

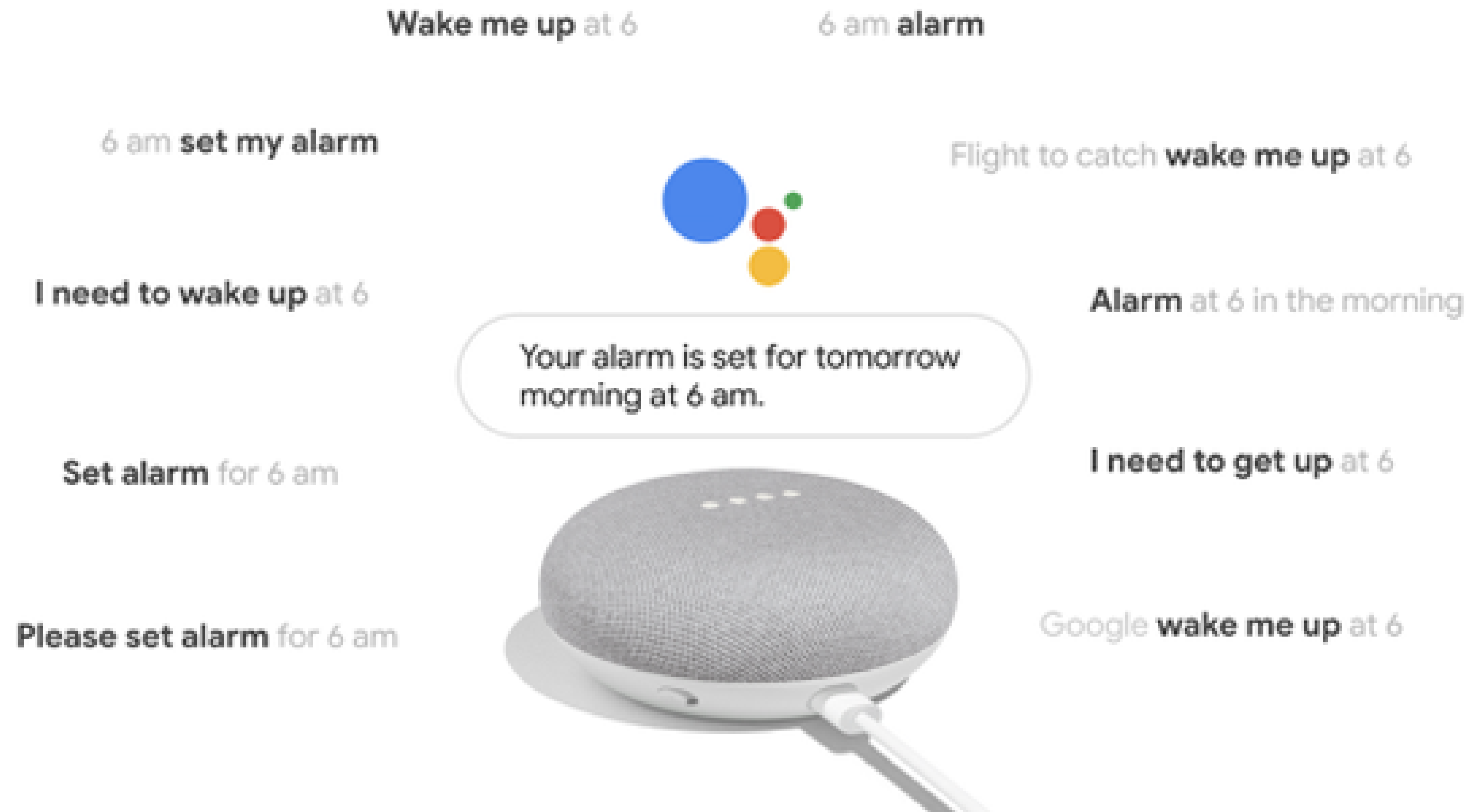
# Title here

Semititle here



# [ 인공지능 비서 ]

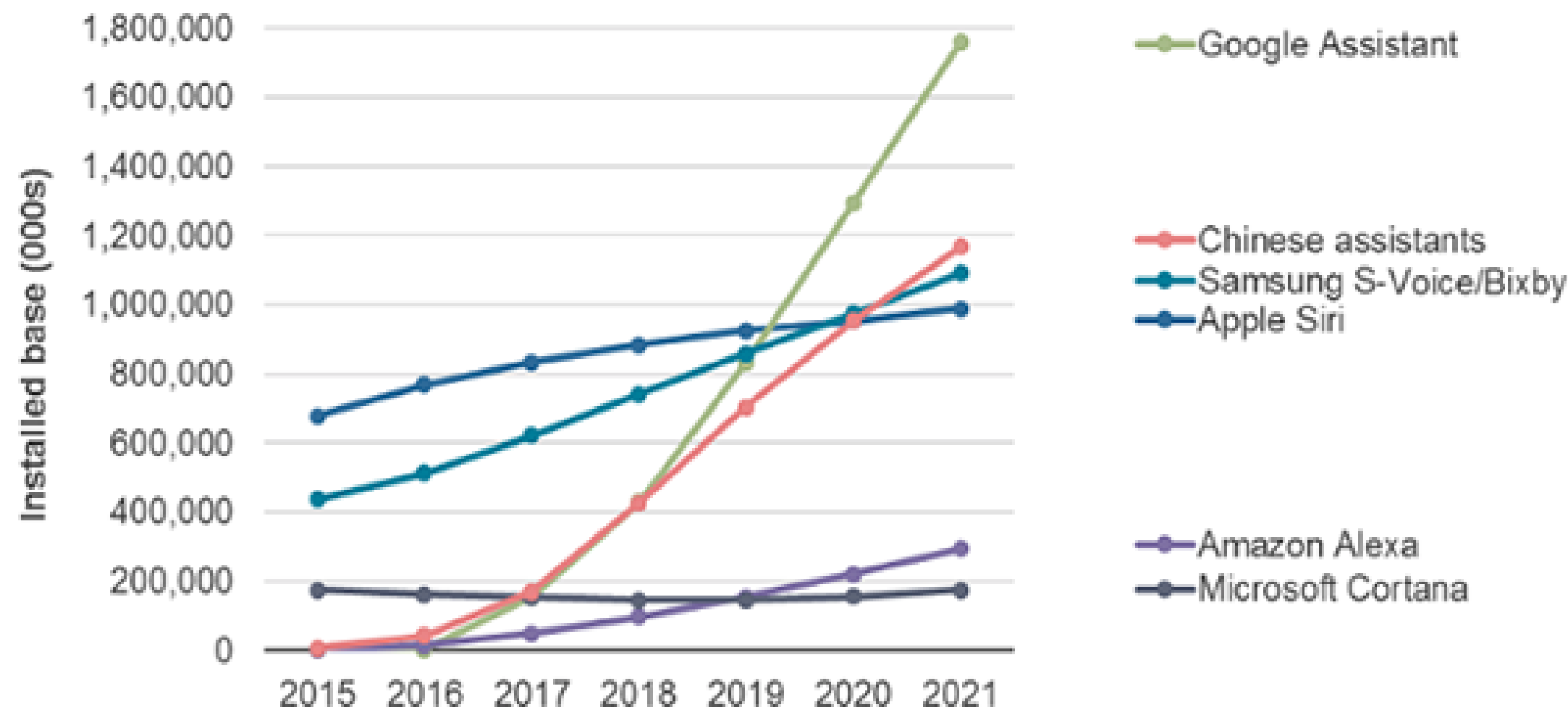
20211727 김성민  
20211728 김연수  
20213062 김성윤  
20213066 김지수



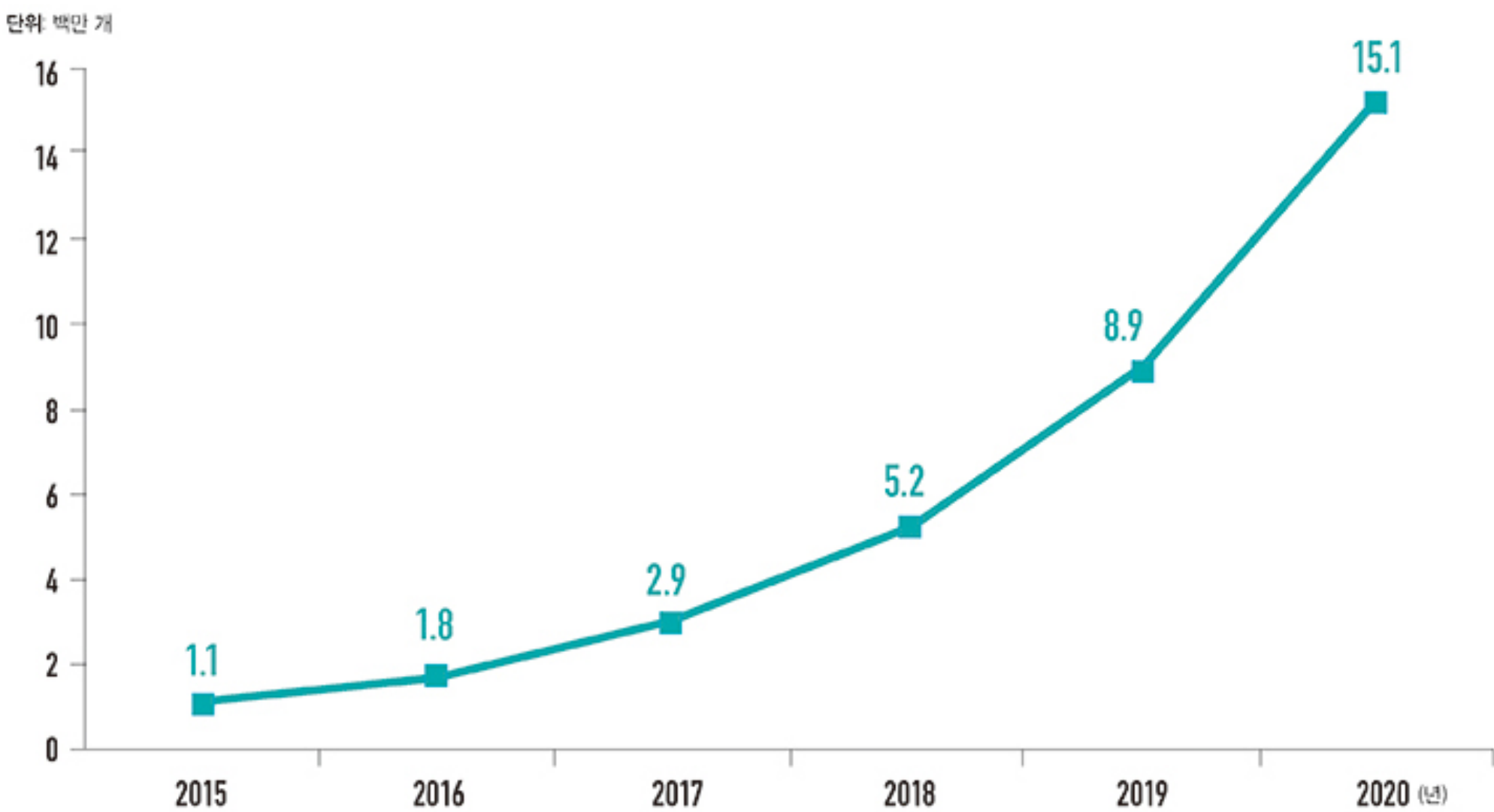
# 음성인식 AI 비서 추세

ai 비서 시장은 스마트폰과 같은 스마트 디바이스의 기술 발전과 함께 폭발적으로 성장 중이다

가상비서와 음성 AI 지원 디바이스 추이



음성인식 AI비서 장치의 세계 판매량




# 구글 어시스턴트 기능


음악 검색, 알람 맞추기, 기본 대화, 주차 위치 찾기, 날씨 검색 등등



안녕하세요? 무엇을 도와드릴까요?

자동차 어디 주차해놨어?

 You parked here · Updated 1hr ago



Based on your location history




안녕하세요? 무엇을 도와드릴까요?

이 음악 뭐야?

Young Lover

Song by St. Vincent




young lover, begging you please to wake up  
Young lover, I wish that I was your drug  
(No I) young lover, I miss the taste of your tongue  
(No I) young lover, I wish yo... [Full lyrics on Google Play](#)


Artist: [St. Vincent](#)


Album: [Masseduction](#)


Released: 2017

Genre: Alternative/Indie

 YouTube

 Spotify

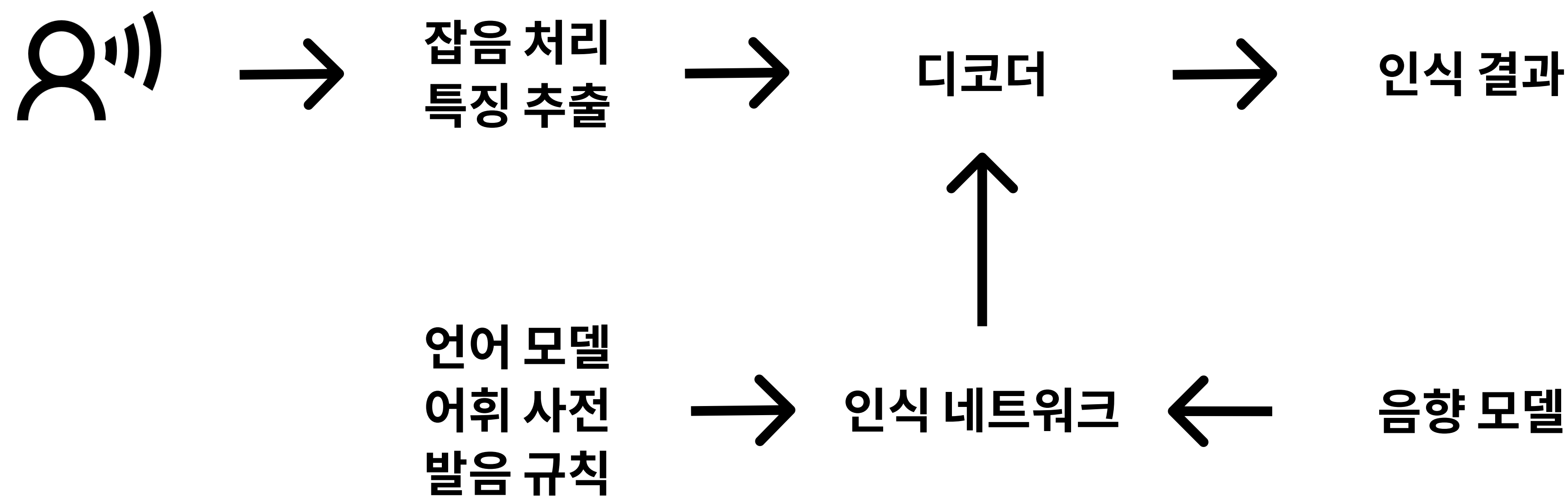
 Get on...

 Search



# 음성인식 기본 구조

음성 인식 후 입력 출력 과정



# 관련 알고리즘-자연어 처리

텍스트와 말을 이해할 수 있는 능력을 컴퓨터에 제공

```
%%time
nouns = noun_extractor.train_extract(
    sentences,
    min_noun_score=0.3,
    min_noun_frequency=20
)
nouns['뉴스']
words = [ '박근혜', '우병우', '민정수석', '트와이스', '아이오아이' ]
for word in words:
    print('%s is noun? %r' % (word, word in nouns))

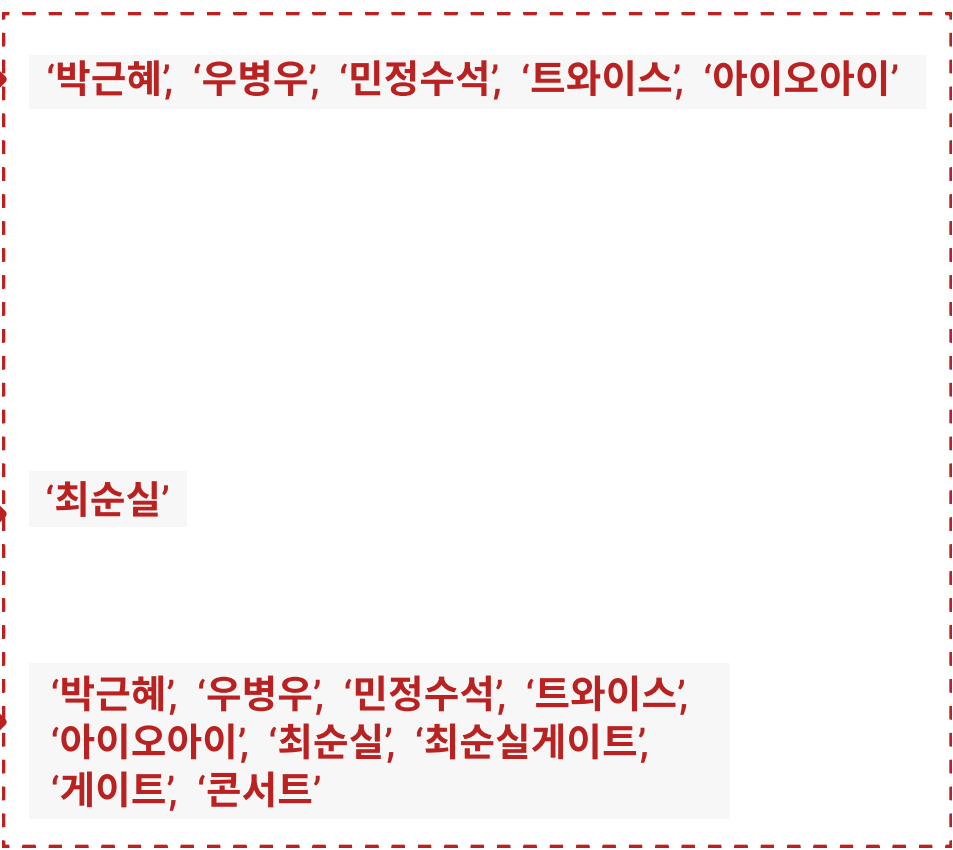
박근혜 is noun? True
우병우 is noun? True
민정수석 is noun? True
트와이스 is noun? False
아이오아이 is noun? True

noun_extractor.is_noun('최순실', min_noun_score=0.3)
True

words = [ '박근혜', '우병우', '민정수석', '트와이스',
          '아이오아이', '최순실', '최순실게이트',
          '게이트', '콘서트' ]
for word in words:
    noun_score, known_r_ratio = noun_extractor.predict(word)
    print('%s : score=%.3f' % (word, noun_score))

박근혜 : score=0.538
우병우 : score=0.757
민정수석 : score=0.834
트와이스 : score=0.262
아이오아이 : score=0.980
최순실 : score=0.828
최순실게이트 : score=0.320
게이트 : score=0.745
콘서트 : score=0.769
```

## 품사 태깅



```
top100 = sorted(nouns.items(),
    key=lambda x:-x[1].frequency * x[1].score)[:100]

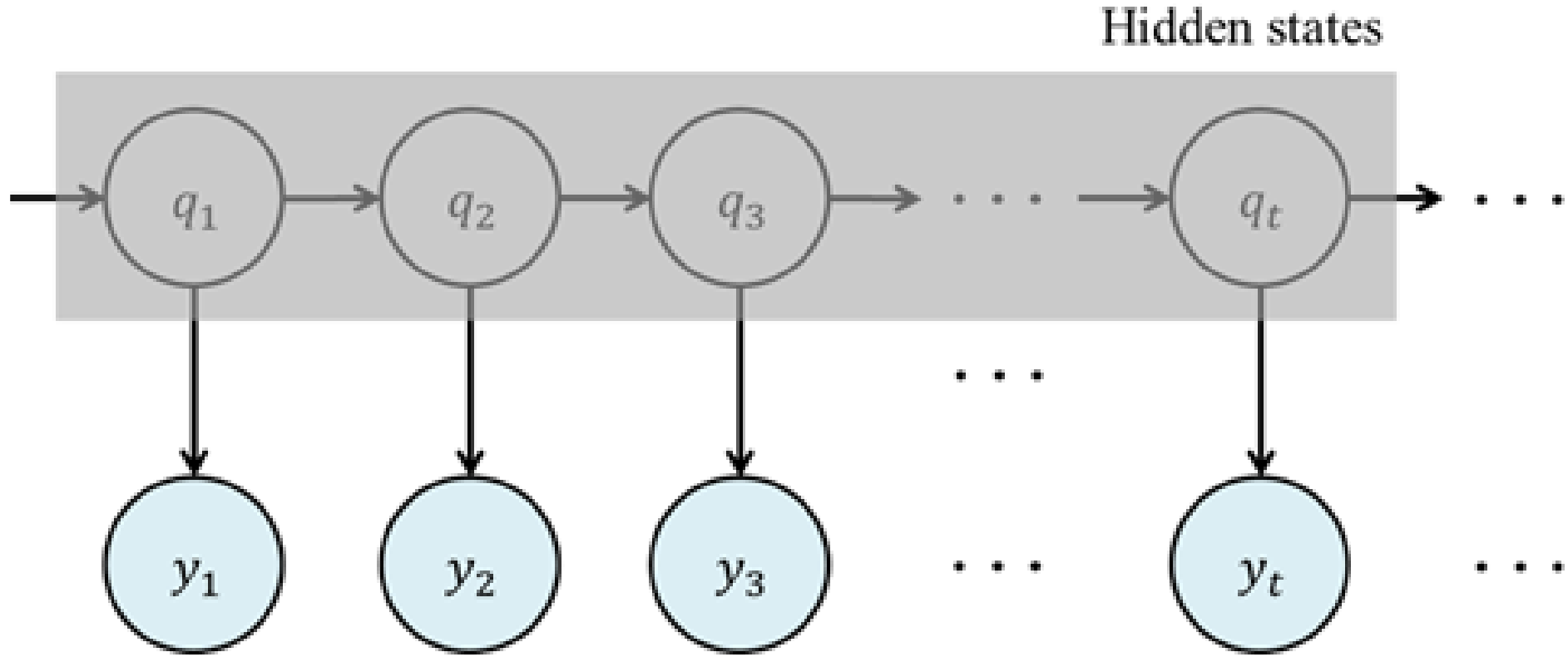
for i, (word, score) in enumerate(top100):
    if i % 5 == 0:
        print()
    print('%6s (%.2f)' % (word, score.score), end='')

무단 (1.00) 재배포 (1.00) 금지 (0.99) 20일 (0.91) 기자 (0.67)
지난 (0.99) 뉴시스 (1.00) 함께 (0.97) 오후 (0.99) 저작권자 (1.00)
진행 (0.92) 때문 (1.00) 이번 (0.99) 관련 (0.96) 참석 (0.98)
이라고 (0.99) 제공 (0.96) 오전 (1.00) 사진 (0.40) 미국 (0.77)
19일 (0.88) 최근 (0.98) 공개 (0.98) 같은 (1.00) 이날 (0.69)
방송 (0.80) 라고 (1.00) 시작 (0.96) 이후 (0.98) 가장 (0.97)
제보 (1.00) 하지 (1.00) 사랑 (0.91) 다양 (1.00) 국내 (0.97)
자신 (0.90) 뉴스 (0.44) 이상 (0.86) 세계 (0.98) 하고 (0.80)
있다고 (1.00) 한국 (0.78) 모습 (0.73) 한편 (0.98) 북한 (0.80)
공감 (0.69) 국회 (0.95) 확인 (0.95) 연합뉴스 (0.69) 당시 (0.99)
조사 (0.93) 설명 (0.95) 발표 (0.95) 계획 (0.89) 상황 (0.83)
이에 (1.00) 생각 (0.81) 모두 (0.99) 기업 (0.82) 대표 (0.39)
기록 (0.91) 의혹 (0.83) 필요 (0.97) 판단 (0.97) 문제 (0.68)
경우 (0.66) 운영 (0.89) 대통령 (0.49) 우리 (0.53) 없는 (1.00)
현재 (0.65) 이런 (1.00) 중국 (0.70) 이들 (0.92) 가능 (0.99)
영상 (0.54) 11월 (0.97) 발생 (0.97) 시장 (0.67) 결과 (0.85)
뉴스1 (0.94) 지역 (0.69) 경제 (0.92) 출연 (0.95) 사람 (0.52)
기대 (0.95) 마련 (0.96) 사용 (0.85) 현장 (0.87) 관계자 (0.68)
마음 (0.91) 처음 (0.99) 브랜드 (0.92) 정부 (0.57) 남자 (0.87)
글로벌 (1.00) 반영 (1.00) 대비 (0.99) 주장 (0.88) 사람들 (0.95)
```

△ 단어 최소빈도수를 점수로 산출하여 0.3점 이상,  
빈도 수 100회 이상인 단어들에서 명사를 추출

# 관련 알고리즘-HMM(Hidden Markov Model)

어떠한 현상의 변화를 확률 모델로 표현한 것에 은닉된 state와 직접적으로 확인 가능한 observation을 추가하여 확장한 것



△ HMM으로 풀고자 하는 문제는 관측 가능한  $y_t$ (observation)뿐이고,  $y_t$ 는  $q_t$ (은닉 state)에 종속적으로 발생한다고 할 때,  $y_t$ 의 사건을 통해  $q_t$ 를 추론하는 것

- 주어진 observation이 나타날 확률 계산

```
Algorithm 1: Forward algorithm
Input : initial probabilities  $\pi$ ,
        transition probabilities  $T$ ,
        emission probabilities  $E$ ,
        observation  $O = o_1, o_2, \dots, o_T$ 
Output: likelihood  $p(O|\pi, T, E)$ 
1 for  $j = 1 : N$  do
2    $\alpha_1(q_j) \leftarrow p(q_j)p(o_1|q_j)$ 
3 end
4 for  $t = 2 : T$  do
5   for  $j = 1 : N$  do
6      $\alpha_t(q_j) \leftarrow \sum_{i=1}^N \alpha_{t-1}(q_i)p(q_j|q_i)p(o_t|q_j)$ 
7   end
8 end
9  $p(O|\pi, T, E) \leftarrow \sum_{j=1}^N \alpha_T(q_j)$ 
10 return  $p(O|\pi, T, E)$ 
```

- 주어진 observation이 나타날 확률이 가장 큰 state 계산

```
Algorithm 2: Viterbi algorithm
Input : initial probabilities  $\pi$ ,
        transition probabilities  $T$ ,
        emission probabilities  $E$ ,
        observation  $O = o_1, o_2, \dots, o_T$ 
Output:  $s_1^*, s_2^*, \dots, s_T^*$ 
1 for  $j = 1 : N$  do
2    $v_1(q_j) = p(q_j)p(o_1|q_j)$ 
3 end
4  $s_1^* \leftarrow \arg \max_{q \in Q} p(q_j)p(o_1|q_j)$ 
5 for  $t = 2 : T$  do
6   for  $j = 1 : N$  do
7      $v_t(q_j) \leftarrow \max_{q \in Q} v_{t-1}(q)p(q_j|q)p(o_t|q_j)$ 
8      $s_t^* \leftarrow \arg \max_{q \in Q} v_{t-1}(q)p(q_j|q)p(o_t|q_j)$ 
9   end
10 end
11 return  $s_1^*, s_2^*, \dots, s_T^*$ 
```

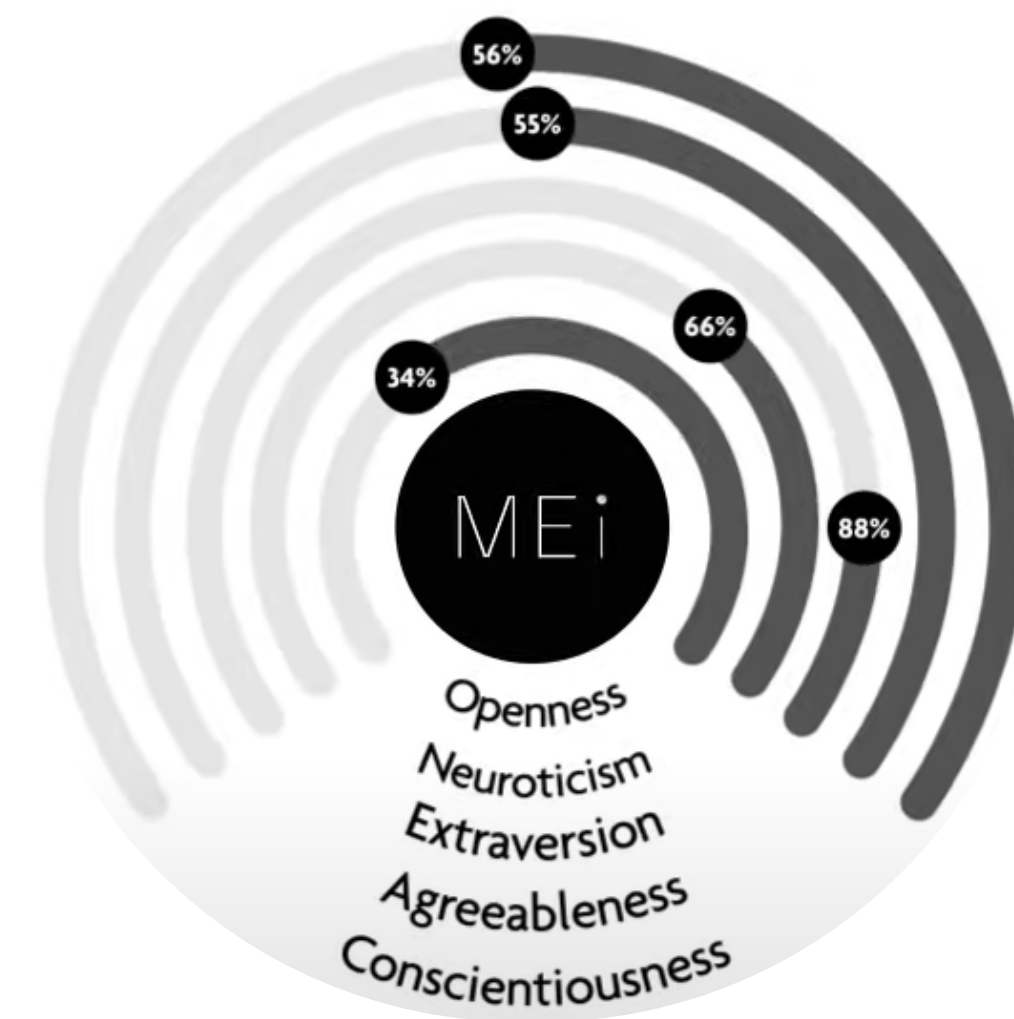




## Mei : SMS 메시지 + AI

관계 개선을 지원하도록 설계된 AI 비서 기반 메시징 앱

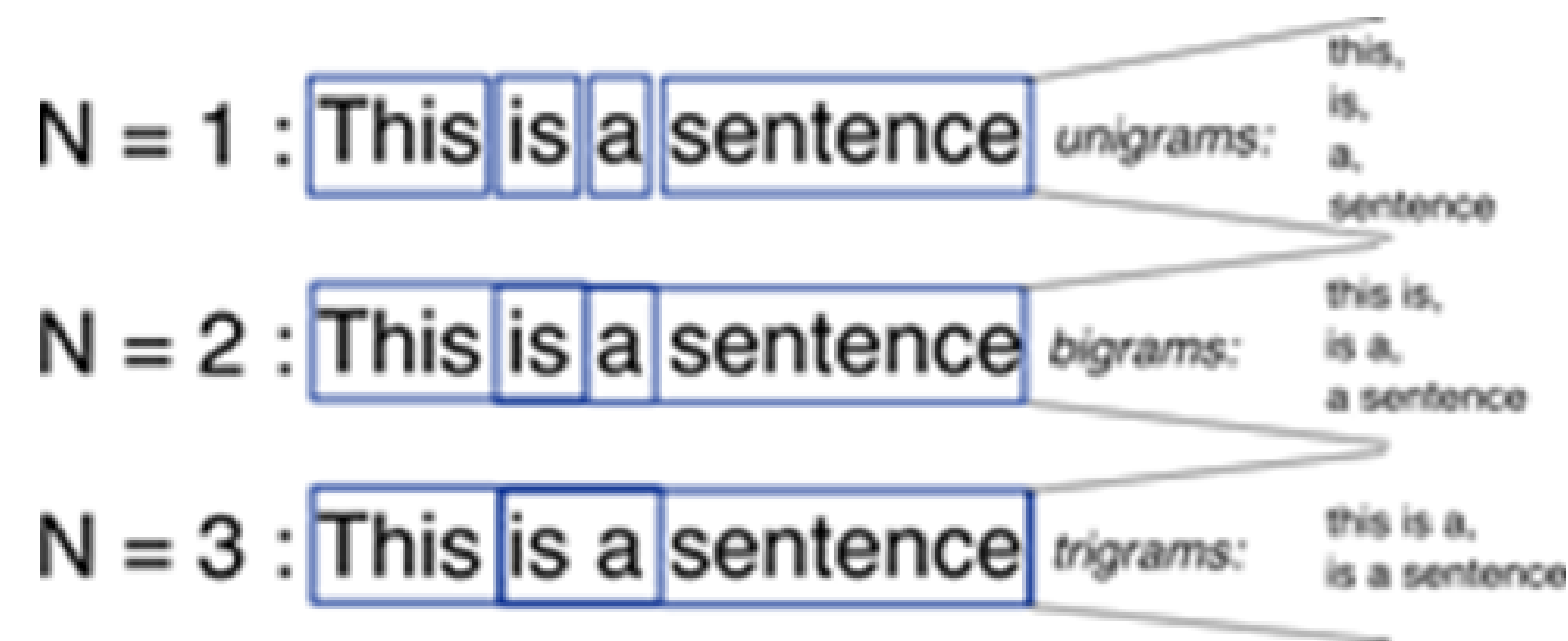
- 상대방과 나눈 문자 대화를 보고 조언이나 관찰을 제공
- 문자 내역에 기반해 상대방의 성격을 분석
- 서로에 대해 느끼는 감정에 대한 전체적 그림을 제공



△ 성격 특성 : 자신과 친구들의 성격 이해

# 관련 알고리즘 - N-gram

연속적인 n개의 토큰(word, character)으로 구성된 것



△ 연속해 몇개의 토큰으로 구성하는지에 따라 n이 달라짐. 이때 만들어지는 토큰은 단어의 숫자에 따라 unigram, bigram, trigram으로 나뉨

- 1-gram (unigram)  
word level : [fine, thank, you]  
character level : [f, i, n, e, , t, h, a, n, k, , y, o, u]
- 2-gram (bigram)  
word level : [fine thank, thank you]  
character level : [fi, in, ne, e, , t, th, ha, an, nk, k, , y, yo, ou]
- 3-gram (trigram)  
word level : [fine thank you]  
character level : [fin, ine, ne, , e t, th, tha, han, ank, nk, , k y, yo, you]

△ “fine thank you”를 세 가지 형태로 word level과 character level에서 N-gram으로 분석한 결과

## 참고 문헌

- <https://github.com/konlpy/konlpy>
- <https://www.ibm.com/kr-ko/cloud/learn/speech-recognition>
- <https://www.ibm.com/cloud/learn/natural-language-processing#toc-nlp-tasks-K4EAXccS>
- <https://untitledtblog.tistory.com/140>
- <https://assets.kpmg/content/dam/kpmg/kr/pdf/2020/kr-im-126-voice-ai-20200414.pdf>
- <https://www.marketsandmarkets.com/Market-Reports/speech-voice-recognition-market-202401714.html>
- <https://www.abiresearch.com/press/20-million-vehicles-shipping-2023-will-come-equipped-amazon-alexa-google-assistant-or-both/>
- [http://tcpschool.com/deeplearning/deep\\_sound](http://tcpschool.com/deeplearning/deep_sound)
- <https://textmei.com/about/>
- <https://youtu.be/y1OXiEo5mT8>