 3]에서 제안되었다. 연상 메모리의 개념은 매우 직관적이며 인간의 뇌에서 수행되는 기본적인 기능의 하나로서 보여진다. 이런 기능을 이용하여 우리는 친구의 얼굴과 이름을, 이름과 전화번호를 짝을 지어 쉽게 연상할 수 있다. 그림 1은 영어 단어를 위한 연상 메모리 모델을 보여준다.

서로 완전히 연결된 문자층은 영어의 26 문자들에 대응되며 기억될 단어들은 단어층에 있는 노드들에 대응된다. 단어층에 있는 두 단어들 사이에는 어떤 연결 관계도 없다. 이 연상 메모리는 저장할 수 있는 샘플들의 수에 있어서 Hopfield 메모리보다 제한을 덜 받으며 누화(crosstalk)에 대한 위험 부담이 더 적다는 장점을 가진다. 또한, 이 모델은 비샘플 견인(non-sample attraction)을 방지할 수 있다는 장점을 가진다. 그러나, 입력 패턴이 이진수로 표현되므로 후처리 시스템으로 구현하고자 할 때에 오류 발생 단어와 올바른 단어사이의 상관 관계를 이용하지 못하며 하나의 음절을 형성하기 위해서 여러 자모들을 이용하는 한국어에 적용하기 위한 보다 정교화된 언어 모델로서는 부적합하다. 이에 본 논문에서는 위의 문제점들을 해결하고자 퍼지 멤버쉽 함수와 변형된 연상 메모리를 이용한 개선된 구조의 한국어 철자 교정기 KSCNN을 제안한다.

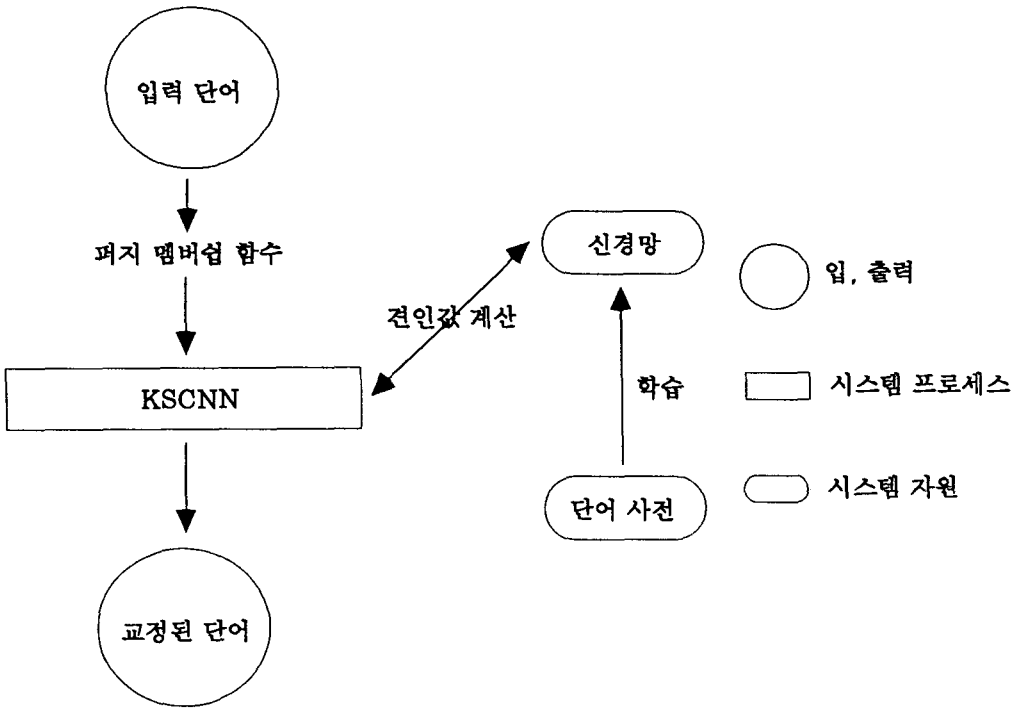

그림 2. 한국어 철자 교정 시스템 KSCNN의 구조

III. 한국어 철자 교정기 KSCNN의 구조

그림 2는 한국어 철자 교정기 KSCNN의 구조를 보여준다. 한국어 단어가 입력될 때 (이 때의 단어는 오류를 가지고 있을 수도 있으며 없을 수도 있다), 퍼지 멤버쉽 함수는 입력된 자소 각각에 대해 자음(또는 모음)간의 자판상의 유사도를 측정하여 신경망의 자소층(그림 4)에 확률적 분포로서 입력 자소들의 활성화 정도를 결정하는 역할을 한다. 즉, 퍼지 멤버쉽 함수는 기존의 일반적인 신경망과는 달리 활성화되지 않은 노드에도 멤버쉽 등급 (membership grade)을 부여하여 인위적인 활성화를 유도함으로서 오류 단어와 올바른 단어사이에의 상관 관계를 신경망에 전달하여 오류를 가진 입력 단어의 올바른 형태를 찾는 데 도움을 준다.

본 철자 교정기는 미리 학습된 신경망에 퍼지 멤버쉽 함수에 의해 결정된 입력값을 부여한 후 단어층의 모든 노드들에 대해서 견인값을 구한다. 만일 가장 큰 견인값을 갖는 단어의 견인값이 시스템이 올바른 단어라고 확신할 수 있는 임계값(threshold)을 넘는다면 교정된 형태가 출력되며 그렇지 못한 경우에는 사용자에게 올바른 후보 단어들의 목록이 제시된다.

IV. 퍼지 멤버쉽 함수와 신경망 알고리즘

기존의 신경망 구조들은 주로 입력층의 입력 값으로서 이진수를 사용하였다. 그러나, 이러한 이진수는 활성화된 노드와 그렇지 않은 노드를 양분시켜서 오류 입력이 많은 교정기의 입력값으로 사용하기에 부적합하다.

이에 본 철자 교정기에서는 퍼지 멤버쉽 함수를 이용하여 입력층의 모든 노드들을 적당하게 활성화시켜 서로간의 경쟁을 통한 올바른 단어로의 견인 가능성을 높인다. 본 논문에서 정의한 퍼지 멤버쉽 함수는 식 (3)과 같이 표현된다.

〔자음(또는 모음)간의 유사도를 위한 멤버쉽 함수〕

$$f(k) = 1 / (1 + x)^{2} \quad (3)$$

(x는 자음(또는 모음)간의 거리, k는 입력자음)
입력이 자판상에서 같은 키를 사용하는 두 자음(또는 모음)중 하나이며 shift 키를 사용하지 않는 자음(또는 모음)에 해당한다면 $x = 1$,
그렇지 않다면 $x = (h + 2 * v)$ (h는 횡적 거리, v는 종적 거리)

f(k)는 입력자음(또는 입력모음)k와 다른 자음(또는 모음)간의 유사도를 나타내는 퍼지 멤버쉽 함수이며 h는 두 자음(또는 모음)간의 횡적 거리를, v는 종적 거리를 나타낸다. 그림 3은 이 함수에 의해 계산된 자음간의 유사도 행렬(본 철자 교정기는 자음과 모음을 위한 두 종류의 유사도 행렬을 가지고 있다)을 보여준다. 본 철자 교정기에서는 중자음등과 같이 자판상에서 하나의 키에 두 개의 자음(또는 모음)이 존재하는 경우에 가상의 종적 거리를 두고 있다. 위에서 정의된 퍼지 멤버쉽 함수는 키보드 입력시 오류가 발생하는 경우에 두 자음

(또는 모음)간의 횡적 거리가 종적 거리보다 오류 발생에 보다 많은 영향을 미친다는 가설에 근거한다. 본 철자 교정기는 N-바이트 코드 체계를 사용하여 철자 교정을 수행하는 데, 이 코드는 자판 배열 특성을 고려하기가 용이하며 신경망의 입력 노드로 사용하기 쉽다는 장점을 가지고 있다.

입력 자음 / 가설 자음	ㄱ	ㄴ	ㄷ	ㄹ	· · ·	ㄲ
ㄱ	1	1/17	1/2	1/5	· · ·	*1/5*
ㄴ	1/17	1	1/10	1/5	· · ·	1/37
ㄷ	1/2	1/10	1	1/10	· · ·	1/10
ㄹ	1/5	1/5	1/10	1	· · ·	1/17
· ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·
ㄲ	*1/2*	1/37	1/10	1/17	· · ·	1
· ·	· · · · · · · · · · · · · · · ·					· · ·

〈자음을 위한 유사도 행렬〉

그림 3. 퍼지 멤버쉽 함수를 이용한 자판상에서의 자음간의 유사도 행렬
(유사도 행렬은 식 (3)에 의해서 비대칭적인 형태(ㄱ과 ㄲ의 경우)로 나타난다.)

그림 4는 KSCNN의 신경망 구조를 보여준다. 철자 교정을 위한 신경망은 변형된 연상 메모리를, 학습을 위해서는 퍼셉트론(perceptron) 학습을 이용한다.

본 철자 교정기의 신경망은 단어층과 자소층으로 구성된다. 단어층은 사전에 등록되어 있는 단어(현재는 명사만을 다루고 있다)의 갯수만큼 노드를 가지며 자소층에는 초성, 중성 및 종성을 위한 61개의 노드가 미리 정의된 단어의 최대 음절수(현재 본 철자 교정기에서는 이 값을 5로 정의하고 있다) 만큼 반복해서 존재한다.

사전의 단어들에 대한 학습을 수행할 때에 다음의 두 가지 요소가 계속적으로 변하면서 학습이 이루어진다. s번째 상태에서 단어 노드의 활성화 정도를 나타내는 X(s)와 두 층간의 웨이트인 W1과 자소층간의 웨이트인 W2가 상호 작용을 하면서 모든 사전 단어에 대해서 더이상의 웨이트 변화가 없을 때까지 학습을 계속한다. 웨이트 W1kj(s)는 시스템의 s번째 상

태에서 자소층의 k번째 노드와 단어층의 j번째 노드사이의, 웨이트 W2kl(s)는 자소층의 k번째 노드와 l번째 노드사이의 연관성의 정도를 나타내며 Njk는 단어층에서 j번째 단어의 k번째 자소의 값을 의미한다. 본 철자 교정기에서 사용하는 단어 노드의 활성화 값을 구하는 식은 다음과 같다.

$$Xj(s+1) = f[(\sum_{\substack{k \\ (j\langle\rangle k)}} W1kj(s)*Njk)*(\sum\sum_{kl} W2kl(s)*Njk)] \qquad (4)$$

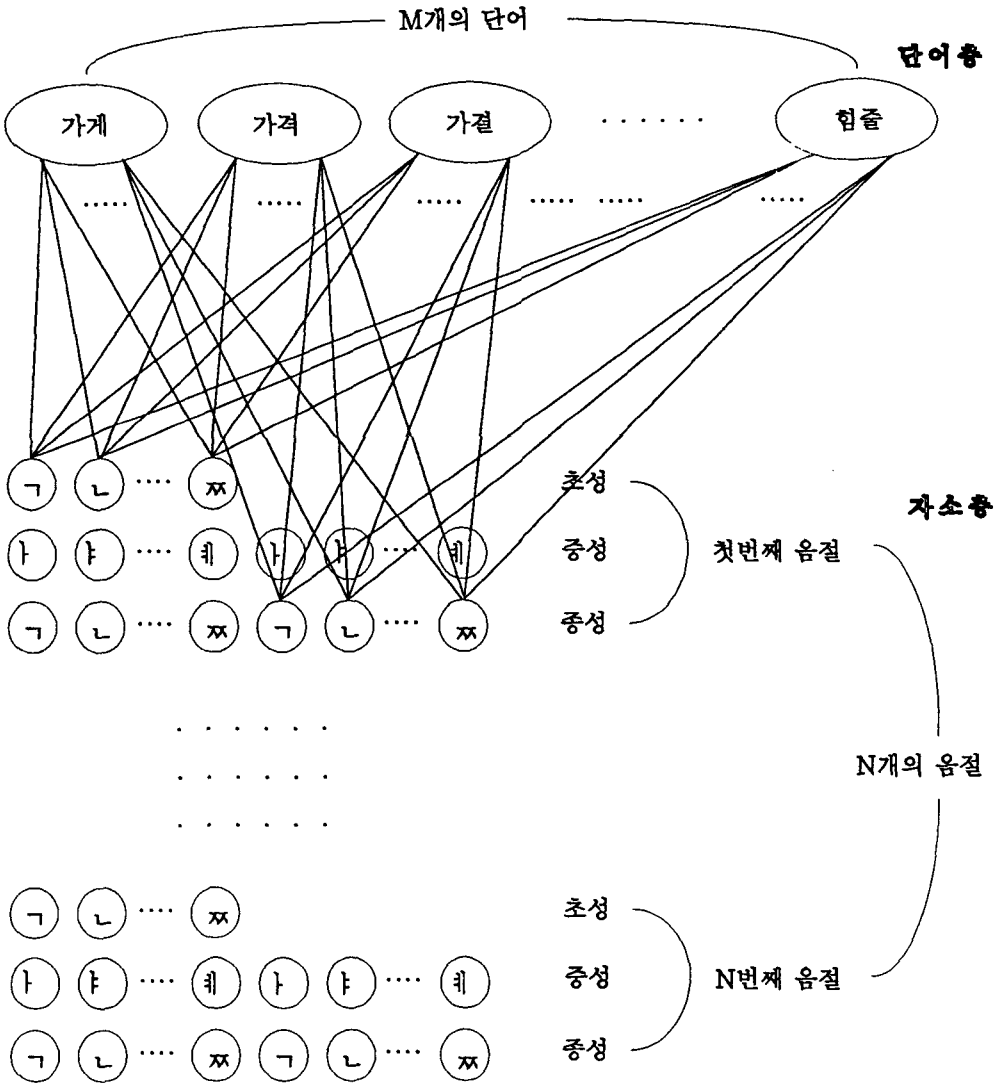

그림 4. 한국어 철자 교정 시스템 KSCNN의 신경망 구조
(M : 사전에 저장되어 있는 총 단어수, N : 단어의 최대 음절수)

Xj(s)는 j번째 단어의 s번째 상태에서의 활성화 정도를 나타내는 값이며 f(x)는 단계 함수로서 식 (5)와 같이 정의된다.

$$f(x) = \begin{cases} 1 & x > T \\ 0 & x <= T \end{cases} \qquad (5)$$

여기에서 T는 미리 정의된 임계값을 나타낸다. 만일 모든 j에 대해서 활성화 정도를 나타내는 Xj의 값이 1일 될 경우 학습이 끝나며 그렇지 않은 경우에는 식 (6), (7)에 의해서 웨이트 W1과 W2가 갱신된다.

$$W1kj(s+1) = W1kj(s) + D1*Njk \qquad (6)$$

$$\text{모든 l에 대해서 } W2kl(s+1) = W2kl(s) + D2*Njk \qquad (7)$$

식 (6), (7)의 D1과 D2(본 철자 교정기는 D1을 0.005로, D2를 0.05로 정의한다)는 학습 속도를 결정한다. 입력이 주어질 때에 웨이트 W2는 입력 자소들에 작용하여 샘플 견인자를 결정하며 웨이트 W1은 학습시에 시스템의 상태 변화에 영향을 준다.

입력 단어가 들어오면 단어층의 모든 노드에 대해서 견인값을 계산하여 가장 큰 견인값을 갖는 단어를 선택한다. 단어층의 j번째 단어에 대한 견인값은 식 (8)과 같이 계산된다. Ik는 입력 단어의 k번째 자소의 멤버쉽 값(그림 3)이며 만일 견인값의 계산후에 선택된 단어가 시스템에서 정의된 임계값을 넘지 못할 경우에는 가장 큰 견인값의 80%내의 견인값을 갖는 단어들을 후보 단어로서 출력한다.

$$\left(\sum_{\substack{k \\ (j<>k)}} W1kj*Ik \right) * \left(\sum\sum_{kl} W2kl*Ik \right) \qquad (8)$$

V. 결과 분석 및 시스템 구현 환경

표 1은 사전의 음절수에 따른 시스템의 학습 결과를 보여 준다. 학습율은 단어의 음절수가 길수록 높아진다. 이 실험으로부터 단어의 길이가 길수록 그에 따른 부가적인 정보들을 신경망이 학습하는 데 보다 많이 이용하고 있다는 사실을 확인할 수 있다.

표 2는 2음절과 3음절 오류 단어에 대한 본 철자 교정기의 교정 결과를 보여준다. 신문의 사회면에서 추출된 단어(현재 약 1000개의 사전 단어를 가지고 있다)들을 실험자가 실제로 타이핑하는 과정에서 발생된 오류 단어들을 모아 실험에 사용하였다. 오류 단어는 1개에서 3개까지의 오류를 포함하고 있었으며 두 종류 이상의 오류를 가진 단어들도 발견할 수 있었다. 2음절 오류 단어의 경우에 단어당 약 1.16개의 오류가 발생하였으며 3음절 오류 단어에서는 평균 1.2개의 오류가 발생하였다. 가장 큰 견인값이 임계값을 넘지 못할 경우에 후보

단어들의 목록이 제시되는 데 후보 단어들의 목록을 제시할 경우의 후보 단어의 갯수를 가장 큰 견인값의 80%이상이 되는 견인값을 갖는 단어들의 수로 정의한다. 본 실험은 2음절 단어의 경우에 77.5%, 3음절 단어의 경우에 100%의 교정율을 보여준다. 또한, 위의 실험 결과로부터 본 철자 교정기가 오류 단어에 대해서 100%의 올바른 오류 교정 또는 후보 제시를 하고 있음을 알 수 있다. 표 3은 퍼지 멤버쉽 함수의 도입이 시스템의 성능을 전반적으로 향상시킨다는 사실을 보여준다. 퍼지 멤버쉽 함수의 사용은 시스템이 오류 철자를 교정할 때에 자판상에서 근접한 자모들을 우선적으로 고려하게 하는 효과를 낳는다. 표 3에서 보듯이 퍼지 멤버쉽 함수를 사용했을 때에는 사용하지 않았을 때보다 후보 제시 여부에 관계없이 교정율이 향상됨을 알 수 있다.

	사전의 단어 길이(단어당 음절수)				
	1	2	3	4	5
학습율	15/16 (93.8%)	512/521 (98.3%)	270/272 (99.3%)	125/125 (100%)	33/33 (100%)

표 1. 사전의 단어 길이에 따른 학습율

실험 결과 / 실험 단어	오류 단어 수	대치 오류	제거 오류	삽입 오류	교정	후보 제시
2음절 단어	80	84 (90.3%)	5 (5.4%)	4 (4.3%)	62 (77.5%)	18 (22.5%)
3음절 단어	60	65 (90.3%)	4 (5.5%)	3 (4.2%)	60 (100%)	0 (0%)

표 2. 오류 단어에 대한 실험 결과
(교정은 후보 제시없이 완전하게 오류 단어가 교정된 경우를 의미하며 후보 제시는 시스템에서 정의된 임계값을 어느 후보도 넘지 못할 경우에 일어난다.
특히, 본 시스템은 후보 제시를 할 경우에 100%의 올바른 오류 교정 또는 후보 제시의 결과를 보인다.)

이상의 실험들을 통해서 본 철자 교정기가 높은 교정율을 보이고 있으며 심한 오류를 가진 단어에 대해서도 최소한 올바른 후보자들을 제시할 수 있는 능력이 있다는 것을 확인할 수 있다. 특히, 본 철자 교정기가 실험에서 사용한 모든 단어들에 대해서 완전한 오류 교정 또는 후보 제시를 한다는 점은 교정기의 신뢰도 측면에서 상당히 중요한 의미를 갖는다. 또한, 퍼지의 도입이 전반적인 시스템의 성능을 향상시키는 데 도움을 준다는 사실도 알 수 있다.

후보 제시 여부 / 퍼지 함수 적용 여부	후보 제시가 있을 경우의 교정율	후보 제시가 없을 경우의 교정율
퍼지 함수를 적용할 때	122/140 (87.1%)	135/140 (96.4%)
퍼지 함수를 적용하지 않을 때	121/140 (86.4%)	128/140 (91.4%)

표 3. 퍼지 함수의 적용 여부에 따른 실험 결과
(후보 제시가 있을 경우란 어느 후보 단어도 임계값을 넘지 못할 때 후보 단어의 목록을 제시하는 경우를 의미하며 후보 제시가 없을 경우란 임계값을 넘지 못해도 가장 큰 견인값을 갖는 후보 단어를 출력하는 경우를 의미한다.)

본 철자 교정기는 SPARC W/S상에서 Motif를 사용자 인터페이스로 하여 구현되었다. 그림 5는 본 철자 교정기의 실제적인 실행 예를 보여준다.

VI. 결론 및 앞으로의 방향

KSCNN은 오류를 가지고 있는 한국어 단어들에 대해서 실시간에 철자 교정을 수행하기 위해 구현된 한국어 철자 교정기로서 인간의 단어 학습과 연상을 모방하여 구현된 언어 모델이다. 또한, 이진입력에 따른 학습을 저하를 막기 위해 퍼지 멤버쉽 함수를 이용하여 가능성 있는 모든 입력 노드들을 활성화시키는 방법을 사용한다. 본 철자 교정기는 VA나 BNA같은 기존의 방법론에 비해 오류 갯수에 보다 적게 영향을 받으며 사전의 구성시에 특별한 자료 구조가 필요하지 않다는 장점을 가진다. 또한, 신뢰성있는 오류 교정을 수행하며 사전의 단어 증가에 따른 시스템 수행 속도의 저하가 비교적 적게 일어난다. 본 철자 교정기는 단어 갯수만큼의 신경망 노드가 필요하다는 단점은 있지만 자연어 명령 및 프로그래밍 오류 교정같은 제한

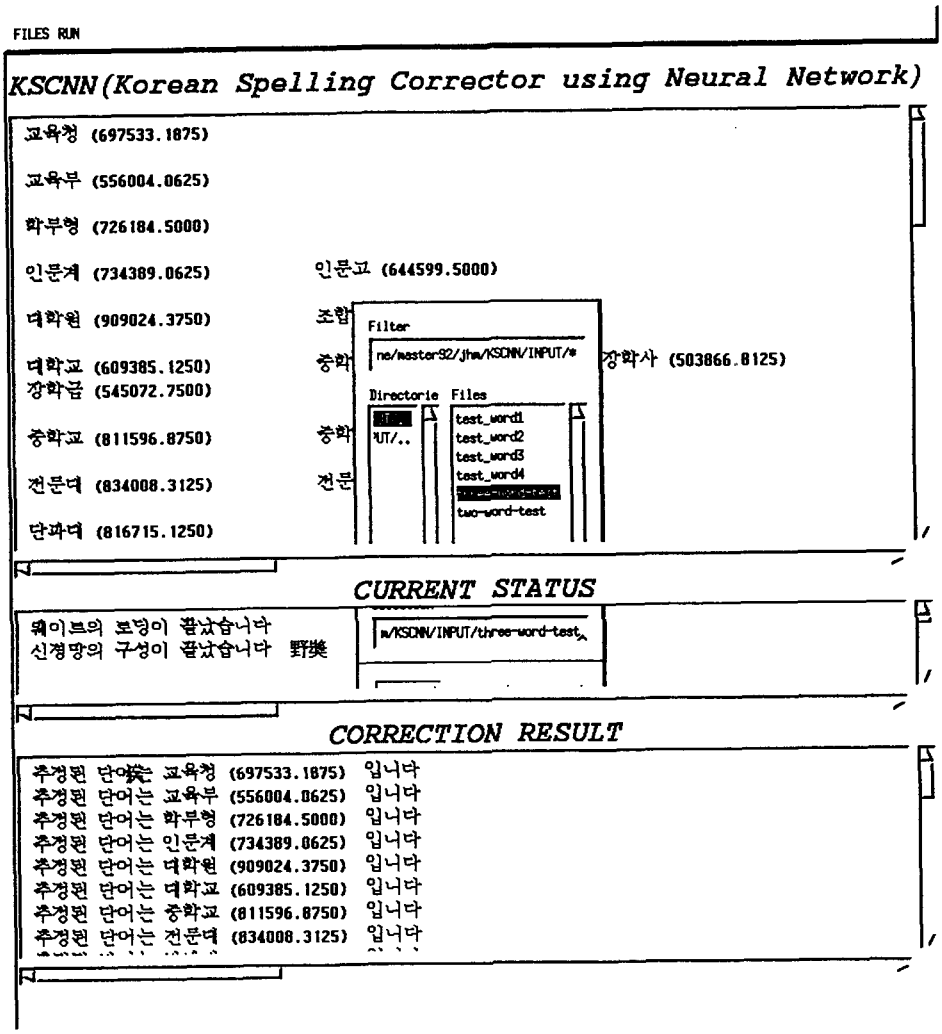

그림 5. KSCNN의 실행 예

된 단어를 가지는 응용에 적합하며 퍼지 멤버쉽 함수의 조정으로 음성인식 및 문자인식 오류 교정에 적합한 시스템으로 개발될 수 있다.본 철자 교정기의 앞으로의 방향은 다음과 같다.

첫째, 무종성 단어들의 학습 방법에 대한 연구가 필요하다. 본 철자 교정기에서 학습에 실패한 단어들을 분석해보면 무종성을 가진 단어들이 많은 비율을 차지하고 있기 때문이다. 이에 대한 해결 방안의 하나인 무종성을 위한 새로운 가상 노드의 추가가 시스템의 학습 능력을 향상시킬 수 있다고 본다.

둘째, 보다 정교한 퍼지 멤버쉽 함수의 정의가 필요하다. 실험을 통해서 오류 단어들의 유형 및 자판상에서의 상관 관계를 분석하고 그에 따라 퍼지 멤버쉽 함수를 다시 정의한다면 교정율을 보다 더 향상시킬 수 있을 것이다. 또한, 문자 및 음성인식을 위하여 각종 인식 디바이스의 특성을 고려한 퍼지 멤버쉽 함수의 정의도 필요하리라 본다.

참고문헌

[1] J. Freeman & D. Skapura, *"Neural Networks: Algorithms, Applications, and Programming Techniques"*, Addison-Wesley, 1991.

[2] Hong Feng Yin & Ju Wei Tai, "A New Kind of Associative Memory Network Model", *International Joint Conference on Neural Networks, Washington, DC*, 1990.

[3] Hong Feng Yin & Ju Wei Tai, "An Associative Memory Model of Language", International Joint Conference on Neural Networks, San Diego, 1990.

[4] J. Hull & S. Srihari, "Experiments in Text Recognition with Binary n-Gram and Viterbi Algorithms", *IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, NO. 3*, 1982.

[5] T. Kohonen, *"Self-Organization and Associative Memory"*, Vol. 8 of Springer Series in Information Sciences, 1988.

[6] A. Viterbi, "Error Bounds for Convolution Codes and an Asymptotically Optimal Decoding Algorithm", *IEEE Trans. Inform. Theory, Vol. IT-13*, 1967.

[7] 민병우 & 이성환, "문자 인식을 위한 오인식 수정 기술", 정보과학회지, 제9권 제1호, 1991.