

애플 Siri와 자연어 인터페이스, 그리고 Wolfram 언어

School of Software, Kookmin University
Seung-Shik Kang



- 웹 검색 결과물을 결합하여 최적의 답변 제공
 - Citysearch, Gayot, Yelp, Yahoo! Local, AllMenus.com, Google지도, BooRah, OpenTable 등
- 시리의 개발자들이 추구한 것은 인터넷을 통해 대화를 나눌 수 있는 “do engine”
- 지능형 비서(intelligent assistant)로 설계
- 2010년 2월에 아이폰 앱으로 출시

Siri 개발 과정

- 가상 비서 개발 프로젝트 CALO

- Cognitive Assistant that Learns and Organizes
- 2003년, DARPA, 역사상 가장 큰 A.I. 연구 과제
- 27개 팀 개발 결과물을 통합하여 하나의 인공지능 비서
- 군 지휘관들의 정보 관련 업무부담을 줄이고 사무실 잡무를 덜어주는 목적

- SRI International, 5년간 500명 규모

- 회의 직전에 동료가 회의 불참 이메일을 보냈을 때
 - AI 비서는 프로젝트에서 각자의 역할을 이미 알고 있으므로 그 회의를 취소하는 게 좋을지를 판단하고 일정을 다시 잡아서 변경된 회의 초대장을 발송 및 회의실 예약
- 회의가 계획대로 진행되는 경우에는
 - 회의에 필요한 문서, 이메일을 수집하여 중요도에 따라 랭킹
 - 회의 내용을 인식하여 참석자들이 논의한 내용을 정리
 - 회의에서 논의되었던 개별적인 작업 정리
- CALO는 발표 자료를 정리해 주거나, 파일들을 폴더로 정리, 수신된 메시지 분류 등의 기능 수행

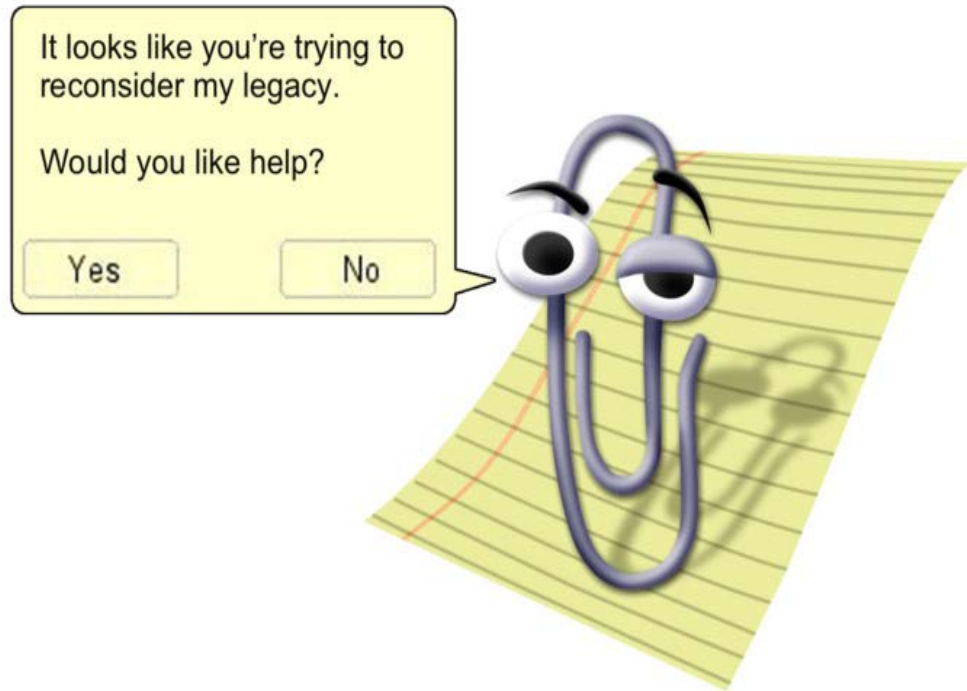
시리 공동 창업자들

- Cheyer, Kittlaus, Gruber



- Adam Cheyer 연구원
 - 실리콘 밸리의 전설 Doug Engelbart와 비교됨
 - 2007년에 SRI를 그만 두고 SRI 동료들과 함께 CALO 프로젝트와 Vanguard의 아이디어를 활용하여 스타트업 창업

Microsoft Clippy



전자 비서 history

- Wildfire Communications, 1994년 10월
 - 음성 전화를 기반으로 하는 전자 비서 “Wildfire”
 - 메시지 처리, 전화 걸어주기, 음성 메일 검색
- 마이크로소프트 Clippy 개인비서
 - 타임지로부터 최악의 발명품 50개에 선정
- General Magic
 - 애플사에서 독립 회사로 분리
 - 1998년, Portico라는 음성 제어 방식으로 인터넷과 이동전화를 연결하여 이메일을 읽고 메시지를 주고받는 전자 비서 개발 계획 → 실패

시리의 프로토타입

- 목적:
 - AI 비서가 음성으로 입력된 문장을 이해하고(understand), 해석하여(interpret), 질문에 대한 답변을 해주도록(answer) 학습시키는 것
- 방법:
 - 입력 문장에 대한 언어 개념(linguistic concept)을 모델링하는 대신에 실세계 객체(real-world object)를 모델링하는 방법을 취함
 - 문장 의미를 파악하는 자연어 처리 기법이 오류 발생 가능성이 높다는 문제점을 해결

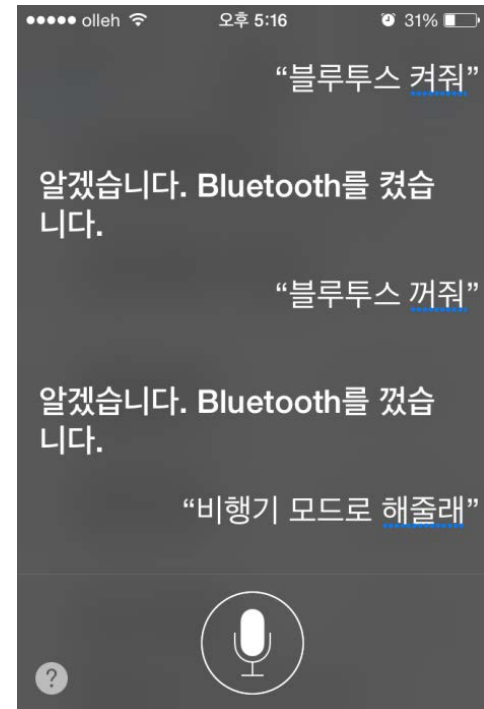
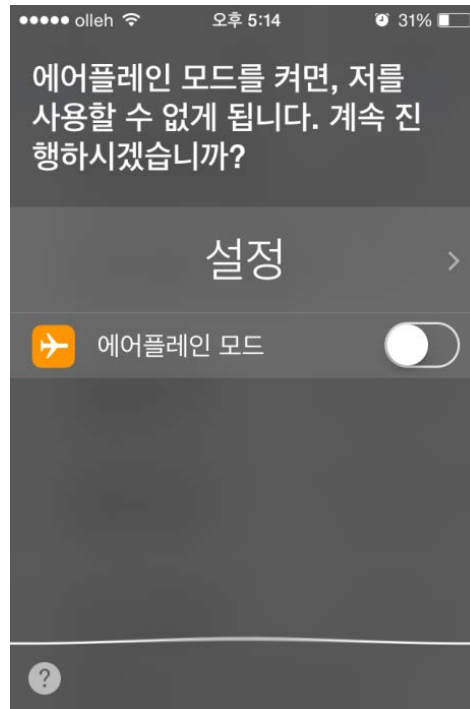
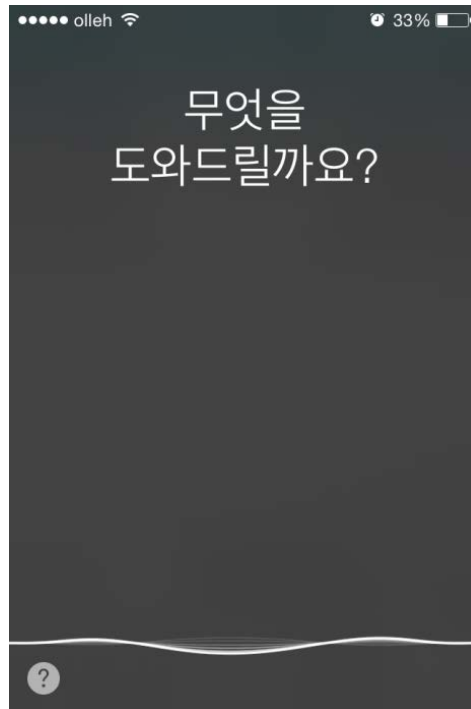
Real-world Object 모델링 예

- “I want to see a thriller”가 입력되면
 - 이 문장이 주어와 목적어가 술어에 어떻게 연관되는지를 분석하는 대신에 “thriller”를 영화 장르로 인식하고, 이와 관련된 영화를 가져오는 방식
 - 질문 내용을 잠재적 행동 영역에 매핑하고 실세계 개념들 간의 관계를 파악하여 가능성이 가장 높은 행위(action)를 선택
- 맛집을 추천할 경우에
 - 시리는 미리 식당들의 등급, 주소, 메뉴와 가격 정보를 알고 있음
 - 사용자의 선호도, 위치 정보, 또는 사용자에게 더 많은 정보를 요구하여 이로부터 사용자가 원하는 답변을 생성

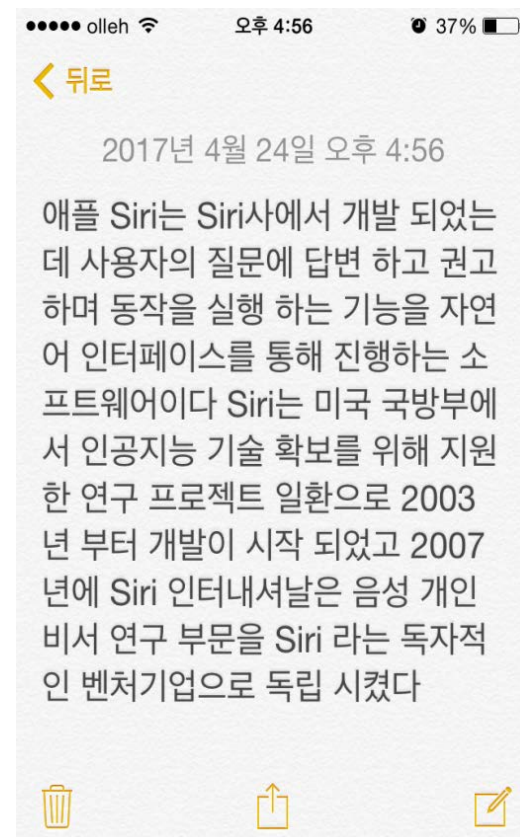
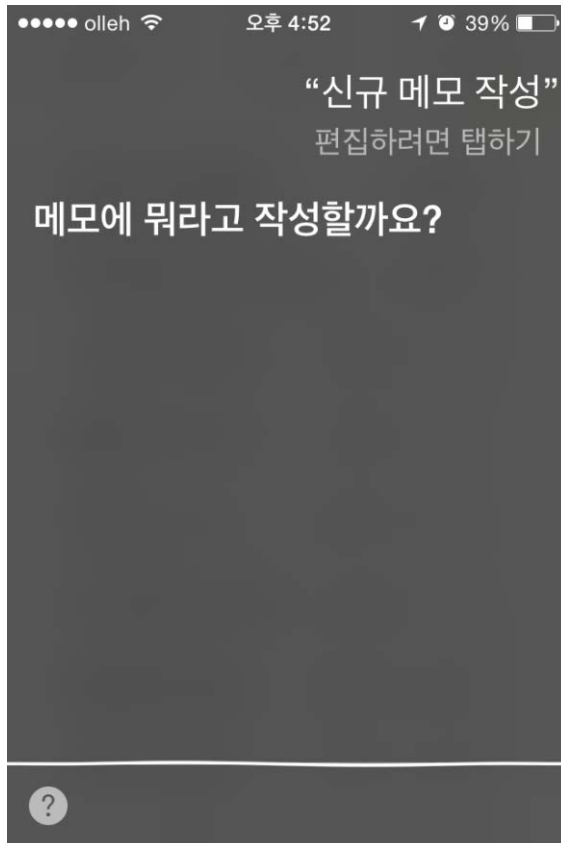
음성인식 오류의 자동 교정

- 음성인식 오류를 자동 교정을 위해 상황 정보 이용
 - 1) 시끄러운 호텔 로비에서 "closest coffee shop"이라고 물을 경우에 "closest call Felicia"로 들릴 수가 있음.
 - 2) 이 때, 호텔 직원은 "가장 가까운" 것이 사람보다는 장소를 지칭하는 경향이 있고, 손님이 식당이나 커피숍 위치를 문의할 가능성이 많은 것을 알고 있으므로 아마도 커피를 마시려고 한다는 것을 유추함.
 - 3) 마찬가지로 시리는 모든 단어의 의미를 파악하지 않더라도 함께 사용되는 구문들로부터 질문의 요지를 파악하는 방법을 사용
 - 4) 카페 목록을 제공해 주기 위해 제3자에게 API를 제공하는 수십 개의 웹서비트에 구축되어 있는 데이터들로부터 질문에 적합한 답을 생성

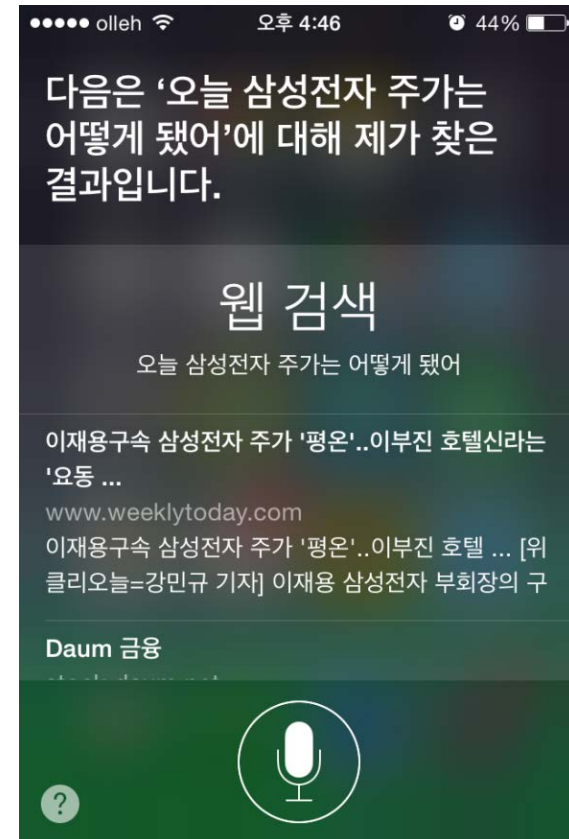
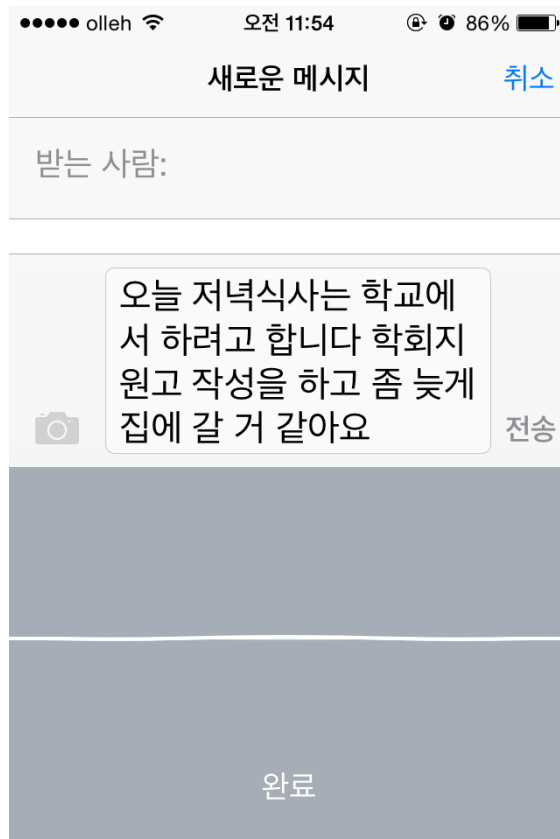
시리의 기능



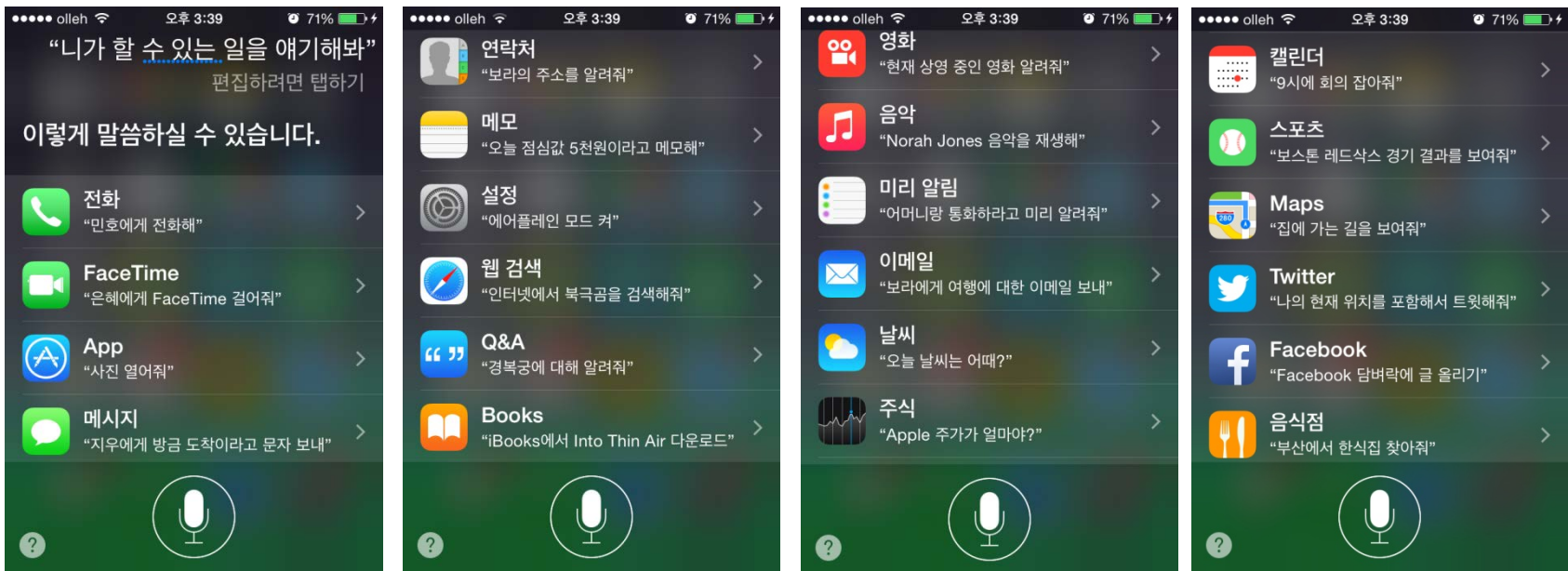
시리의 기능



시리의 기능



시리의 기능



시리의 성공 요인

- 시리의 개발자들은 시리가 스스로 학습할 수 있는 상투적인 질문과 대답을 자동화해야 할 분량이 거의 무제한
 - 수많은 웹서비스 API를 데이터베이스에 추가할 수 있게 하였음.
- 시리가 지능이 있고 인공적인 재미가 있어야 성공할 수 있음을 알고 있었음.
 - 개발자들은 시리에게 물어볼 가능성이 있는 질문들을 만들어 내고 시리가 제공하는 대답을 사용자들에게 다양한 형태로 제공하여 개성이 가미된 패키지를 선택하여 설치할 수 있게 함.
- 시리는 사용자의 말하는 습성을 인식할 수 있으므로 사람들의 개성을 모방할 수 있는 기능을 구현하고자 하였음.

시리의 주요 기술

1. 음성 인식? **Nuance 엔진**
2. 지식 검색? **Wolfram alpha 지식검색**
3. 아이폰 기능 작동: 전화걸기, 문자전송 등
4. 인공지능 비서: 식당예약, 티켓구매 등

음성 인식 엔진

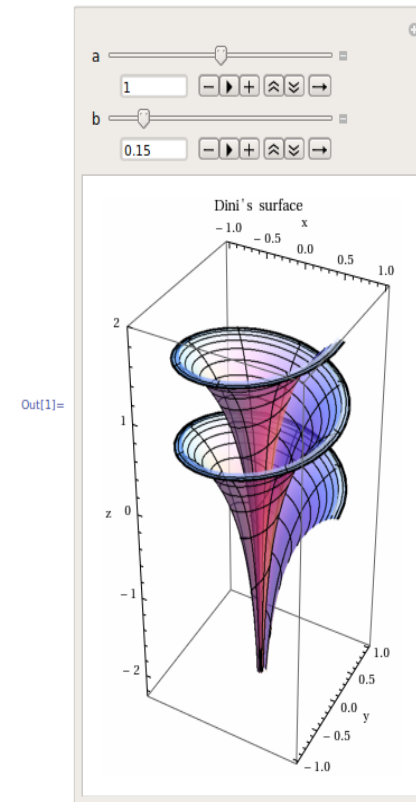
- 음성 신호를 텍스트로 변환해 주는 STT(Speech to Text) 기술
- PC에 적용된 최초의 사례, 1980년대 초
 - 드래곤 딕테이트(Dragon Dictate)
 - 드래곤 시스템(Dragon Systems), 도스용
- IBM, ViaVoice
- Nuance 엔진
 - 뉘앙스 커뮤니케이션즈(Nuance Communications)
 - 1994년, SRI의 STAR(Speech Technology and Research) 랩에서 독립
- 삼성전자 갤럭시의 S 보이스: 뉘앙스의 자회사 블링고의 음성인식 엔진
- LG전자의 Q 보이스: 자체 음성인식 기술
- 구글 API: 자체 개발한 음성인식 엔진 API 공개

울프람 알파와 울프람 언어

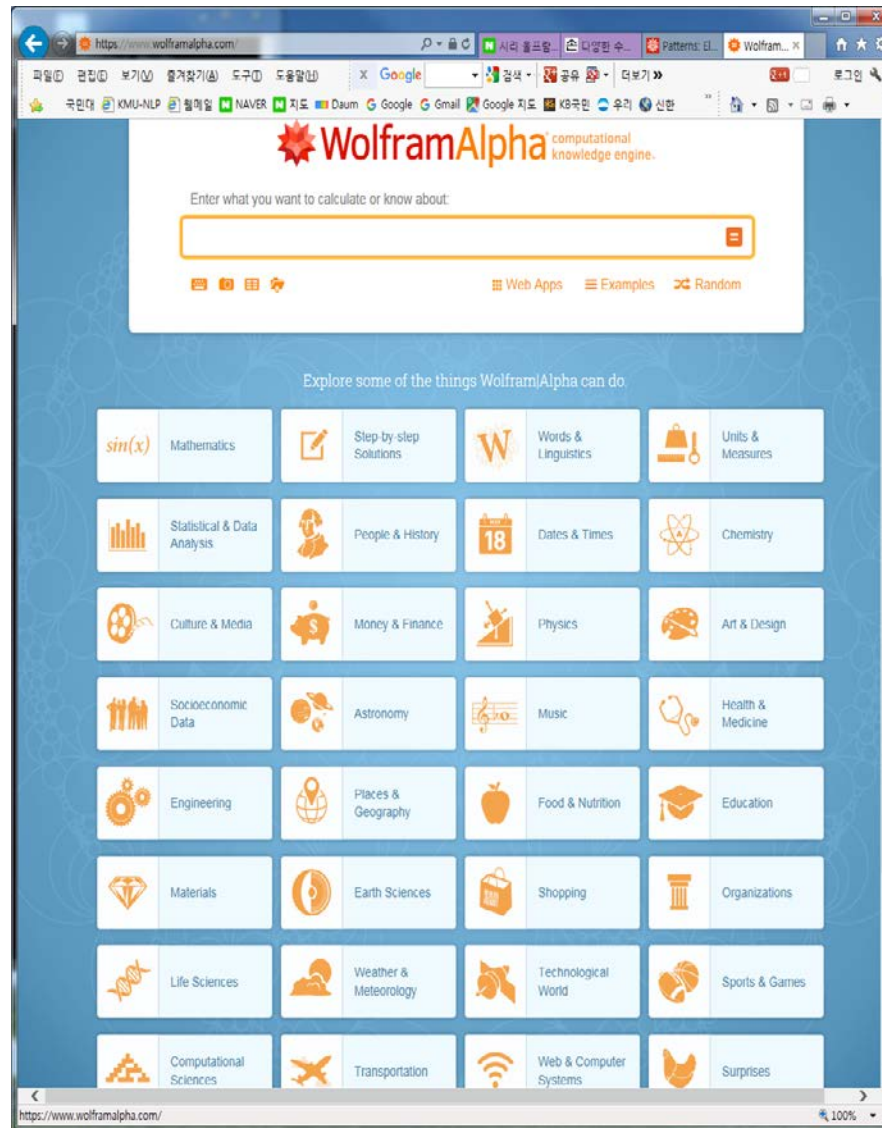
매스매티카, 1988년

- CalTech의 SMP 기반

```
▼In[1]:= Manipulate[
  ParametricPlot3D[
    {a Cos[u] Sin[v], a Sin[u] Sin[v],
     a (Cos[v] + Log[Tan[ $\frac{v}{2}$ ]]) + b u}, {u, 0, 4  $\pi$ },
    {v, 0, 2}, AxesLabel -> {"x", "y", "z"},
    PlotLabel -> "Dini's surface", PlotPoints -> 64,
    PlotStyle -> Opacity[0.7]},
  {{a, 1}, 0, 2}, {{b, 0.15}, 0, 1}]
```



울프람 알파 검색엔진



울프람 알파의 특징

- 울프람 알파는 데이터를 수집하고 선별하여 큐레이팅된 지식베이스와 정형화된 데이터로부터 정답을 계산하여 시각적으로 보여줌
 - 큐레이팅된 데이터는 울프람 알파를 많은 수의 응답을 색인화 한 후에 질문을 그중의 하나와 일치시키려고 시도하는 시맨틱 검색 엔진(semantic search engine)과 차별화
- Wolfram Alpha는 사회 과학이나 문화, 역사 등 세밀하고 복잡하게 요구되는 질문이 아니라 컴퓨터로 계산되는 사실을 기반으로 한 견고한 질의에 대한 결과를 제공
- Example
 - "Mary Robinson은 어디에서 태어났습니까?"
 - "1974년에 Queen Elizabeth II는 몇 살이었습니까?"
 - "율리우스력에서 6월 1일은 며칠입니까?"
 - "1 인당 GDP는 50 번째로 작은 나라는?"
- 사실적인 또는 계산적인 질문에 대한 대답을 생성

- 데이터베이스에는 "All Current and Historical Weather"와 같은 수백 개의 데이터 세트
- Facebook 사용자 프로파일: 실시간 데이터 세트 예
 - 사용자가 자신의 계정 정보를 Wolfram 사이트와 공유하도록 권한 부여
 - "facebook report" 쿼리를 통해 가져옴
 - 울프람 알파는 친구들의 연령 분포, 상태 업데이트에 사용된 단어의 빈도 및 기타 세부 정보를 포함하는 "개인 분석" 보고서를 생성
 - Facebook의 분석 서비스는 시작한지 2주 만에 40만 명의 사용자들이 사용

Wolfram Language

- 2014년, 지식검색 분야에 적합한 언어
 - 지식검색과 추론 기능을 구현이 편리함
 - 지식기반 프로그래밍 언어
 - 매스매티카와 Wolfram Alpha 개발에 사용
- 혼합형 언어(multi-paradigm programming language)
 - 기호 계산(symbolic computation)
 - 함수형 프로그래밍(functional programming)
 - 규칙기반 프로그래밍(rule-based programming) 기능 포함
- Wolfram Alpha는 1,500만 줄의 Wolfram 언어 코드로 작성되었으며 10,000개 이상의 CPU에서 실행된다.

울프람 언어의 내장함수

- 약 5,000 개의 내장 함수 포함

```
↓  
In[1]:= Range[20]  
Out[1]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}  
  
↓ ↓  
In[2]:= NestList[f, x, 5]  
Out[2]= {x, f[x], f[f[x]], f[f[f[x]]], f[f[f[f[x]]]], f[f[f[f[f[x]]]]]}
```

ListPlot ▫ GeoDistance ▫ Entity ▫ BinaryReadList ▫ Cases ▫ StringReplace ▫ FixedPoint ▫ SortBy ▫
Select ▫ Abs ▫ ParallelTry ▫ MovingAverage ▫ Classify ▫ Flatten ▫ CurrentImage ▫ ColorNegate ▫
VoronoiMesh ▫ FindShortestPath ▫ TimeSeries ▫ Module ▫ CloudDeploy ▫ ...

전체 가이드:

Core Language & Structure	Data Manipulation & Analysis	Visualization & Graphics	Symbolic & Numeric Computation
Strings & Text	Graphs & Networks	Images	Geometry
Sound	Time-Related Computation	Geographic Data & Computation	Scientific and Medical Data & Computation
Engineering Data & Computation	Financial Data & Computation	Social, Cultural, & Linguistic Data	Higher Mathematical Computation
Documents & ...	User Interface	System Operation	External Interfaces

울프람 언어

- Wolfram 언어는 기호를 사용한 수식(expression)으로 표현
- 수식 표현 대상: 수치, 문자열, 이미지, 배열, 그래프, 수식, 문서, 인터페이스, 코드 등
- 기호를 사용하는 모든 수식 표현은 “헤드[인수]” 형식으로 기술

$\{a, b, c\}$ `List[a, b, c]`

$2+2$ `Plus[2, 2]`

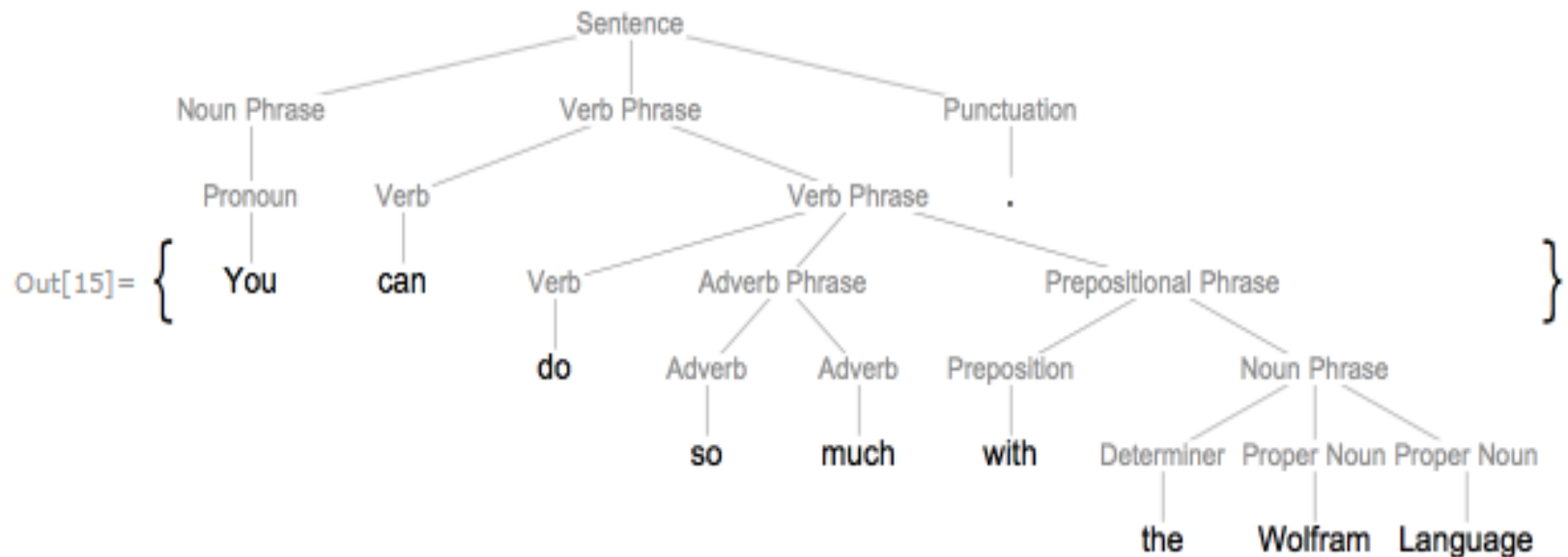
$x^2 + 3y^3$ `Plus[Power[x, 2], Times[3, Power[y, 3]]]`



..... `Graphics3D[Sphere[]]`

울프람 언어: 자연어 이해 시스템

```
In[15]:= TextStructure["You can do so much with the Wolfram Language.", "ConstituentGraphs"]
```



울프람 언어: 자연어 인터페이스

- "Picture of a wolf"
- "Plot a sine curve"

picture of a wolf ← ENTER 를 누르면 해석을 시작합니다

In[1]:= **picture of a wolf** »

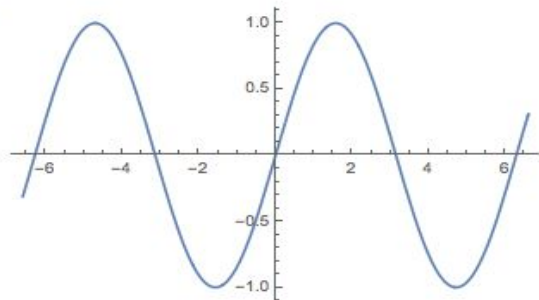
gray wolf (species specification) **image** ← 해석

Out[1]=  ← 결과

plot a sine curve

In[2]:= **plot a sine curve** »

Plots (1 of 2)
Plot[Sin[x], {x, -6.6, 6.6}]

Out[2]= 

울프람 언어: 자연어 명령

- GrammarToken, GrammarRules
- "add A and B"

```
In[1]:= addition =  
  CloudDeploy[GrammarRules[{FixedOrder["add",  $a$ : GrammarToken["SemanticNumber"],  
    "and",  $b$ : GrammarToken["SemanticNumber"]]  $\Rightarrow a + b$ ]  
  ]  
]
```

```
Out[1]= CloudObject[https://www.wolframcloud.com/objects/03de39cf-c9f2-4432-8128-45697064eb9b]
```

Apply the deployed grammar to a natural language string:

```
In[2]:= GrammarApply[addition, "add six and seven"]
```

```
Out[2]= 13
```

가전제품 제어 프로그램: 작성

```
In[1]:= controller = CloudDeploy[
  GrammarRules[{
    FixedOrder["turn", state : GrammarToken["OnOff"], OptionalElement["the"],
      appliance : GrammarToken["Appliance"]] := appliance → state
  },
  {
    "OnOff" → GrammarToken["On"],
    "OnOff" → GrammarToken["Off"],
    "On" → "on" :=> "On",
    "Off" → "off" :=> "Off",
    "Appliance" → GrammarToken["Stove"],
    "Appliance" → GrammarToken["Refrigerator"],
    "Appliance" → GrammarToken["Furnace"],
    "Appliance" → GrammarToken["AirConditioning"],
    "Appliance" → GrammarToken["Oven"],
    "Stove" → FixedOrder["stove", OptionalElement["top"]] :=> "Stove",
    "Refrigerator" → "fridge" | "refrigerator" :=> "Refrigerator",
    "Furnace" → "heat" | "furnace" | "boiler" :=> "Furnace",
    "AirConditioning" → FixedOrder["air", OptionalElement["conditioning"]] |
      "ac" | "a/c" :=> "AirConditioning",
    "Oven" → "oven" :=> "Oven"
  }
]
```

```
Out[1]= CloudObject[https://www.wolframcloud.com/objects/23aa0b1c-ddbf-4118-9446-42ee92ed8edc]
```

스토브와 에어컨 작동 예제

```
In[2]:= GrammarApply[controller, "turn off the stove"]
```

```
Out[2]= Stove → Off
```

```
In[3]:= GrammarApply[controller, "turn on a/c"]
```

```
Out[3]= AirConditioning → On
```


맺음말

- 시리가 탄생하게 된 계기: CALO 프로젝트
 - AI 에이전트
 - 사람의 습성이나 관련 업무를 파악하여 개인비서 역할을 수행하는 지능형 비서
- 스마트폰 탑재되어 상용화 과정에서
 - 전화걸기, 문자 보내기 등 음성명령 인터페이스
 - 울프람 알파를 이용한 지능형 검색
 - 티켓 구매, 식당 예약, 택시 호출, 주가 정보 제공 등
 - 다양한 웹서비스를 자연어 인터페이스로 편리하게 처리
- Intelligent Speaker
 - 아마존 Alexa, SK 텔레콤의 Nugu, 구글 Now, 삼성전자의 빅스비(Bixby)

감사합니다!

sskang@kookmin.ac.kr

<http://nlp.kookmin.ac.kr/>