



한글 단어 및 문장의 음성 인식률 향상을 위한 음성 인식 API 개발

임대환*, 남기훈**

Development of Speech Recognition API to Improve Speech Recognition Rate of Korean Words and Sentences

Dai-Hwan Lim*, Ki-Hun Nam**

요약

본 논문에서는 한글 단어와 문장의 음성 인식률을 높이기 위해 새로운 API를 제안한다. 음성 인식은 마이크와 같은 센서를 사용하여 얻은 음향학적 신호를 단어나 문장으로 변환시키는 기술이다. 현재 음성 인식을 지원하고 있는 다양한 프로그램들도 한글 인식 결과에는 만족스럽지 못하고 있다. 이를 개선하기 위해 기존 방식의 음성 인식 구조에 새로운 알고리즘을 추가하여 음성 인식률을 높이게 하였다. 알고리즘은 전처리, 예외, 숫자 표기 변환, 근사치 비교 및 레벤스테인 거리, 해싱, 자음 비교 등이다. 실험 환경은 스마트폰 테스트 앱을 개발하여 실험하였다. 제안된 구조는 기존 방식에 비해 인식률이 표준어는 대략 8.2% 향상되었음을 확인하였다.

Abstract

In this paper, we propose a new API to improving the speech recognition rate of Korean words and sentences. Speech recognition is a technology that converts acoustic signals from sensors such as microphones into words or sentences. Various programs currently supporting speech recognition are also not satisfied with the results of Hangul recognition. To improve this, a new algorithm was added to the existing speech recognition structure to increase the speech recognition rate. The algorithms are preprocessing, exceptions, numeric conversion, approximate comparison, Levenshtein distance, hashing, consonant comparison, etc. The experimental environment was tested by developing a smartphone test application. The proposed structure confirmed that the recognition rate of standard words was approximately 8.2% better than that of conventional structure.

Keywords

speech recognition, api(application programming interface), levensthein distance, hasing, recognition rate

* 서경대학교 컴퓨터공학과 교수

- ORCID: <https://orcid.org/0000-0002-0749-7983>

** 서경대학교 컴퓨터공학과 교수(교신저자)

- ORCID: <https://orcid.org/0000-0002-0724-912X>

· Received: Oct. 15, 2019, Revised: Oct. 30, 2019, Accepted: Nov. 02. 2019

· Corresponding Author: Ki-Hun Nam

Dept of Computer Engineering, SeoKyeong University, 4-ding, Jeongneung, Seongbuk-gu, Seoul, Korea

Tel.: +82-2-940-7667, Email: namkh@skuniv.ac.kr

I. 서 론

음성 인식 기술은 인간의 자연어 발화를 컴퓨터가 자동으로 이해하고 처리하는 알고리즘을 연구하는 분야이다. 음성 인터페이스 다양한 기능으로 인해 소통형 서비스가 이루어지는데 애플의 Siri, 마이크로소프트의 코타나, 구글의 어시스턴트, SKT의 누구, KT의 기가지니 등이 서비스를 진행하고 있다 [1]. IT 시장조사 기관 가트너에서 2019년에는 스마트폰과 사용자 간의 상호 작용 중 20%정도 가상 개인 비서(Virtual personal assistants)를 통해 이루어 질것으로 예측하였으며 많은 기기 및 IoT 장비가 제로터치 사용자 인터페이스 기반으로 작동할 것으로 전망하고 있다[2]. 한국어는 형태소 분석에 기반한 어휘 선정의 문제와 함께 인식 대상 어휘에 제한으로 인한 인식 오류 발생이 많은 편이어서 한국어 특성에 적합한 음성 인식 처리 기술의 개선이 절실히 필요하다[3].

본 논문에서는 이러한 문제점을 개선하고자 단어 및 문장 음성 인식 처리 기술을 제안한다. 2장에서 관련 연구를 3장에서는 제안 시스템을 4장은 실험 및 결과를 5장에서는 결론 순으로 구성된다.

II. 관련 연구

음성 언어 기술은 자동적 수단에 의하여 음성으로부터 언어적 의미 내용을 식별하는 기술이다. 마이크와 같은 소리 입력 센서를 통해 얻은 음향학적 신호(Acoustic speech signal)를 음성 분석, 음소 인식, 단어 인식, 문장 해석, 의미 추출 등으로 분류하고, 특징을 추출하여 음성 모델 데이터베이스와 비교하는 방식으로 음성 인식을 하게 된다. 음성 분석은 단어 인식까지를 말하는 경우가 있다. 음성 인식의 목표는 자연스러운 발성에 의한 음성을 인식하여 완전한 단어 또는 문장으로 정확하게 변환시키는 것이다[4][5].

그림 1은 음성 인식 기술의 기본적인 원리 구조이다. 그림 1의 구조를 바탕으로 음성 인식을 지원하는 Open API는 Google의 Cloud Speech API가 있으며, 카카오의 뉴톤(Newtone)과 네이버의 Clova

Speech API가 제공되고 있다. Google과 네이버 API는 인공 지능을 함께 지원하고 있다. 숫자, 한글 음성, 문장 음성 인식의 테스트 결과 Google Speech의 error rate은 36.3%, 네이버의 Clova Speech는 16.6%, 카카오의 Newtone은 8.6%의 error rate을 보였다[6]. 인공 지능을 사용하지 않는 Newtone API가 실험 결과 음성 인식률이 높게 나왔다는 점을 감안하여 은닉 마르코프 모델(Hidden Markov Model, HMM)과 신경회로망(Neural network), 선형예측부호화(Linear Predictive Coding, LPC)등의 혼합 기법을 사용하지 않는 방법으로 개발하려 한다[7][8].

음성 인식률을 높이기 위해 기본 원리 구조를 통해 얻어진 인식결과 데이터를 재처리하는 방식을 제안하려 한다.

그림 2에서의 제안하는 전체 구조도를 보여준다. 입력 문장은 전처리 과정을 거친 후, 문장 분류 및 처리하여 단어를 추출하고 알고리즘들을 거쳐 표준어 단어와 문장들을 비교한다. 적합한 단어와 문장이 있으면 표준어로 출력되고 불일치되는 단어들은 가중치 처리를 하여 표준어 테이블에 저장한다.

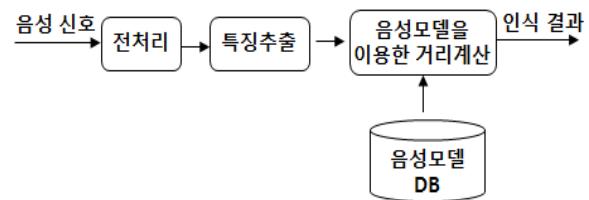


그림 1. 음성 인식 기술의 기본 구조

Fig. 1. Structure of speech recognition technology

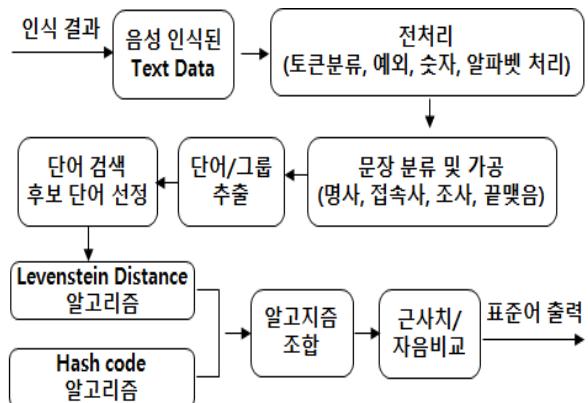


그림 2. 제안하는 음성 인식 기술

Fig. 2. Proposal of speech recognition technology

III. 제안 시스템

3.1 데이터 전처리

문장 단위의 음성 인식은 음성 입력을 하나의 문장 형태로 전환하여 받는다. 전처리 과정은 그림 3과 같이 TTS(Text to Speech)를 거쳐 나온 문장을 띠어쓰기 기준으로 분리하여 예외 단어가 있는지 검사한다.

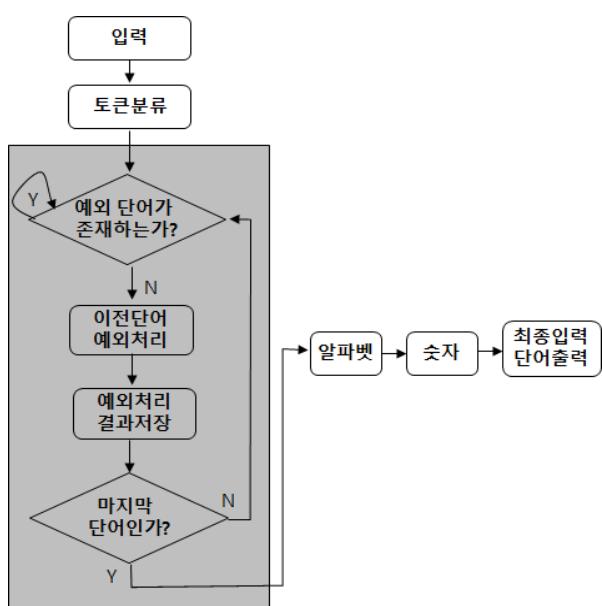


그림 3. 전처리 알고리즘 그림

Fig. 3. Preprocessing algorithm

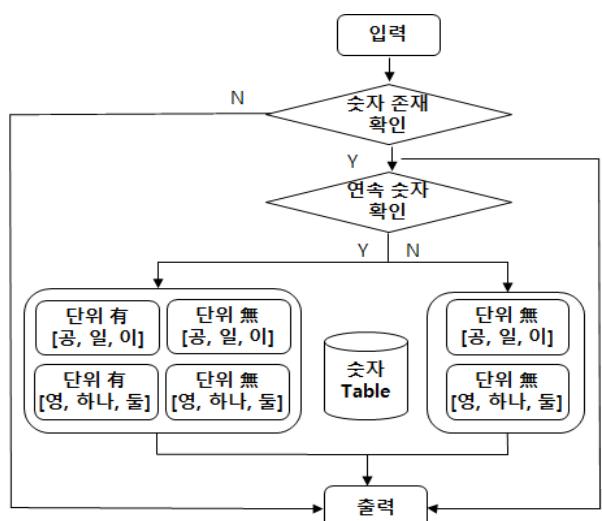


그림 4. 숫자 표기 변환 알고리즘

Fig. 4. Algorithm of numeric conversion

분류된 토큰 수만큼 반복처리를 하며 예외, 숫자, 알파벳 처리를 한다. 토큰으로 분리된 문장은 각 단어의 순서에 기준을 두어 예외 단어 테이블에 존재하는 예외 단어 리스트와 부분 일치 여부를 확인한다.

임의의 단어가 예외 단어와 부분 일치 한다면 그 단어 뒤에 오는 단어와 조합하여 다시 예외 단어 리스트와 부분 일치 여부를 확인한다. 부분 일치 하지 않는다면 이전에 부분 일치 되었던 앞 단어는 예외처리 과정으로 넘어가고 부분 일치 되지 않은 단어는 다시 예외 단어 리스트와 부분 일치 하는지 검증한다. 예외 단어 리스트 중 일치하는 단어가 있을 경우 해당 원래 단어로 변환한다.

숫자처리는 그림 4와 같이 입력 값에 숫자가 포함된 단어가 있을 경우 숫자를 한글로 변경하기 위한 알고리즘으로 숫자의 단위와 한글 표기법을 고려하고 간혹 숫자가 포함된 명칭이 있는 경우를 대비하여 숫자가 한글 표기로 변경되지 않은 결과도 함께 저장한다. 만약 연속되는 숫자가 존재할 경우 단위와 숫자 표기를 고려한 결과를 출력하고 없을 경우에는 숫자 표기만 고려한 결과를 출력한다.

3.2 입력 문장 분류 및 가공

전처리를 거친 입력 문장들은 접속사격 조사를 기준으로 분리하여 문장을 분류한다. 그림 5와 같이 접속사격 조사를 가진 단어를 기준으로 문장을 나누고 같은 기준으로 단어를 재처리하고 조합한다.

조사기준 문장 분류는 전처리를 거쳐 나온 입력 문장을 접속사격 조사 리스트와 비교하여 단어 끝에 접속사격 조사를 가진 단어가 있을 때 그 단어를 기준으로 단어를 분리한다. 분리된 단어들은 하나의 단어로 취급되며 그 단어에서 명사 추출과 “먹었다”, “먹었어”, “먹음”과 같은 끝맺는 단어의 삭제와 접속사격 조사의 제거과정을 거친다. 단어 조합을 거쳐 하나의 단어로 가공된다. 명사 추출 성공 시 다음 단어로 넘어가고 실패 시 끝맺는 단어 삭제과정으로 들어간다.

끝맺는 단어 삭제는 끝맺는 가이드에 존재하는 “먹었다”라는 동사를 제거하기 위한 것으로 끝맺는 단어가 존재할 시 그 단어는 삭제되어 단어 조합 과정에서 제외시킨다.

접속사격 조사 제거 과정은 접속사격 조사 테이블에 존재하는 조사가 단어의 끝에 존재하는지 확인하여 제거하고 명칭 추출과 끝맺는 단어 삭제 과정에서 걸려내지 못한 단어들을 확인하는 마지막 절차이다.

단어 조합은 접속사격 조사를 가진 단어만 거치는 과정으로 명칭 추출, 끝맺는 단어 제거, 접속사격 조사 제거 과정을 모두 거쳐 추출된 단어를 입력 문장 순서 기준으로 조합한다.

3.3 단어와 그룹 추출

입력 데이터추출을 거쳐 만들어진 단어를 명사 테이블에 존재하는 단어와 매칭하여 추출한다. 접속사격 조사로 나누어진 단어인지 판별하고 접속사격 조사로 나누어진 경우는 입력데이터 추출과정에서 접속사격 조사로 인해 변환되지 못한 예외 단어가 있는지 확인하여 예외 처리 과정을 좀 더 진행하고 없을 경우 명사 부분 일치 추출 및 단어 검색하여 결과 단어를 추출한다.

명사 부분 일치 추출은 화자가 정확하지 않은 단어를 말한 경우, 그 단어로 시작되는 단어 혹은 그

단어로 끝나는 단어가 기준이 된다. 만약 명사 일치 과정에서 단어가 추출되지 못했다면 단어 검색과정을 사용한다.

검색은 입력 단어와 가장 유사한 단어를 찾아내는 것으로 TTS 결과가 좋지 않거나 화자의 발음이 좋지 않음으로 인해 초래된 결과를 좀 더 유연하게 대처할 수 있다. 접속사격 조사로 나누어진 입력 문장의 경우 위 세 과정을 거쳐 나온 결과가 최종 결과가 된다. 접속사격 조사로 나누어지지 않은 경우는 입력 데이터 추출 처리에서 이미 예외 단어 확인이 끝났기에 예외 처리를 하지 않고 단어의 개수에 따라 단어 검색 또는 조합 단어 추출 및 명사 부분 일치 추출을 거친다. 단어의 개수가 1개 이상이라면 단어의 입력 순서에 기준을 두어 조합될 수 있는 단어를 찾는 조합단의 추출과 명사 부분 일치 추출 과정을 통해 결과 단어를 추출한다. 이때 조합된 단어가 있다면 그룹 추출 과정을 실행하여 조합 단어와 조합된 단어의 일부를 차지하는 부모 단어리스트들의 상관관계를 정리한다. 단어 추출 과정을 통해 나온 결과와 다른 의미 결과 내에 존재하는 사투리를 표준어로 변환하고 그 결과를 최종 결과 및 최종 다른 의미 결과로 내보낸다.

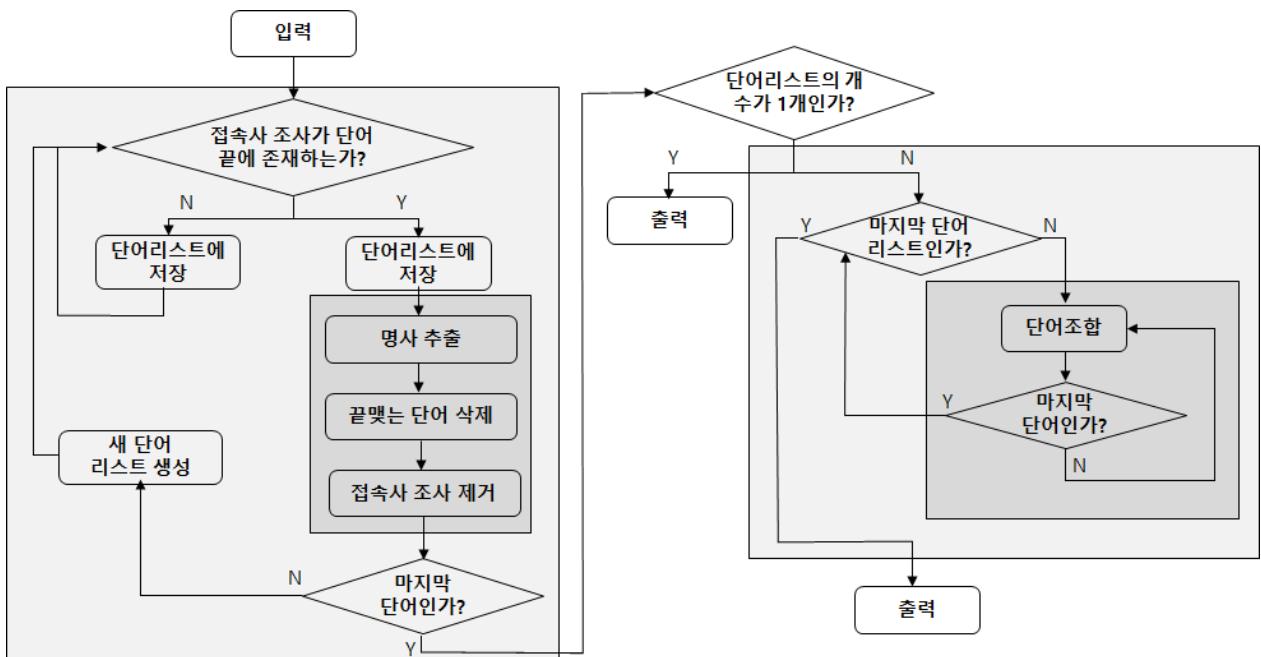


그림 5. 입력 문장 분류 및 가공 알고리즘
Fig. 5. Algorithm of sentence classification and processing

명사 부분 일치 검증은 부분 일치된 단어를 출력으로 내놓는 것이 아닌 true, false 값을 통하여 부분 일치하는 명사가 있는지 알려준다. 이때의 기준은 명사 부분 일치 추출과정과 같은 “입력 단어로 시작하거나 끝나는 명사가 존재하는가?”이며 명사 부분 일치 추출과는 다르게 부분 일치 하는지 검증하는 단계로 한 글자일 경우에도 부분 일치를 허용한다.

조합 단어 추출은 접속사격 조사로 나누어지지 않은 입력 문장일 경우에는 입력 문장을 나눌 기준이 없으므로 단어를 순서대로 조합하여 적절한 명사를 찾아내서 조합된 단어가 유효한 단어인지를 검사한다.

그룹 추출은 접속사격 조사로 나누어진 문장이 아니면서 단어의 개수가 1개 이상일 때 조합 단어 추출로 만들어진 조합 단어와 부모 단어 사이의 관계를 정리하기 위함이다.

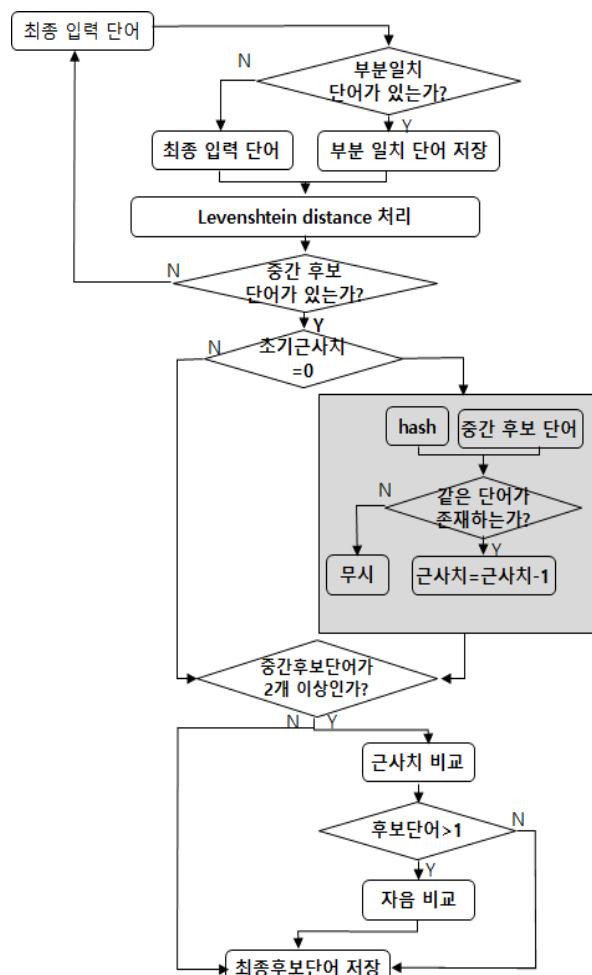


그림 6. 후보 단어 선택 알고리즘
Fig. 6. Algorithm of candidate word selection

부모 단어의 부분 일치 결과 생성 시 조합 단어와 부분 일치한 단어는 제외시키며 조합 단어의 부분 일치 생성 시 다른 제약 없이 생성된다. 단어 검색은 입력된 단어와 유사한 단어를 찾아낸 과정으로 그림 6과 같이 부분 일치 확인을 통해 최종 입력 단어와 부분 또는 모두 일치하는 단어를 찾아 초기 후보 단어를 생성한다.

최종 후보 단어 선출 방법은 전처리 때 만들어진 초기 근사치 값이 0일 경우에 Hashing과 Levenshtein distance algorithm의 결과로 나온 단어들을 매칭시켜 같은 단어가 있을 경우 그 단어의 근사치에 -1 값을 더해준다. 만약 0이 아닐 경우에는 Hashing algorithm을 사용하지 않으며 근사치 값을 그대로 저장한다. 그 후 근사치 비교 과정을 거쳐 후보 단어의 수를 줄이고 만약 후보 단어의 수가 2개 이상일 경우 자음 비교를 거쳐 최종 후보 단어를 선출한다.

Levenshtein distance algorithm은 표준어 테이블과 사투리 테이블에 존재하는 단어들 중 어떤 단어가 전처리를 거친 입력 단어와 얼마나 유사한지 검사하는 알고리즘으로 그림 7과 같이 후보 단어를 색출한다.

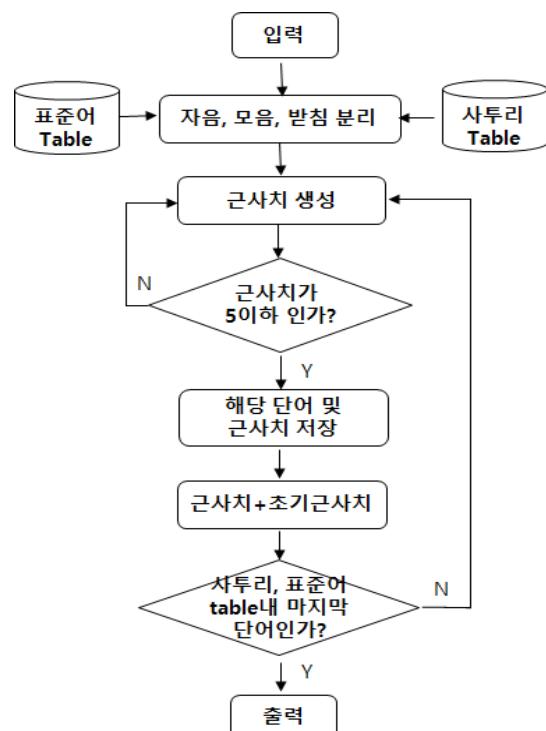


그림 7. Levenshtein distance 알고리즘
Fig. 7. Algorithm of levenshtein distance

입력받은 단어와 같아지기 위해서 몇 번의 글자 변경이 일어나야 하는지 기록한 값이 근사치이다. 근사치의 경우 Levenshtein distance algorithm을 통해 얻은 후보 단어의 근사치에 해당 최종 입력 단어의 초기 근사치 값을 더한다[9][10]. 현재 후보 선정에 있어서 근사치 값이 5이하일 경우에만 후보에 올린다.

Hashing algorithm은 그림 8과 같은 형태로 실행된다. 먼저 발음 유사도를 기준으로 만든 그림 9에서 보여주는 가중치 테이블을 기준으로 입력 단어의 Hash code를 구하여 표준어 테이블과 사투리 테이블에 존재하는 단어들의 Hash code값과 비교하여 같은 값을 가지는 단어를 색출한다. 단, 원래 입력 단어와 글자 수가 같을 경우이면서 숫자가 포함되지 않은 경우에만 실행한다.

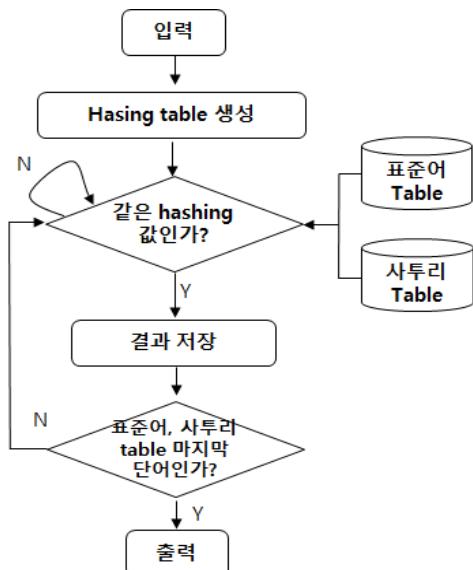


그림 8. Hashing 알고리즘

Fig. 8. Algorithm of Hashing

Hashing algorithm 처리로 나온 후보 단어와 Levenshtein distance algorithm에 나온 후보 단어 중 같은 단어가 있을 때 전처리를 거친 입력 단어와 글자 수가 같은 경우에만 그 단어의 근사치 값에 -1을 더해준다. 단, 원래 입력 단어와 글자 수가 같을 경우에만 실행한다.

3.4 근사치 비교를 통한 최종 후보 단어 선정

근사치 비교는 그림 10과 같이 각 후보 단어에 근사치를 비교하여 가장 적은 근사치 값을 가지는 후보 단어를 최종 결과로 저장한다. 이때 같은 근사치를 갖는 단어도 함께 저장한다.

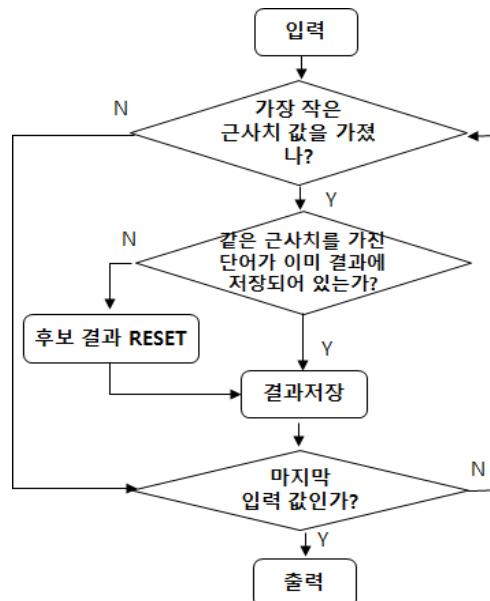


그림 10. 근사치 비교 알고리즘

Fig. 10. Algorithm of approximate comparison

2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
ㅂ	ㄷ	ㅅ	ㄱ	ㅈ	ㅎ	ㅁ	-	-	ㅏ	ㅓ	ㅗ	ㅜ	ㅡ	ㅣ	ㅐ	ㅔ	ㅣ	ㄷ	ㅅ	ㅈ	ㅂ	ㄱ	ㅁ	ㄹ	-	-	
ㅃ	ㄸ	ㅆ	ㄲ	ㅉ	ㅎ	ㅋ	ㄴ		ㅏ	ㅓ	ㅗ	ㅜ	ㅡ	ㅣ	ㅖ	ㅔ	ㅣ	ㅌ	ㅆ	ㅈ	ㅂ	ㄱ	ㅁ	ㄹ	ㄺ		
ㅍ	ㅌ	ㅎ	ㅋ	ㅊ	ㅌ	ㅍ	ㄴ		ㅏ	ㅓ	ㅗ	ㅜ	ㅡ	ㅣ	ㅖ	ㅔ	ㅣ	ㅎ	ㅆ	ㅈ	ㅂ	ㄱ	ㅁ	ㄹ	ㄺ		

그림 9. 가중치 테이블

Fig. 9. Table of weighted

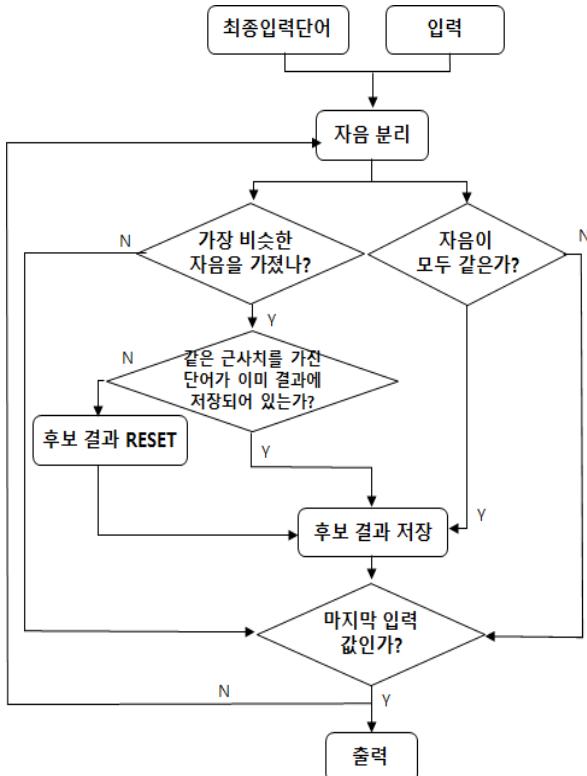


그림 11. 자음 비교 알고리즘

Fig. 11. Algorithm of consonant comparison

자음 비교는 근사치를 비교한 뒤에도 단어가 2개 이상일 경우 그림 11과 같이 최종 입력 단어의 자음 순서를 비교하여 가장 비슷한 단어를 최종 결과로 저장한다. 단, 원래 단어의 글자 수와 같지 않은 입력 단어가 후보 경우 자음의 순서가 모두 같아야만 최종 후보 결과에 저장된다.

최종 후보 단어의 개수가 2개 이상일 때 근사치 비교 과정과 같은 알고리즘으로 각 최종 후보 단어의 근사치를 비교하여 가장 적은 값을 가지는 단어를 최종 결과로 저장한다. 이때 같은 근사치를 갖는 단어도 저장한다. 그림 12는 문장 음성 인식 전체 구조도이다.

IV. 실험 결과

단어 및 문장 음성 인식 실험을 위해 그림 13과 같은 앱을 개발하여 테스트하였다. 실험 대상은 20대 여자 3명이 각각 3번씩 음성을 발화하면 음성 인식 프로그램을 통해 표준 테이블에 존재하는 단어들은 원본 단어 영역에 표시되고 인식된 단어는 음성 인식 결과 단어 영역에 출력된다.

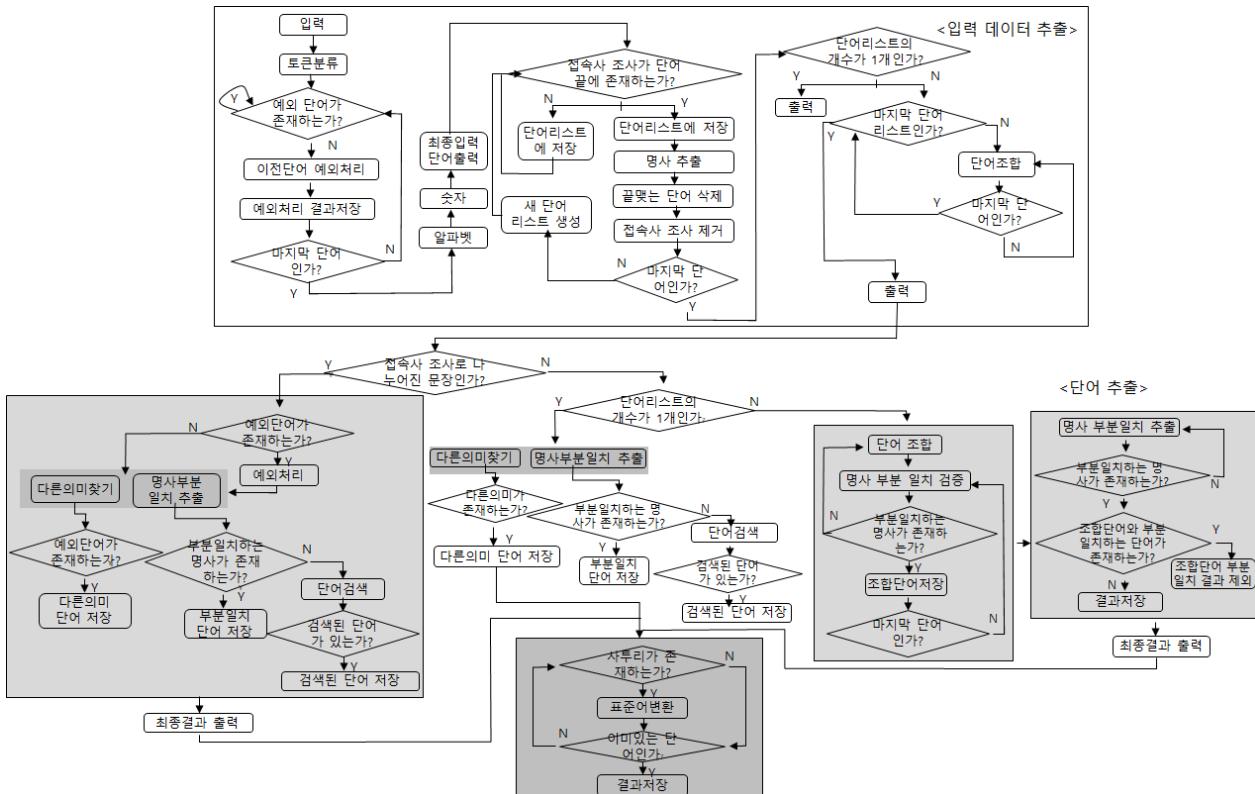


그림 12. 문장 음성 인식의 전체 구조

Fig. 12. Total structure diagram of sentence speech recognition

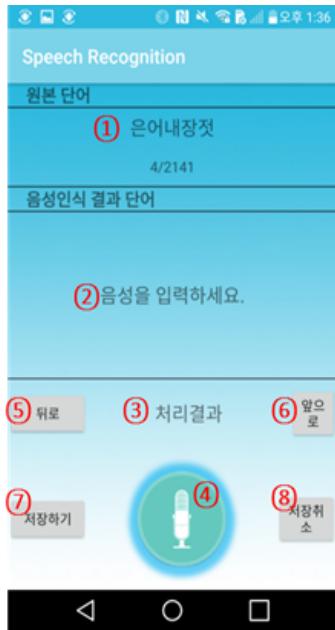


그림 13. 모바일 기반 성능 평가 앱

Fig. 13. Mobile based performance evaluation App

표 1. 표준어와 불일치된 단어

Table 1. Mismatches between standard and word

단어	결과	비고
나노	가나초콜릿->가나쵸콜릿	부분일치
딸기썬대	딸기선테이, 딸기, 선데이->딸기	
밤꼴밥	단팥빵(사전에 존재)	
	밤밥(사전에 존재)	
살치찌개	참치찌개(사전에 존재)	

표 1은 표준어와 불일치되는 단어 결과를 보여준다. 단순히 단어 명사만을 테스트 한 경우에 표준어는 1378개 중 2개가 인식을 못하여 0.1%를 사투리는

표 2. 실험 결과

Table 2. Result of experiment

테스트단어	SST결과	프로그램결과		테스트 단어의 표준어	비고
		선택된 입력값	결과		
가리찜	갈비찜 하고 먹었어	갈비찜	갈비찜	갈비찜	표준어일치
곤피	콤비랑 먹었어	콤비락	콜라, 콩밥	다시마	가중치
웨지의	외질	외질	엥지, 왜주, 어죽	오이지	가중치
편육채	현 육채	현, 육체		편육거자체	예외추가
가다리	갓 아리랑 먹었어	갓아리	가오리	가다랭이반건품	자음분류
멸구	내일 구랑 먹었어	내일구	매실주	머루	가중치
녹디발	똑띠 바라고 먹었어	똑띠, 바라고		숙주나물	예외추가
질굼	지금하고 먹었어	길금	길금	콩나물	사투리일치
...

818개 중 14개가 불일치하여 1.7% error rate를 보였다.

표 2는 인식률에 대한 근거인 표준어 일치와 불일치, 가중치 등에 대한 결과를 보여준다. 명사 뒤에 “~(으)랑 먹었다”의 테스트 결과는 표준어 1378개 중 4개가 인식을 못하여 0.2%, 사투리는 818개 중 32개가 불일치하여 3.9% error rate를 보였다. “~하고 먹었어”의 테스트 결과는 표준어 1378개 중 2개가 인식을 못하여 0.1%, 사투리는 818개 중 22개가 불일치하여 2.6% error rate를 보였다.

표 3은 현재 서비스 중인 API들과 비교한 결과를 보여준다.

표 3. 비교 결과

Table 3. Result of comparison

	Error rate	Hit rate
Cloud API	36.3%	63.7%
Clova API	16.6%	83.4%
Newtone	8.6%	91.4%
Proposal API	0.4%	99.6%
LSP	5.0%	95%

V. 결론 및 향후 과제

본 논문에서 한글 단어와 한글 문장 음성 인식 API를 구현하였다. 기존의 한글 음성 인식 시스템보다 표준어 인식률이 높았으며, 평균 95% 인식률을 보인 영어 음성 인식 시스템과 비교하여도 인식률의 높음을 확인하였다[11].

제안한 구조에서 한글 사투리는 대략 91.8%로 인식률을 보여 기존의 사투리 인식률보다 상대적으로 높음을 보였다[12]. 한글 단어는 대략 50만개 정도이며 이들 단어들로 완성된 문장의 수는 몇 천만 개로 구성될 수 있다고 국립국어원에서 발표하였다. 모든 단어와 문장을 테스트하기에는 제약이 따르기에 상대적으로 적은 약 2천개의 단어를 가지고 샘플 비교한 결과이다.

향후 과제로는 안정적으로 음성 인식을 하기위해 세 번 발화를 하여야하는 사용자의 불편한 점이 문제로 남는다. 3회에 걸친 발화 방식을 1회 발화로 음성을 정확하게 인식할 수 있는 방법을 개발하는 것을 목표로 두고 있다.

References

- [1] Kyung Nim Lee, "Speech language processing technology, how far", National Institute of Korean Language New Language Life, Vol. 27, No. 4, pp. 99-116, Dec. 2017.
- [2] <https://www.gartner.com/en/newsroom/press-releases/2019-01-09-gartner-predicts-25-percent-of-digital-workers-will-u>. [accessed: Aug. 30. 2019]
- [3] Byung gon Yang, "Google speech recognition of an English paragraph produced by college students in clear or casual speech styles", The Korean Society of Speech Sciences, Vol. 9, No. 4, pp. 43-50, Dec. 2017.
- [4] G. Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition", The IEEE Signal Processing Magazine, Vol. 29, No 6, pp. 18-27, Apr. 2012.
- [5] Dai-Hwan Lim, "Personal Authentication System Using Multimodal Biometric Algorithm", Journal of KIIT, Vol. 15, No. 12, pp. 147-156, Dec. 2017.
- [6] Seung Joo Choi and Jong-Bae Kim, "Comparison Analysis of Speech Recognition Open AIPIs' Accuracy", Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol. 7, No. 8, pp. 411-418, Aug. 2017.
- [7] Ki-mu Yoon and Wooil Kim, "Effective Recognition of Velopharyngeal Insufficiency (VPI) Patient's Speech Using DNN-HMM-based System", Journal of the Korea Institute of Information and Communication Engineering, Vol. 23, No. 1, pp. 33-38, Jan. 2019.
- [8] F.Hassan, M.R.A. Kotwal and M.N. Huda, "Bangla AST design by suppressing gender factor with gender-independent and gender-based HMM classifiers", World Congress on Information and Communication Technologies, pp. 1276-1281, Dec. 2011.
- [9] Jong-Sub Lee and Sang-Yeob Oh, "Vocabulary Retrieve System using Improve Levenshtein Distance algorithm", The Journal of Digital Policy & Management, Vol. 11, No. 11, pp. 367-372, Nov. 2013.
- [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. 156-162, Mar. 1976.
- [11] J. Onshaunjit and J. Srinonchat, "LSP Trajectory Analysis for Speech Recognition", Fifth International Conference on Computer Graphics, Imaging and Visualisation, CGIV, pp. 276-279, 2008.
- [12] Hee-Kyung Roh and Kang-Hee Lee, "A Basic Performance Evaluation of the Speech Recognition APP of Standard Language and Dialect using Google, Naver, and Daum KAKAO APIs", Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol. 7, No. 12, pp. 276-279, Dec. 2017.

저자소개

임 대 환 (Dai-Hwan Lim)



2002년 2월 : 한양대학교 정보
처리공학과(공학석사)
2014년 2월 : 한양대학교
정보통신공학과(공학박사)
2008년 9월 ~ 2014년 3월 : LG
전자연구원
2015년 3월 ~ 현재 : 서경대학교

컴퓨터공학과 조교수

관심분야 : 인공지능, 사물인터넷(IoT), 자율주행 시스템

남 기 훈 (Ki-Hun Nam)



2006년 2월 : 서경대학교
컴퓨터과학과(이학박사)
2006년 3월 ~ 2017년 2월 :
소프트웨어학과 겸임교수
2006년 8월 ~ 2009년 9월 :
한양대학교 디스플레이 연구소
연구원

2009년 10월 ~ 2011년 2월 : 충북대학교 BK21 연구원 및
초빙교수

2011년 3월 ~ 2017년 2월 : 서경대학교 컴퓨터공학과
초빙교수

2017년 3월 ~ 현재 : 서경대학교 컴퓨터공학과 조교수

관심분야: 인공지능, 사물인터넷(IoT), 임베디드 시스템