개선된 Levenshtein Distance 알고리즘을 사용한 어휘 탐색 시스템

이종섭*, 오상엽**

세명대학교 교양과정부*

가천대학교 글로벌캠퍼스 IT대학 컴퓨터미디어융합학과*

Vocabulary Retrieve System using Improve Levenshtein Distance algorithm

Jong-Sub Lee*, Sang-Yeob Oh**

Dept. of General Education, Semyung University*

Dept. of Computer Media Convergence, College of IT, Gachon University**

요약 기존의 Levenshtein distance 알고리즘은 어휘들 간의 순서가 정해져 있지 않은 경우에 사용되므로 어휘 탐색 작업의 중요도를 구분할 수 없는 단점을 가진다. 본 연구에서 제안하는 개선된 Levenshtein 방법에서는 효율적으로 사용빈도에 따라 어휘들을 탐색하고, 어휘들 간의 순서를 가지는 가중치를 부여한다. 따라서 어휘의 수가 증가하는 경우에도 효율적으로 사용빈도에 따라 어휘를 탐색하여 인식률이 저하되는 단점을 해결하고, 인식 시간을 향상 및 탐색 공간의 효율적으로 관리할 수 있는 장점을 가진다. 제안한 시스템을 분석한 결과 실내 환경에서 어휘 중속 인식률은 97.81%, 어휘 독립 인식률은 96.91%의 인식률을 나타냈다. 또한, 실외 환경에서 어휘 중속 인식률은 91.11%, 어휘 독립 인식률은 90.01%의 인식률을 나타냈다.

주제어 : 레빈스타인 거리, HMM, 어휘 인식, 가중치

Abstract In general, Levenshtein distance algorithm have a problem with not distinguish the consideration of vocabulary retrieve, because Levenshtein methode is used to vocabulary order are not defined. In this paper, we propose a improved Levenshtein methode, it effectively manage the vocabulary retrieve by frequency use of a vocabulary, and it gives the weight number which have a order between vocabularies. Therefore proposed methode have a advantage of solve the defect of perception rate in the case of increase the vocabulary, improve the recognition time become higher and it can be effectively retrieval space management.. System performance as a result of represent vocabulary dependence recognition rate of 97.81%, vocabulary independence recognition rate of 96.91% in indoor environment. Also, vocabulary dependence recognition rate of 91.11%, vocabulary independence recognition rate of 90.01% in outdoor environment.

Key Words : Levenshtein distance, HMM, Vocabulary Recognition, weight number

* 이 논문은 2013년도 가천대학교 교내연구비 지원에 의한 결과임.(GCU-2013-R265)

Received 12 September 2013, Revised 5 October 2013

Accepted 20 November 2013

Corresponding Author: Sangyeob Oh(The University of Gachon)

Email: syoh1234@gmail.com

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

음성 인식은 인식 대상 어휘를 기본으로 데이터베이스를 대상으로 모델을 생성하며 모델의 형태는 HMM(Hidden Markov Model)이 사용된다[1]. HMM은 처리하는 대상인 어휘 데이터를 기반으로 필요한 확률 파라미터를 추정하게 된다. 이러한 샘플들은 특징 파라미터로써 신호 처리 기법을 이용하여 추출하며 순수하게 이 특징 파라미터들만으로 이루어진 모델을 음향모델이라 한다[2, 3]. 즉, HMM은 시간 천이에 의해 변경되는 신호를 모델링하는 방법으로서 숨겨진 상태천이 확률 과정과 각 상태에서 관측신호를 발생하는 관측 확률과정의 두 개의 랜덤프로세스로 구성되며, HMM의 상태천이 확률 및 관측확률은 시간에 독립이며 현재 상태에만 의존한다고 가정한다.

어휘 인식을 처리하는 방법에는 대부분 통계적 어휘 인식 방법이 사용되며, 이들 방법에서는 학습단계에서 추출된 특징 벡터를 사용하여 기준이 되는 음향모델이나 어휘 대상의 기준 패턴 정의하고, 어휘 인식을 위해 비터비(viterbi)와 같은 패턴 정합 알고리즘을 이용하여 정의된 학습된 패턴 중에서 가장 유사한 것을 인식결과로 출력한다.

최근 여러 종류의 휴대용 단말기가 생활에 밀접하게 이용되면서 이들 기기에서 대용량의 데이터 처리와 인식이 어려운 문제가 있으며, 이들 데이터들을 사용이 어려운 문제를 가지고 있다. 일반적으로 어휘 인식 알고리즘 등에서는 미리 학습된 패턴 중에서 유사도가 가장 높은 것을 찾아서 인식 결과로 출력하지만[4,5,6], 데이터베이스에서 인식 대상 어휘의 수가 늘어나면 어휘 인식 알고리즘이 복잡해지고 대규모의 탐색공간을 필요로 하게 되며 처리시간이 길어지므로 인식 시스템에서 소요되는 메모리의 용량이 커지는 단점이 있다. 기존의 Levenshtein distance 알고리즘은 어휘들 간의 순서가 정해져 있지 않은 경우에 사용되며, 어휘 탐색 작업에 의해 추출되는 어휘에 음향확적인 정보가 부가적으로 포함하고 있어 어휘 데이터에 대한 메타 정보를 제공하여 구분하는 단점을 가진다[7].

본 연구에서는 개선된 Levenshtein distance 알고리즘에서는 어휘들 간의 순서를 부여하는 가중치를 부여하여 어휘의 수가 증가하는 경우에도 효율적으로 사용빈도에

따라 어휘를 탐색하여 처리하여 인식율이 저하되는 단점을 해결하고, 인식 시간 및 어휘수가 증가하여도 탐색공간의 효율성을 가진다. 개선된 Levenshtein distance 알고리즘을 적용한 시스템을 분석한 결과 기존의 알고리즘에 비해 시스템 성능 평가 결과 실내 환경에서의 어휘 종속 인식률은 97.81%, 어휘 독립 인식률은 96.91%의 인식률을 나타냈으며 인식률에서 비터비 알고리즘에 비해 2.3% 향상된 결과를 나타냈다. 또한, 실외 환경에서는 어휘 종속 인식률은 91.11%, 어휘 독립 인식률은 90.01%의 인식률을 나타냈으며 비터비 알고리즘에 비해 1.1% 향상된 결과를 나타냈다.

제 2장에서는 HMM과 기존의 Levenshtein distance 알고리즘을 다루고, 제 3장에서는 본 연구에서 제안한 시스템에 대하여 설명한다. 제 4장에서는 제안한 시스템의 실험결과에 대하여 설명하고 제 5장에서 결론을 기술한다.

2. 관련 연구

2.1 HMM(Hidden Markov Model)

HMM(Hidden Markov Model)은 음성 신호가 가지는 고유의 스펙트럼 변화와 이 스펙트럼이 시간이 지나면서 발생하는 변화를 함께 처리하는 특징을 가진다. HMM 음성 인식 단계에서는 입력된 음성 데이터를 가지고 적합한 모델을 생성하며, 일련의 연속된 상태들로부터 이산 신호를 생성하는 확률 과정 모델을 사용한다. 모델의 파라미터 추정은 카테고리 정보가 있는 음성 데이터베이스를 사용한다.

[Fig. 1] HMM of 3-State

HMM에서 음성 인식을 수행하기 위해서 관측 열 특징 벡터 x 가 입력되었을 때 확률이 어떻게 계산되는지의 문제로 전향 알고리즘과 후향 알고리즘을 이용하여 계산한다. 또한, 은닉 상태 열을 찾는 관측 열 $O = \{o_1, o_2, \dots, o_T\}$ 과 모델 $\lambda = (A, B, \pi)$ 가 주어졌을 때, 최적의 상태 열 $Q = \{q_1, q_2, \dots, q_T\}$ 을 찾기 위해서 Viterbi 알고리즘을 사용한다[8].

2.2 Levenshtein distance Algorithm

음성 인식에서 Levenshtein distance 알고리즘은 처리하고자 하는 음소들이 우열을 갖지 않는 경우에 적합한 알고리즘이지만 사용되는 음소열에 음향 정보를 가지므로 음소들 사이의 변화에 대한 가중치를 다르게 적용해야 하는 단점이 있다. Levenshtein distance 알고리즘은 2개의 음소에 패턴 정합을 위해 사용되며, 이를 위해 DP(Dynamic Programming) 정합 매트릭스를 사용하여 2개 음소 패턴에 필요한 변환 횟수를 패턴간의 거리로 표시한다. 즉, 음소열 X와 음소열 Y사이의 Levenshtein distance는 음소열 X와 음소열 Y가 같아지기 위해서 필요한 최소한의 삽입, 삭제, 대체 변환의 수를 의미한다. 두 개의 음소열이 $X = x_1, x_2, \dots, x_m$ 과 $Y = y_1, y_2, \dots, y_n$ 이고, x_i, y_j 는 각각 음소열 X, Y의 i 번째와 j 번째의 음소이고 음소열의 음소 길이는 각각 $|X| = m, |Y| = n$ 으로 표현된다. 이 경우 Levenshtein distance $LD_{m,n}$ 의 의미는 하나의 음소열 $x_1, x_2, \dots, x_m$ 을 다른 음소열 $y_1, y_2, \dots, y_n$ 로 정합하는데 필요한 최소 연산수이며, 두 음소열 사이의 Levenshtein distance는 아래 식1과 같이 계산된다.

$$LD_{m,n} = \min \begin{cases} LD_{m,n-1} + 1 \\ LD_{m-1,n} + 1 \\ LD_{m-1,n-1} + \delta_{(m,n)} \end{cases} \quad (1)$$

알고리즘에서 삽입, 삭제, 대체에 대한 가중치를 조절하여 사용하며 삽입 시 $LD_{m,n} = LD_{m,n-1} + 1$ 로 계산하고, 삭제 시 $LD_{m,n} = LD_{m-1,n} + 1$ 로 계산한다. 대체를 위한 $x_m \neq y_n$ 인 경우 $LD_{m,n} = LD_{m-1,n-1} + 1$ 로 계산하고, 일치 시 $x_m = y_n$ 인 경우 $LD_{m,n} = LD_{m-1,n-1}$ 으로 계산한다[9,10,11].

3. 시스템 모델

3.1 개선된 Levenshtein distance 알고리즘

인식 과정은 문맥 종속적인 음향 모델을 바탕으로 인식을 수행할 수 있는 어휘를 발음 사전에 등록시켜 사용한다. 등록된 어휘에 대하여 인식을 수행하고 인식 어휘가 변경되거나 추가되었을 때 인식 어휘 변경 및 추가 과정을 거쳐 변경되거나 추가된 인식 어휘를 음소열로 변환된다. 음소열은 PLU(Phone Likely Unit)로 Tagging되어 발음 사전에 등록되고 인식을 할 수 있는 어휘로 등록된다.

개선된 Levenshtein distance을 위한 알고리즘을 다음과 같이 구성하였다.

```

Begin
  for I in D'Range (1) loop
    D (I, 0) := I;
  end loop;

  for I in D'Range (2) loop
    D (0, I) := I;
  end loop;

  for J in T'Range loop
    for I in S'Range loop
      if S (I) = T (J) then
        D (I, J) := D (I - 1, J - 1);
      else
        D (I, J) := Natural'Min
          (Natural'Min (D (I - 1, J) + 1, D (I, J - 1) + 1),
           D (I - 1, J - 1) + 1);
      end if;
    end loop;
  end loop;
  return D (S'Length, T'Length);
end Levenshtein_Distance;

Begin
  Text_IO.Put_Line
    ("kitten -> sitting:" & Integer'Image
     (Levenshtein_Distance ("kitten", "sitting")));
  Text_IO.Put_Line
    ("rosettaacode -> raisethysword:" & Integer'Image
     (Levenshtein_Distance ("rosettaacode", "raisethysword")));
end

```

기존 Levenshtein distance 알고리즘에 나타나는 단점을 음소들 사이의 변화에 대한 가중치를 다르게 적용하여 개선하였으며 어휘수가 증가하여도 탐색공간의 효율성과 처리시간을 단축하였다.

두 음소열 “가천”과 “가청”을 기존의 Levenshtein

distance를 사용하여 거리를 계산할 경우 그림 2와 같이 최소거리가 계산된다. 비교가 되는 두 개의 음소열을 X는 “ㄱ, ㅏ, ㅓ, ㅕ, ㅗ”이라 하고, Y는 “ㄱ, ㅏ, ㅓ, ㅕ, ㅗ”이라 할 때, 그림 2의 (1)은 두 음소열을 통하여 행렬을 초기화했을 때를 나타낸다. (2)는 음소열 X의 i=1번째 음소와 음소열 Y의 모든 음소를 대상으로 거리를 계산한 결과이다. 완성된 그림 2의 (4)에서 오른쪽 하단의 숫자 ‘2’가 두 음소열에 대한 Levenshtein distance가 된다.

제안한 방법을 사용하였을 경우 어휘수가 증가하여도 음소열을 사용하여 탐색하므로 인식율을 효율적으로 향상시킨다.

		ㄱ	ㅏ	ㅓ	ㅕ	ㅗ
	0	1	2	3	4	5
ㄱ	1					
ㅏ	2					
ㅓ	3					
ㅕ	4					
ㅗ	5					

		ㄱ	ㅏ	ㅓ	ㅕ	ㅗ
	0	1	2	3	4	5
ㄱ	1	0	1	2		
ㅏ	2	1	0	1		
ㅓ	3	2	1	0		
ㅕ	4	3	2	1		
ㅗ	5	4	3	2		

		ㄱ	ㅏ	ㅓ	ㅕ	ㅗ
	0	1	2	3	4	5
ㄱ	1	0	1	2	3	4
ㅏ	2	1	0	1	2	3
ㅓ	3	2	1	0	1	2
ㅕ	4	3	2	1	0	1
ㅗ	5	4	3	2	1	2

[Fig. 2] Levenshtein distance calculation for two phoneme strings

3.2 유사 어휘 가중치 조정

기존의 Levenshtein distance 알고리즘은 변환 단계의 어휘들이 우열을 갖지 않는 경우에 사용되어 어휘 변화에 대한 가중치를 다르게 적용해야 하는 단점을 개선하여 어휘들 간의 순서를 부여하는 가중치를 부여하고 어휘의 수가 증가하는 경우에도 효율적으로 사용빈도에 따라 어휘를 탐색하여 처리하여 인식율이 저하되는 단점을 해결하고, 인식 시간을 향상할 수 있는 장점을 가진다.

음성을 모델링하는 방법으로는 HMM 기법을 주로 사용한다. 이는 확률 모델의 일종으로써 Markov Model의 형태를 가지며 상태열(state sequence)이 은닉되어 있는 특징을 갖는다. “ㄱ”이라는 음소가 N개의 상태를 갖는 HMM으로 모델링 될 경우 각각의 상태는 현재 상태에서 특정한 스펙트럼 특징을 나타낼 확률, 즉 출력 확률과 현

재 상태에서 임의의 상태로 천이할 확률, 즉 천이확률을 갖는다. 모든 음소의 모델은 위와 같은 고유의 출력 확률과 천이 확률을 갖는다. 음성인식 과정은 입력된 음성이 임의의 음소 모델로부터 생성되었을 확률을 계산하는 과정이다. 즉 입력된 음성이 “ㄱ”, “ㅏ”, “ㅓ”이라는 각각의 음소 모델로부터 순차적으로 생성되었을 확률이 가장 크다면 입력된 음성은 “강”이라는 말로 인식되며 다음과 같은 식으로 표현된다.

$$\hat{S} = \underset{k}{\operatorname{argmax}} \left[\prod_{t=1}^T p(\vec{x}_t | \lambda_k) \right] \quad (2)$$

λ_k 는 화자 모델을 나타내며, $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_T\}$ 는 음성 데이터의 프레임들에서 추출한 특징 벡터 $\vec{x}_t$ 들의 집합이고, T 는 음성 데이터의 총 프레임 개수이다. 화자 모델에 대한 조건부 확률을 등록된 모든 화자모델에 대하여 계산한 다음 이 중에서 최대 확률을 갖는 인덱스 k 의 화자를 인식의 결과로 결정하게 된다.

4. 실험 결과

본 논문에서 제안한 개선된 Levenshtein distance 알고리즘의 성능 검증을 위하여 인식 실험을 수행하였다. 훈련과정과 실험환경과의 불일치 문제를 해결하기 위해 잡음처리하는 위너(wiener) 필터를 사용하였다. 어휘 목록은 지역명으로 구성하였다. 인식 실험에서는 실험에 참가한 화자가 어휘 중에서 임의로 10단어씩 발음하여 총 60단어를 대상으로 실험을 수행하였다. 어휘는 실내 환경과 잡음 환경에서 이동기기 등에 내장되어 있는 내장형 마이크로폰을 사용하여 16kHz Mono로 녹음 하였고, 16bit PCM 양자화를 사용하였다. 실험 어휘는 실내 5명, 실외 5명 등 총 15명의 성인 남성이 참가하였다. 실내 환경은 50~55dB이고, 실외 환경은 70~75dB의 소음환경 하에서 실험하였다.

제안한 시스템의 성능 평가를 위하여 기존 방식과 비교 실험을 하였다. 실험은 화자 종속형과 화자 독립형으로 구분하여 실험하였으며 화자 종속형은 음성 모델을 만들 때 참가하였던 화자가 직접 인식에 참여하여 실험한 것이며 화자 독립형은 음성 모델에 참가하지 않았던

화자가 인식을 한 실험을 나타낸다.

〈Table 1〉 Indoor Environment Recognition Rate

Vocabulary	Error Correction Rate (%)		
	Viterbi	DTW	Levenshteini
Speech Dependent	95.51	92.17	97.81
Speech Independent	94.97	90.91	96.91

표 1의 결과에서 보는 것과 같이 시스템 성능 평가 결과 실내 환경에서의 어휘 종속 인식률은 97.81%, 어휘 독립 인식률은 96.91%의 인식률을 나타냈으며 인식률에서 Viterbi 알고리즘에 비해 2.3% 향상된 결과를 나타냈다.

표 2의 결과에서의 시스템 성능 평가 결과 실외 환경에서는 어휘 종속 인식률은 91.11%, 어휘 독립 인식률은 90.01%의 인식률을 나타냈으며 비터비 알고리즘에 비해 1.1% 향상된 결과를 나타냈다.

〈Table 2〉 Outdoor Environment Recognition Rate

Vocabulary	Error Correction Rate (%)		
	Viterbi	DTW	Levenshteini
Speech Dependent	90.01	88.97	91.11
Speech Independent	89.15	86.81	90.01

5. 결론

음성 인식은 데이터베이스에 저장된 어휘 모델과의 비교를 통해 인식 대상을 판단하며, 모델의 생성과 모델의 HMM을 기반으로 한다. HMM은 처리하는 대상인 어휘 데이터를 기반으로 필요한 확률 파라미터를 추정하고, 이러한 샘플들은 특징 파라미터로써 신호 처리 기법을 이용하여 추출하며, 어휘 인식을 처리하는 통계적 어휘 인식 방법에서는 학습단계에서 추출된 특징 벡터를 사용하여 기준이 되는 음향모델이나 어휘 대상의 기준 패턴 정의하고, 어휘 인식을 위해 비터비와 같은 패턴 정합 알고리즘을 이용하여 정의된 학습된 패턴 중에서 가장 유사한 것을 인식결과로 출력한다. 그러나 데이터베이스에서 인식 대상 어휘의 수가 늘어나면 어휘 인식 알고리즘이 복잡해지고 대규모의 탐색공간을 필요로 하게 되며

처리시간이 길어지므로 인식 시스템에서 소요되는 메모리의 용량이 커지는 단점이 있다. 기존의 Levenshtein distance 알고리즘은 어휘들 간의 순서가 정해져 있지 않은 경우에 사용되며, 어휘 탐색 작업에 의해 추출되는 어휘에 음향학적인 정보가 부가적으로 포함하고 있는 단점을 가진다.

본 연구에서는 개선된 Levenshtein distance 알고리즘에서는 어휘들 간의 순서를 부여하는 가중치를 부여하여 어휘의 수가 증가하는 경우에도 효율적으로 사용빈도에 따라 어휘를 탐색하여 처리하여 인식율이 저하되는 단점을 해결하고, 인식 시간을 향상할 수 있는 장점을 가진다. 개선된 Levenshtein distance 알고리즘을 적용한 시스템을 분석한 결과 기존의 알고리즘에 비해 시스템 성능 평가 결과 실내 환경에서의 어휘 종속 인식률은 97.81%, 어휘 독립 인식률은 96.91%의 인식률을 나타냈으며 인식률에서 비터비 알고리즘에 비해 2.3% 향상된 결과를 나타냈다. 또한, 실외 환경에서는 어휘 종속 인식률은 91.11%, 어휘 독립 인식률은 90.01%의 인식률을 나타냈으며 비터비 알고리즘에 비해 1.1% 향상된 결과를 나타냈다.

ACKNOWLEDGMENTS

This work was supported by the Gachon University research fund of 2013.”(GCU-2013-R265)

REFERENCES

- [1] Chan-Shik Ahn, Sang-Yeob Oh. Efficient Continuous Vocabulary Clustering Modeling for Tying Model Recognition Performance Improvement. Journal of the Korea Society of Computer and Information. Vol. 15, No. 1, pp. 177-183, 2010.
- [2] Chan-Shik Ahn, Sang-Yeob Oh. Echo Noise Robust HMM Learning Model using Average Estimator LMS Algorithm. The Journal of Digital Policy and Management. Vol. 10, No. 10, pp. 277-282, 2012.
- [3] Kris Demuynck, Tom Laureys, Dirk van

- Compernelle, and Hugo van Hamme, FLAVOR: a flexible architecture for LVCSR, In EUROSPEECH-2003, pp. 1973-1976, 2003.
- [4] K. Demuyne, J. Duchateau, and D. Van Compernelle, A static lexicon network representation for cross-word context dependent phones, In Proc. EUROSPEECH, Vol.1, pp. 143-146, 1997.
- [5] Chan-Shik Ahn, Sang-Yeob Oh. Gaussian Model Optimization using Configuration Thread Control In CHMM Vocabulary Recognition. The Journal of Digital Policy and Management. Vol. 10, No. 7, pp. 167-172, 2012.
- [6] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, Prentice-Hall, 2000.
- [7] S. M. Naqvi, M. Yu, J. A. Chamber. A Multimodal Approach to Blind Source Separation of Moving Sources. IEEE Trans. Signal Processing. Vol. 4, No. 5, pp. 895-910, 2010.
- [8] Y. Shao, S. Srinivasan, Z. Jin, D. Wang. A Computational Auditory Scene Analysis System for Robust Speech Recognition. Computer Speech & Language. Vol. 24, No. 1, pp. 77-93, 2010.
- [9] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.
- [10] Eiichi Tanaka and Tamotsu Kasai, Synchronization and Substitution Error-correcting codes for the Levenshtein Metric, IEEE Trans. Information Theory, Vol.IT-22, No.2, pp. 156-176, 1976.
- [11] S. Ortmanns, A. Eiden, H. Ney, and N. Coenen, Look-ahead Techniques for Fast Beam Search, InProc. IEEE ICASSP-1997, pp. 1783-1786, 1997.
- [12] W. Daelemans, S. Buchholz, and J. Veenstra, Memorybased shallow parsing, in Proc. CoNLL, pp. 53-60, 1999.
- [13] Justin Zobel and Philip Dart Phonetic String Matching: Lessons from Information Retrieval, SIGIR'96, pp. 166-173, 1996.

오 상 엽(Oh, Sang Yeob)

- 1991년 2월 : 광운대학교 대학원 전 자계산학과(이학석사)
- 1999년 2월 : 광운대학교 대학원 전 자계산학과(이학박사)
- 2007년 2월 ~ 현재 : 가천대학교 IT대학 인터랙티브미디어학과 교수
- 관심분야 : 버전관리, 형상관리, 음성/음향 신호 처리, 차량 통신

• E-Mail : syoh1234@gmail.com

이 종 섭(Lee, Jong Sub)

- 1993년 8월 : 광운대학교 대학원 전 자계산학과 (이학석사)
- 1997년 2월 : 광운대학교 대학원 전 자계산학과 (박사수료)
- 2012년 3월 ~ 현재 : 세명대학교 교양과정부 교수
- 관심분야 : 분산객체처리시스템, 음성/음향 신호 처리, 차량 통신

• E-Mail : 99jslee@semyung.ac.kr