“헤이, 카카오!”를 불러야 하는 이유

카카오미니를 켜면, “이제 ‘헤이, 카카오!’라고 불러주세요”라는 인사말을 한다. 여느 음성 인식 스피커와 마찬가지로, 카카오미니에게 원하는 명령을 하려면 일단은 무언가 불러야 카카오미니가 귀를 기울이고 사용자의 명령을 들을 준비를 한다. “헤이, 카카오! 아이유 노래 틀어”와 같은 식이다.

‘헤이, 카카오!’와 같이 스마트 스피커를 부르는 단어를 호출 명령어(wake-up word) 또는 호출어라고 한다. 그냥 “야, 노래 좀 틀어봐”라고 하면 좋겠지만, 그러면 스피커는 이용자가 자신을 부르는 지를 인식하지 못한다. 조금 귀찮더라도 정해진 호출 명령어를 불러줘야 한다. 각 제품마다 여러 가지 호출 명령어가 있으며 사용자는 이들 중 하나를 선택해서 사용한다.

글 | 박종세 carlos.park@kakaocorp.com

어쩌다 보니 음성 관련 개발만 10년 넘게 했는데, 과거에 비해 요즘은 음성 기술이 정말 많이 대중화 되었다는 걸 실감합니다. 하지만 그래도 그 끝은 없는 분야인 것 같습니다.

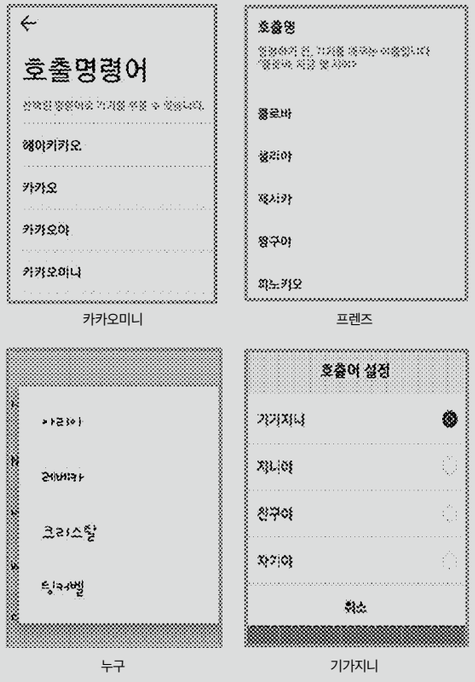
글 | 정대성 denzel.jung@kakaocorp.com

10여 년 간 음성 인식 연구와 서비스 개발을 하고 있습니다. 최근 인공지능 기술이 음성 인식에 접목됨으로써 인식 품질이 개선되고 다양한 서비스에 적용되어 재미와 보람을 느끼고 있습니다. 카카오의 모든 서비스에 음성 인식 기술이 적용될 때까지 열심히 일하는게 제 목표입니다.

호출 명령어는 어떻게 정해질까?

스피커마다 호출 명령어는 다르다. [그림 1]과 같이 국내에 출시된 인공지능 스피커는 그 이름도 많지만 이를 부를 때 사용하는 호출 명령어는 더 다양해서 헷갈릴 정도다.

[그림 1] 국내 인공지능 스피커의 호출 명령어 종류



이렇게 다양한 호출 명령어는 기술적인 측면과 사업적 판단이 함께 고려되어 선정된 결과다.

일반적으로 호출 명령어로는 3음절에서 5음절의 단어가 적당하다. 너무 짧으면 스마트스피커가 호출 명령어인지 아닌지를 식별하기 어렵고, 너무 길면 이용자가 말하기 힘들다. 단어의 끝이 /ㅏ/, /ㅑ/ 모음인 명령어는 이용자가 발음하기 쉬울 뿐만 아니라, 해당 단어의 음성은 또렷하기에 스마트스피커가 알아듣기도 용이하다. 반면에 /ㅜ/, /ㅡ/ 모음으로 끝나는 단어는 음성의 크기가 작아서 적절하지 않다. 또한 /ㅋ/, /ㅌ/, /ㅍ/, /ㅅ/ 과 같이 구분이 잘 되는 발음을 만들어 내는 단어가 호출 명령어로 적합하다.

이러한 기술적 특성과 함께 기업명이나 플랫폼 명칭 등을 포함하거나 브랜드 아이덴티티에 대한 고려가 종합적으로 이뤄져, 호출 명령어가 선정된다. 결국 사용자가 부르기 쉽고 회사의 이미지에 어울리는 단어가 선정되는 것이다.

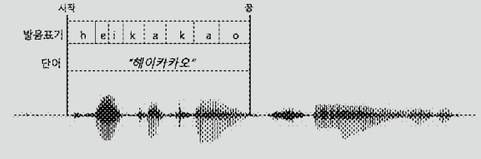
호출 명령어는 단순히 스피커를 부르는 용도로만 쓰이는 게 아니라 그 자체로 브랜드가 되어가고 있다. 아마존의 알렉사(Alexa)는 완전히 플랫폼 명칭이 되었고, 구글 또한 최근 구글 어시스턴트가 아닌 헤이 구글(Hey Google)이란 호출 명령어를 스마트 스피커의 브랜딩에 적극 활용하고 있다. 호출 명령어가 상품명보다 더 기억하기 쉽고 사람들에게 친숙하기 때문이다.

[그림 2] CES 2018에 전시된 Hey Google 광고들

카카오미니는 어떻게 “헤이, 카카오!”를 인식할까?

그렇다면 카카오미니는 일상 대화와 호출 명령어를 어떻게 구분할까? 스피커가 호출 명령어를 알아듣게 하기 위해서 핵심어 검출(keyword spotting)[1] 기반의 음성 인식 기술을 사용한다. 이는 사람의 음성을 계속 듣고 있다가 특정 키워드가 발성 되었는지를 검출하는 방법이다. 예를 들어 “헤이, 카카오!, 신나는 노래 틀어”라고 말하면, [그림 3]과 같이 연속으로 입력되는 음성 구간에서 키워드(헤이, 카카오!)에 해당하는 발음 시퀀스(sequence)가 순차적으로 들어오는지를 지속적으로 감지한다.

[그림 3] 음성 구간에 대한 핵심어 검출 결과



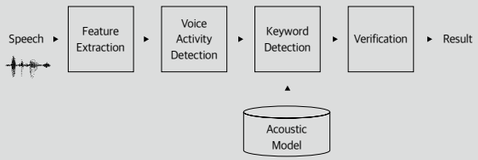
발화자의 음성에서 핵심어 검출하는 방법

1) 발화자 음성에서 특징 추출

앞서의 과정을 조금 더 구체적으로 살펴보자. 핵심어 검출은 [그림 4]와 같은 순서로 동작한다. 우선 발화자의 음성(raw speech)에서 특징 벡터(feature)를 추출한다. 특징 벡터는 음성의 특성을 잘 반영하고 편리한 계산을 위해 사용된다. 일반적으로 사용되는 특징 벡터에는 멜 주파수 캡스트럼(Mel frequency cepstral coefficients,

MFCC)이나 필터 뱅크 에너지(filter bank energy) 또는 지각 선형 예측(perceptual linear prediction, PLP) 등이 있다.

[그림 4] 핵심어 검출 구조도



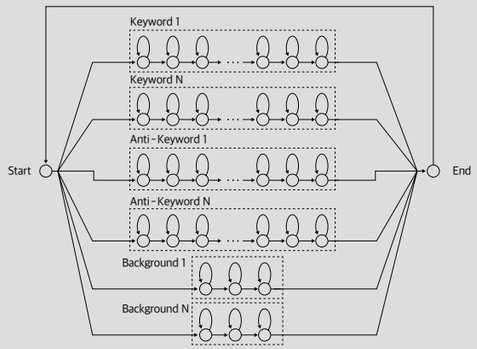
2) 음향 모델을 활용한 키워드 판별

현재 음성 신호가 들어오는지를 확인하고, 해당 음성 구간이 키워드인지 아닌지를 판별한다. 키워드 판별을 위해서는 미리 구성한 음향 모델(acoustic model)을 사용한다. 음향 모델은 많은 사람의 목소리가 저장된 음성 데이터베이스를 이용하며, 원하는 기계 학습(machine learning)을 선택하여 구축한다.

일반적으로 음향 모델은 음소(phoneme) 단위를 기반으로 한 은닉 마르코프 모델(hidden Markov model, HMM) 형태로 구성된다. 음소는 우리말의 자음, 모음과 유사한 개념으로 소리의 기본 단위를 말한다. 음소 단위 HMM 모델에서는 음소를 1~3개의 상태(state)로 나누며 이 모델은 각 상태의 연결 확률(transition probability)과 관측 확률(observation probability)로 구성된다. 과거에는 음성 인식 분야에서 관측 확률값을 계산할 때 가우시안 혼합 모델(Gaussian mixture model, GMM)을 많이 사용하였으나, 최근에는 깊은 신경망(deep neural network, DNN)을 적용하여 상당한 성능 향상을 보이고 있다.

핵심어 검출을 위한 음성 인식 네트워크는 [그림5]와 같이 키워드 모델, 안티키워드(anti-keyword) 모델 그리고 백그라운드(background) 모델과 이들의 연결 구조로 이루어진다. 시작 노드(node)에서 시작해서 각각의 모델 경로를 지나갈 수 있으며, 음성이 지속적으로 입력되기 때문에 종료 노드에서 다시 시작 노드로 되돌아갈 수 있다. 여기서 키워드 모델은 [그림 3]과 같은 키워드 단어의 발음열에 해당하는 음소 단위 HMM을 연결한 것이다. 백그라운드 모델(또는 garbage 모델)은 키워드가 아닌 모든 음성에 대응 가능한 모델이며 일반적으로 각각의 음소 모델을 활용한다. 음소는 자음이나 모음과 같이 음성의 발음을 표현하는 기본 단위로 만약 N개의 음소를 정의한다면, [그림 5]와 같이 N개의 백그라운드 음소 모델을 구성할 수 있다. 안티키워드 모델은 키워드와 유사한 발음을 가진 단어에 대응 가능한 모델이다. 만약 키워드가 '카카오'라면 이와 유사하게 들릴 수 있는 단어를 안티키워드로 등록하여 잘못된 인식을 방지할 수 있다.

[그림 5] 핵심어 검출을 위한 음성 인식 네트워크 구성

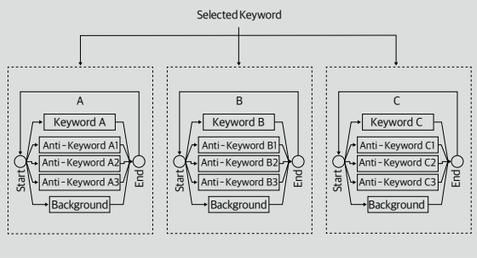


카카오미니의 핵심어 검출 엔진

카카오미니의 핵심어 검출 엔진은 안티키워드 모델과 DNN 기법을 적용하여 높은 인식률을 가진다. 저전력, 빠른 응답 그리고 적은 메모리 사용을 위한 알고리즘 최적화 과정이 수행되었다.

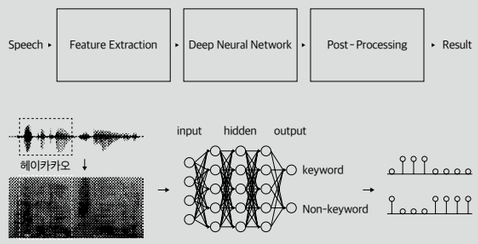
대부분의 인공지능 스피커는 몇 개의 호출 명령어 중에서 하나를 선택하는 방식을 사용한다. 이 경우 실제로 동작하는 키워드는 한 개가 되며, [그림 6]처럼 싱글키워드(single keyword)를 검출하는 형태로 동작한다. 만약 사용자가 호출 명령어로 '키워드 A'를 선택하면 그에 대응되는 안티키워드 A1, 안티키워드 A2 그리고 안티키워드 A3이 함께 적용되며, 호출 명령어를 변경하면 그에 맞는 안티키워드 목록도 함께 변경된다.

[그림 6] 하나의 호출 명령어를 선택하는 싱글키워드 방식



앞에서 음소 기반 음성 인식 네트워크를 구성하는 방법에 대해 살펴보았는데, 음성 인식 네트워크를 구성하지 않고 더 간단하게 핵심어를 검출하는 방법[*2]도 있다. [그림 7]은 DNN을 이용한 핵심어 검출 시스템을 보여준다. 입력되는 음성에서 10msec(millisecond) 단위의 프레임 구간마다 특징 벡터를 추출한 뒤 이를 바로 DNN 입력(input)으로 넣는다. 이때 특징 벡터는 앞뒤로 여러 프레임의 값을 합친 슈퍼 벡터(super vector) 형태로 구성한다. 그리고 입력된 특징 벡터는 미리 학습한 DNN의 계산을 통해 keyword 노드와 non-keyword 노드의 확률 값으로 변환된다. 프레임마다 발생하는 이 출력 값에 대해 후처리 과정을 거치면 키워드 검출이 가능하다. 이 방법은 백그라운드 모델을 설계할 필요가 없고 구조가 간단하다는 장점이 있다. 또한 오류가 발생한 음성 데이터를 재학습할 수도 있다. 하지만 학습 데이터를 음소 단위로 쪼개어 사용하지 않기 때문에 키워드 전체가 포함된 학습용 음성 데이터가 많이 필요하다.

[그림 7] DNN을 이용한 핵심어 검출 시스템

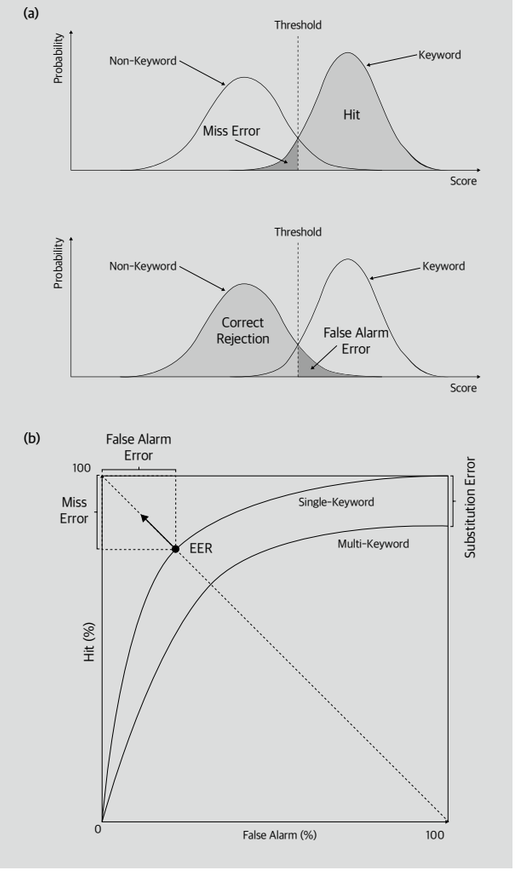


인공지능 스피커의 음성 인식은 아직 완벽하지는 않지만 더욱 기술이 발전하면 지금보다 더 좋아질 것이다.

현재의 한계, 그리고 발전을 위한 노력

[그림 8] (a)는 핵심어 검출 동작 시 발생하는 오류 유형을 그림으로 표현한 것이다. 키워드를 정확히 말했을 때 핵심어 검출 프로그램의 스코어가 임계치(threshold)보다 크면 검출 성공(hit)이 되고, 스코어가 낮으면 검출 실패(miss)가 된다. 반대로 키워드가 아닌 말을 한 경우에는 스코어가 낮게 나타나 정확히 거절(rejection)이 되어야 한다. 그러나 가끔씩 키워드와 유사한 발음이 포함되면 스코어가 높게 나타나 잘못 검출되는 상황(false alarm error)이 발생한다. 이때 임계치를 크게 하면 잘못 검출될 확률(false alarm rate)은 줄어들지만 성공 확률(hit rate)도 낮아지고, 반대로 임계치를 작게 하면 성공 확률이 높아지지만 잘못 검출될 확률도 함께 커지게 된다.

[그림 8] (a) 핵심어 검출 오류의 유형 (b) 핵심어 검출 시스템 ROC 커브



[그림 8] (b)는 핵심어 검출 시스템의 특성을 살펴보기 위해 많이 사용되는 수신자 조작 특성(receiver operating characteristic, ROC) 곡선이다. 곡선 상에서 성공 확률이 증가할수록 잘못 검출될 확률도 함께 증가하는 것을 볼 수 있다. 이때 성공 확률과 잘못 검출될 확률이 같아지는 지점을 동일 오류율(equal error rate, EER)이라고 하며 시스템의 성능 지표로 많이 사용한다. 개발자의 궁극적인 목표는 ROC 곡선에서 EER이 낮아지는 방향으로 이동하여 0으로 수렴하게 하는 것이다. 그러나 실제로는 사용자의 목소리 차이, 발성 패턴 차이, 주변 잡음 등 여러 가지 성능 저하 요소가 존재한다. 카카오의 개발자들은 사용자의 음성을 더 똑똑하게 잘 알아듣는 카카오미니를 만들기 위해 더욱 정교하고 진보된 음성 인식 알고리즘을 연구하고 많은 실험 과정을 거치고 있다.

음성 인터페이스로서 카카오미니의 발전 방향들

지금보다 더 쉽고 편리하게 카카오미니를 사용하는 방법은 없을까? 카카오미니의 서비스가 다양해짐에 따라 음성 인터페이스에서의 새로운 요구 사항도 많아질 것이다. 여러 가지 발전 방향들 중에서 몇 가지를 꼽아보면 개인화, 멀티 플랫폼, 멀티 모달, 상황 인지가 있다.

개인화(personalization)

사람의 외모나 성격, 취향 등이 모두 다른 것처럼 사용자마다 목소리 특성, 사용 방법, 선호하는 호출 명령어도 다르다. 이러한 개인의 차이를 반영하는 것도 중요한 과제다. 인공지능 스피커에 입력된 사용자의 목소리를 스피커 내에서 바로 학습해서 해당 사용자의 호출 명령어를 더 잘 인식하도록 하는 방법도 많이 연구되고 있다.*3

보통 카카오미니를 집에서 사용하기 때문에 본인뿐 아니라 가족들도 함께 쓰는 경우 카카오톡 메시지나 일정을 확인할 때 보안 문제가 생길 수 있다. 따라서 사용자의 목소리에만 반응하게 하는 화자인식 기술이 요구된다.*4 가족 중 누구의 목소리인지 구분되면 이를 통해 인증이나 추천 등의 개인별 맞춤 서비스가 가능하다. 이렇게 화자인식을 적용할 때 매번 부르는 호출 명령어를 활용하는 것이 도움이 된다. 사용자가 미리 본인 목소리로 “헤이, 카카오!”를 등록하면, 이후에도 항상 해당 호출 명령어를 말하기 때문에 해당 사용자의 목소리가 맞는지 판단하기 쉬워진다.

또한 지금은 미리 정해진 몇 개의 호출 명령어 중에서 하나를 선택하는 방식을 사용하지만, 사용자는 자신이 호출 명령어를 직접 정하고 싶어한다. 이 기능이 실제로 가능해 진다면 작명의 재미가 생기는 것이다. 어쩌면 “개똥아”, “돌머리”처럼 짓궂은 단어를 많이 입력할지도 모르겠다. 다만 사용자가 설정한 호출 명령어가 음성 인식 기술로 검출하기 어렵거나 오인식을 많이 유발하는 경우에 대한 대책도 반드시 필요하다. 해당 단어를 인식하는 데 문제가 없는지 개발자가 미리 테스트해 볼 수 없기 때문이다.

멀티 플랫폼(multi-platform)

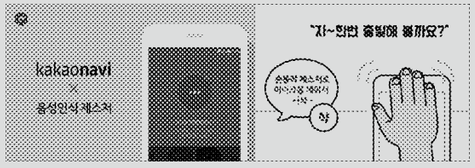
최근 인공지능 스피커에 많은 기능과 서비스가 탑재되다 보니 하나의 플랫폼에서 모든 서비스를 처리하는 게 쉽지 않다. 사용자가 환경에 따라 각각 다른 음악 서비스나 스마트홈 서비스를 이용하기 때문이다. 또한 각 업체의 인공지능 플랫폼이 잘 하는 영역도 모두 다르다. 이런 상황에서 하나의 플랫폼보다는 여러 플랫폼이 연동하는 방식도 고려할 수 있다. 애플 시리(Siri)가 수학 연산이나 질의응답 도메인에 대해서 또 다른 인공지능 플랫폼인 울프럼 알파(Wolfram Alpha)*5를 연동하여 답변해주는 것은 오래된 사례다. 또 다른 방식으로 스마트폰이나 가전 제어는 빅스비(Bixby)를 이용하고, 카카오톡 보내기, 음악 재생, 길찾기 등은 카카오I(아이)를 이용하는 경우도 상상해 볼 수 있다. 이때 사용자는 호출 명령어를 “하이, 빅스비”와 “헤이, 카카오!”를 선택해서 부를 수 있을 것이다. 또한 똑같은 명령어라도 호출 명령어에 따라 다르게 동작하도록 설정할 수도 있다.

멀티 모달(multi-modal)

음성 외에 버튼, 센서, 카메라 등 다른 입력 수단을 이용하여 스마트 기기를 웨이크업 하거나 이를 음성 인식과 함께 사용하는 방식도 많다. 손목을 위로 들면 스마트 워치에 탑재된 음성 비서가 바로 실행되거나, 집안에 있는 카메라를 똑바로 쳐다보기만 하면 사용자의 얼굴을 인식하여 “장동건 님, 무엇을 도와드릴까요?”라고 물어볼 수도 있다.

현재 카카오의 길 찾기 어플리케이션인 카카오내비에는 음성 인식 기능이 탑재돼 있어서 목적지를 음성으로 검색할 수 있다. [그림 9]처럼 근접 센서를 이용하여 스마트폰 근처에 손을 가져가면 바로 음성 인식이 동작한다. 이것도 제스쳐를 이용한 멀티모달의 한 가지 적용 예다.*6

[그림 9] 카카오내비 음성 인식 제스처



상황 인지(Context Aware)

이심전심(以心傳心)이라는 말처럼 카카오미니가 내 마음을 알아주면 참 편리할 것이다. 말하지 않아도 외출할 때면 전등을 꺼주고, 요리할 때는 타이머를 설정해주고, 잠이 들면 음악도 꺼지게 하는 것처럼 말이다. 이처럼 눈치 100단 수준의 카카오미니라면 목 아프게 “헤이, 카카오!”를 부르지 않아도 되지 않을까. 쉽지는 않겠지만 앞으로 조금씩은 눈치가 생기지 않을까 싶다. 지금의 카카오미니는 사용자 명령에 대한 답변이 끝나고 나면, 호출 명령어를 부르지 않고도 연속으로 명령할 수 있도록 설계되어 있다. 이는 사용자가 연달아서 질문하는 경우가 많다는 점을 고려한 음성 인터페이스이다.

*1 논문 | Rose,R. & Paul, D.(1990). A hidden Markov model based keyword recognition system, Proceedings of the International Conference on Acoustics, Speech and Signal Processing. , 논문 | Wilpon, J., Rabiner, L., Lee, C. and Goldman, E. (1990). Automatic recognition of keywords in unconstrained speech using hidden Markov models, IEEE Transactions on Acoustics, Speech, and Signal Processing. *2 논문 | Chen, G., Parada, C. and Heigold, G. (2014) Small-footprint Keyword Spotting using Deep Neural Networks, Proceedings of the International Conference on Acoustics, Speech and Signal Processing *3 논문 | Kumatani, K., et al. (2017). Direct Modeling of Raw Audio with DNNs for Wake Word Detection, Automatic Speech Recognition and Understanding Workshop *4 참고 | 김명재. (2017). 카카오미니는 어떻게 목소리를 인식할까, 카카오AI리포트 *5 참고 | 스티븐 울프럼이 만든 Knowledge Engine (www.wolframalpha.com) *6 참고 | 카카오 기업 블로그 blog.kakaocorp.co.kr/654