

Python과 Tensorflow를 활용한 AI Chatbot 개발 및 실무 적용

WRITTEN BY SuSang Kim
healeess1@gmail.com



1. 도커실행환경

https://github.com/TensorMSA/skp_edu_docker

2. 소스설명코드 - jupyter (/chap13_chatbot_lecture)

git clone https://github.com/TensorMSA/tensormsa_jupyter.git

3. Python 3.5 / Tensorflow 1.2



I am Susang Kim as a developer

- Chatbot Developer
 - Released in POSCO (Find people using by NLP/AI)
 - Deep Learning MSA (ML,DNN, CNN, RNN)
- Agile Developer (Experienced in Pivotal Labs)
 - TDD, CI, Pair programming, User Story
- iOS Developer (Ranked App store in 100th - 2011 Korea)
- Front-End Developer (React, D3, Typescript and ES6)
- POSCO MES ... (working at POSCO ICT for 10 year)



Contents

1. 도입

2. AI Chatbot 소개

Chatbot Ecosystem

Closed vs Open Domain

Rule Based vs AI

Chat IF Flow and Story Slot

3. AI기반의 학습을 위한 Data 구성 방법

Data를 구하는 법 / Train을 위한 Word Representation

Data의 구성 / Data Augmentation(Intent, NER)

4. 자연어처리 위한 AI 적용 방안

Intent (Char-CNN) / QnA (Seq2Seq)

Named Entity Recognition (Bi-LSTM CRF) / Ontology (Graph DB)



Contents

6. Chatbot Service를 위한 Architecture 구성

- Chatbot Architecture

- NLP Architecture

- Web Service Architecture

- Bot builder / Chatbot API

- Test Codes for Chatbot

7. 실무에서 발생하는 문제와 해결 Tips

- Ensemble and voting / Trigger / Synonym(N-Gram)

- Tone Generator / Parallel processing / Response Speed

8. 마무리

[설명 코드]

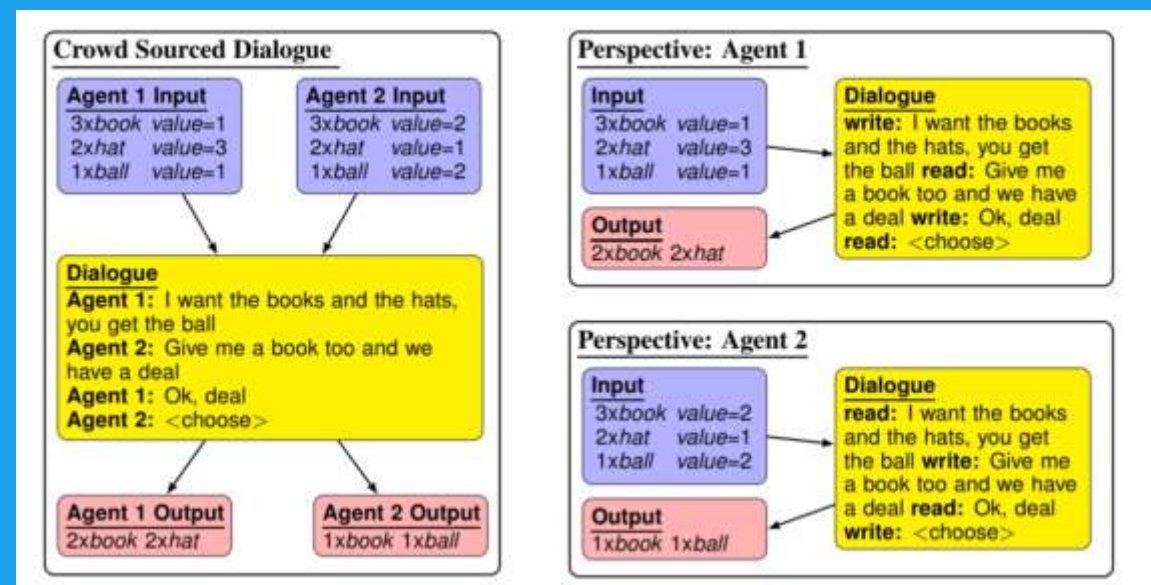
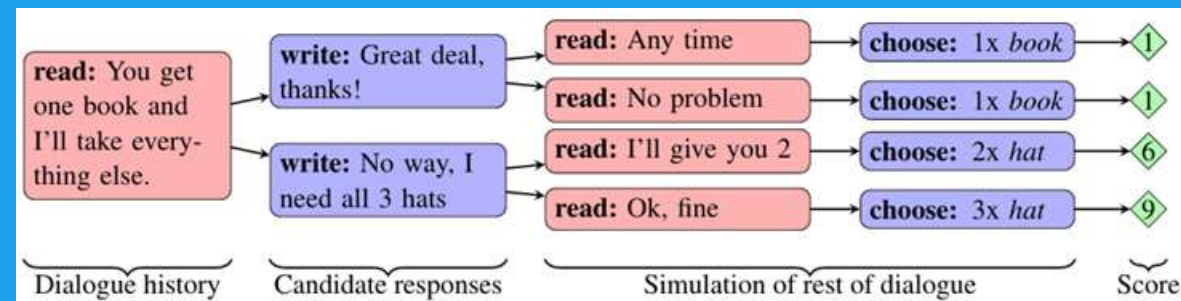
Text Augmentation / Char-CNN / NER / Slot Bot / QA Bot / Graph DB / Response Generator



도입



Facebook AI shut down after creating their own language

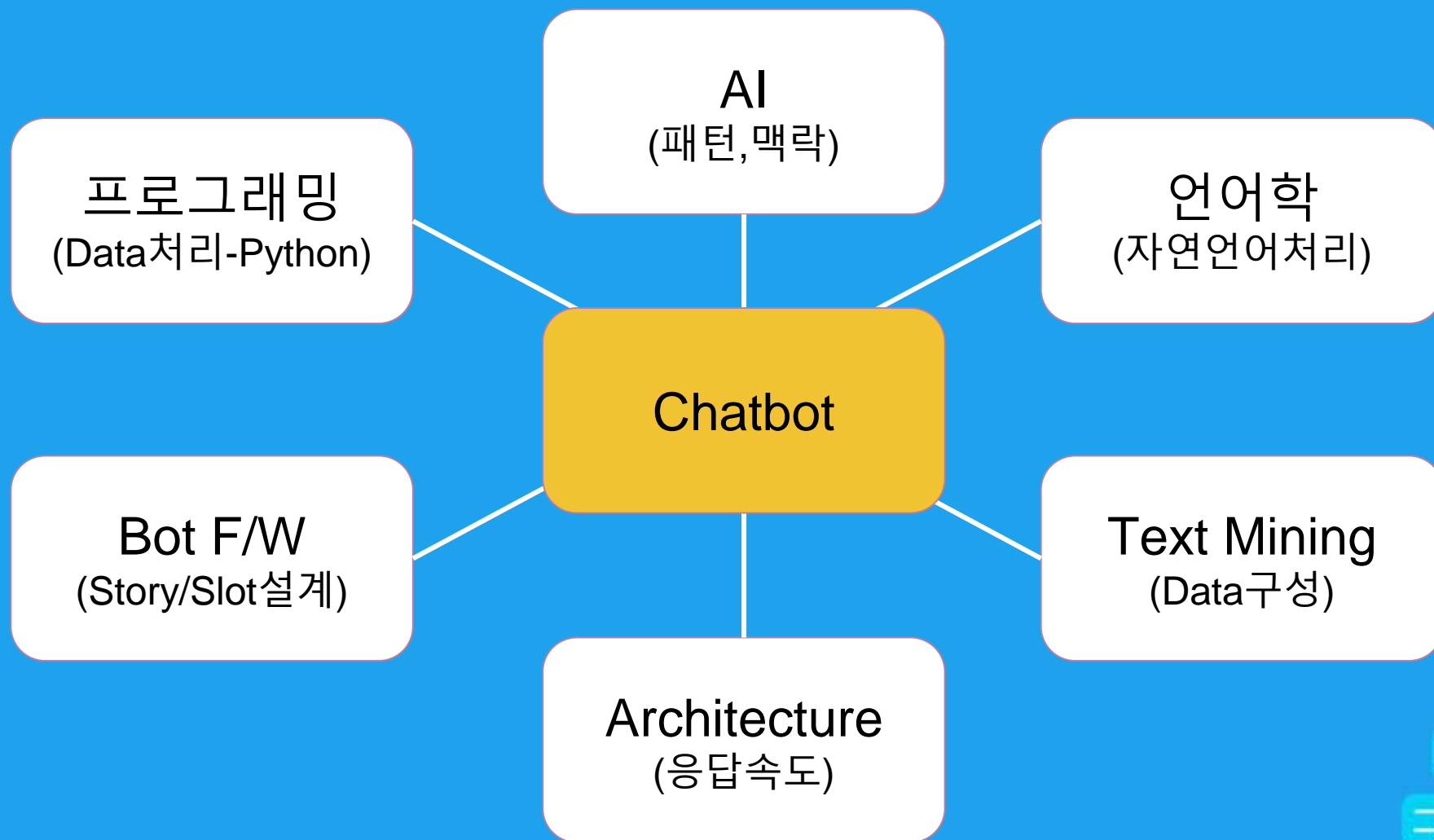


Chatbot을 개발하게된 이유?

- 많은 기술이 필요 (NLP, AI, F/W, Text Mining and 다양한 개발 skill)
- Deep Learning을 공부하는 입장에서 결과 확인이 빠름
 - 적은 Computing으로 빠른 결과확인 가능 (Text 기반)
- 재미가 있음(Micro Data처리에 비해 Biz dependency가 적은편)
 - 이미지(CNN)이나 정형Data(DNN)보다는 Data처리에 대한 부담감이 적음
(형태소 분석기등으로 쉽게 전처리 쓴다는 가정하에)
- 응용분야가 많음 (API기반의 다양한서비스 연결 Smart Management)
 - Intent와 Slot만 채워주면 어느 서비스와 연결가능
- 관련 오픈소스가 적어 블루오션 (한글은 대부분 자체개발해야함)
 - 다행인건 딥러닝 기반의 언어독립적 Text algorithm이 많이 공개되어 활용 가능
- Bot Service가 있으나 가격부담, 한국어는 잘안됨, Customize 불가



Chatbot은?



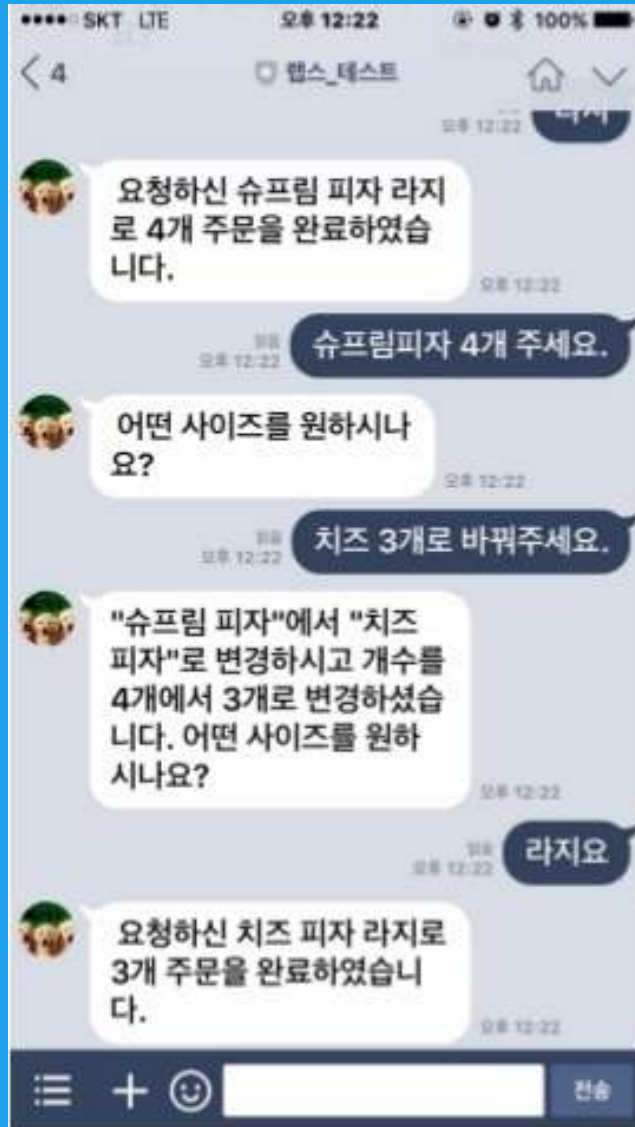
Chatbot 구현을 위해서는 많은 분야의 다양한 기술 필요



AI Chatbot 소개



요즘 왜 Chatbot이 뜨는가??



- Chatbot 으로 서비스 하려면 ?

- "슈프림피자 4개 주세요"

- 메뉴 : 슈프림피자
- 수량 : 4개
- 의도 : 주문

- Natural Language Understanding

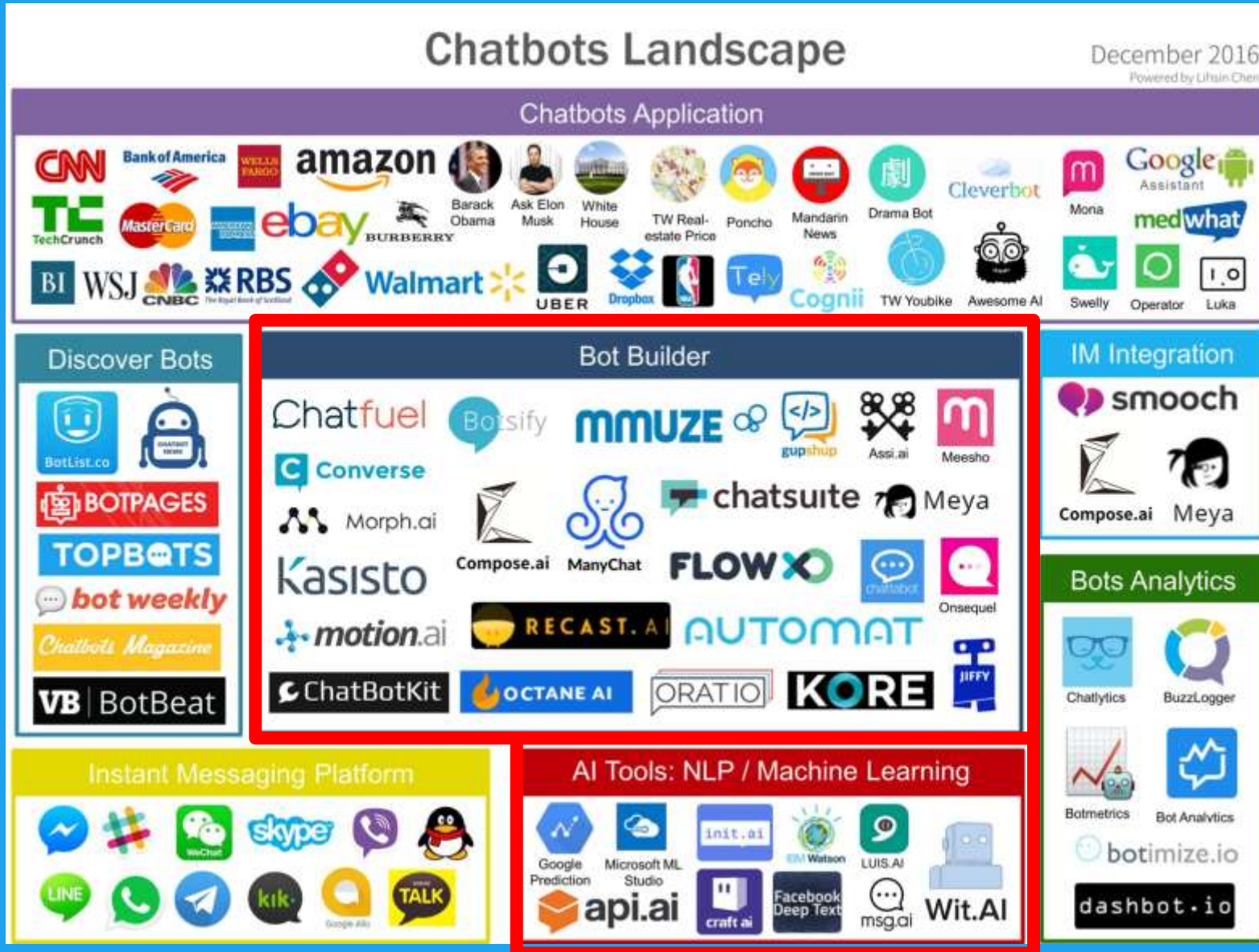
- Who?

- 서비스 개발자
- 어렵다. 귀찮다

직관적인 UX
일관성 있는 경험
음성과 연결 가능
별도 App 설치 필요 없음
다양한 서비스와 연결 가능
빠른 Feedback
플랫폼에 독립

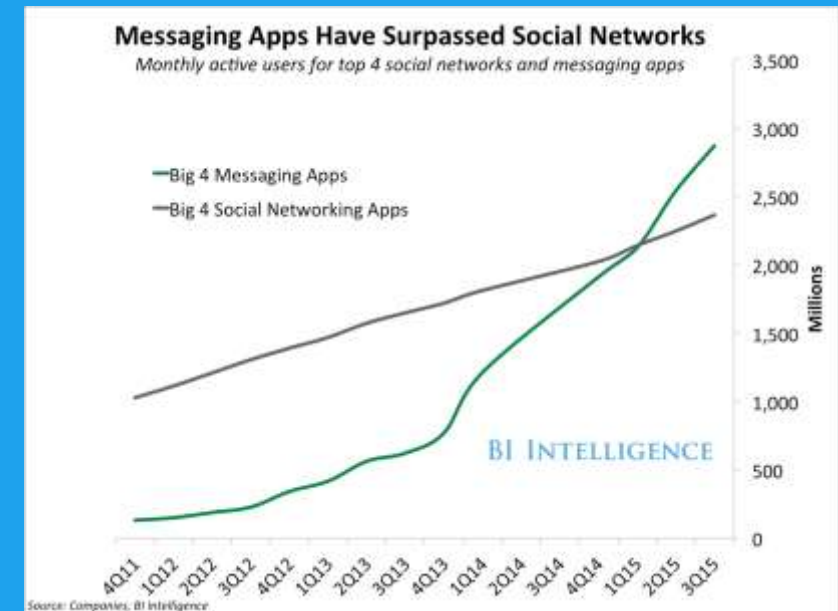


Chatbot Ecosystem

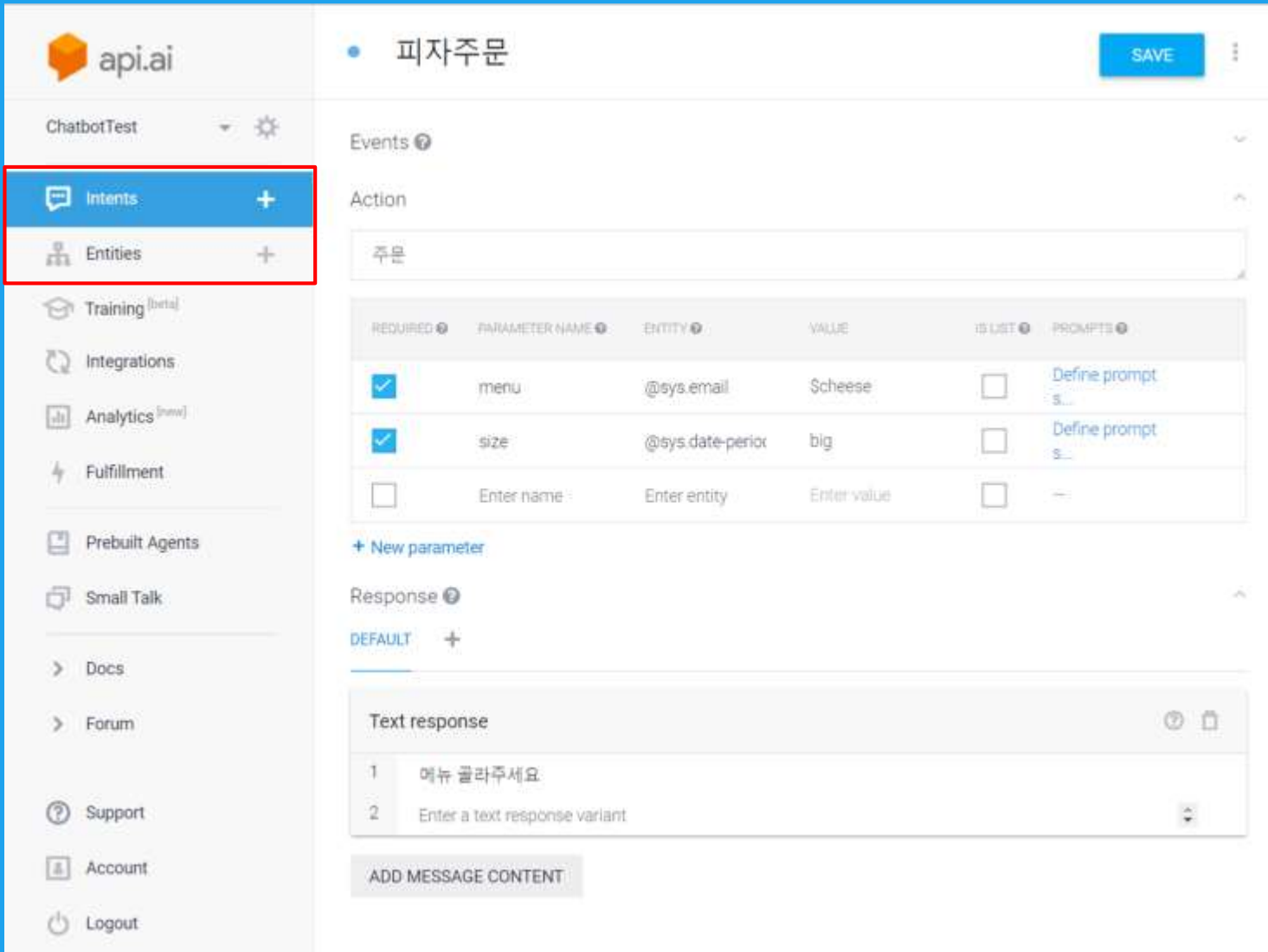


Chatbot 생태계는 계속 증가하고 있음

이번에 증점적으로 설명할 부분은 NLP와 Bot Builder (Slot기반 챗봇 / QA봇 위주)



다양한 Chatbot Platform이 존재는하고 있음



The screenshot shows the api.ai web interface for configuring a chatbot. The left sidebar contains navigation links: ChatbotTest, Intents, Entities, Training, Integrations, Analytics, Fulfillment, Prebuilt Agents, Small Talk, Docs, Forum, Support, Account, and Logout. The 'Intents' and 'Entities' links are highlighted with a red box. The main area is titled '피자주문' (Pizza Order) and shows the configuration for an intent named '주문' (Order). It includes a table for parameters, a 'Response' section with a 'Text response' variant, and a 'SAVE' button.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	menu	@sys.email	Scheese	<input type="checkbox"/>	Define prompt s...
<input checked="" type="checkbox"/>	size	@sys.date-period	big	<input type="checkbox"/>	Define prompt s...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

Response

DEFAULT +

Text response

- 1 메뉴 골라주세요
- 2 Enter a text response variant

ADD MESSAGE CONTENT

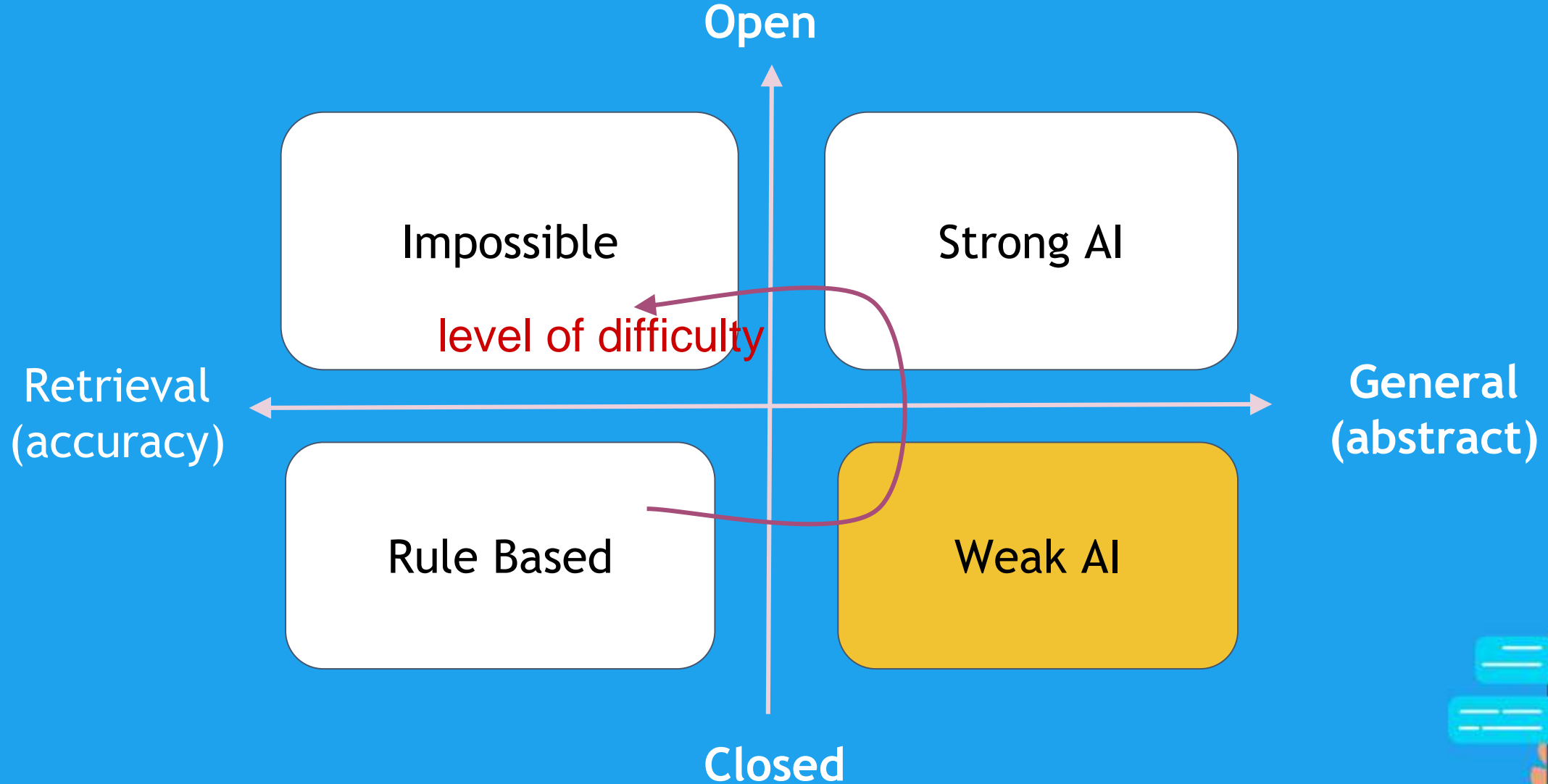
모든 챗봇에는 의도와 개체인식이 존재 또한 그 것을 위해서는 Data가 중요함!!!

api.ai에 가입해서 챗봇을 만들어보면서 원리를 파악해보면 도움이 됨



API.AI로 코딩없이 챗봇 만들기 <https://calyfactory.github.io/api.ai-chatbot/>

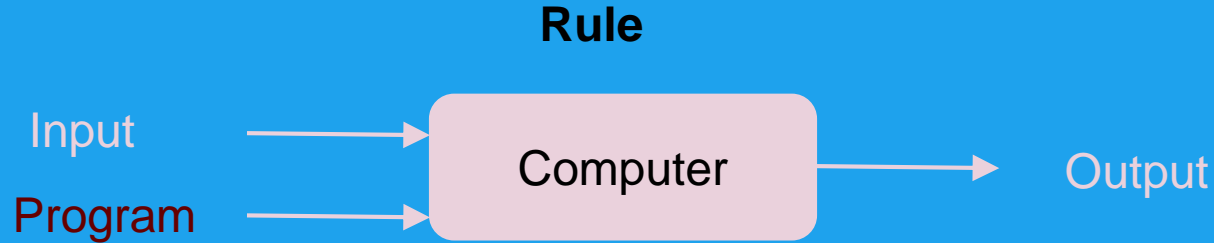
Closed Domain vs Open Domain



작은 Biz 도메인으로 시작해서 정확도를 높이면서 여러 Biz를 추가하는 상황

Rule Based vs AI

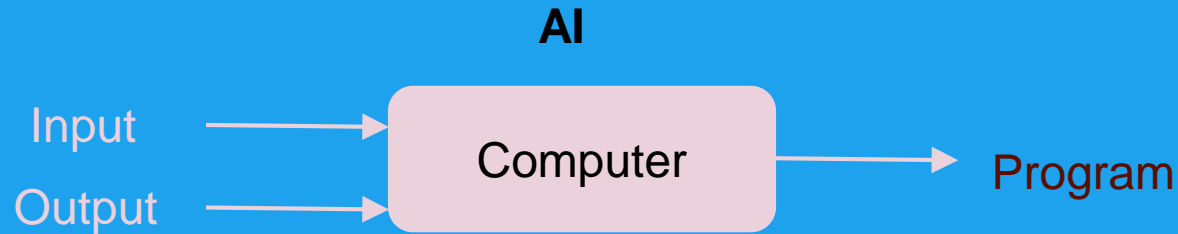
정확한 결과를 얻을 수 있으나 모든 질문은 불가



이름, 지역, 팀등 조건별로 일일이 rule을 등록해야한다
- 정확도는 올라가나 모든 질문을 다 등록??

```
If (loc = 판교 and comp = 포스코ICT)
    person = 김수상
elif (loc = 판교 and comp = SK Planet)
    person = 임재우
else
    person = 홍길동
```

비슷한 유형의 질문은 적당히 잘 찾아줌 Data가 많을 수록 정확도 향상(학습효과)

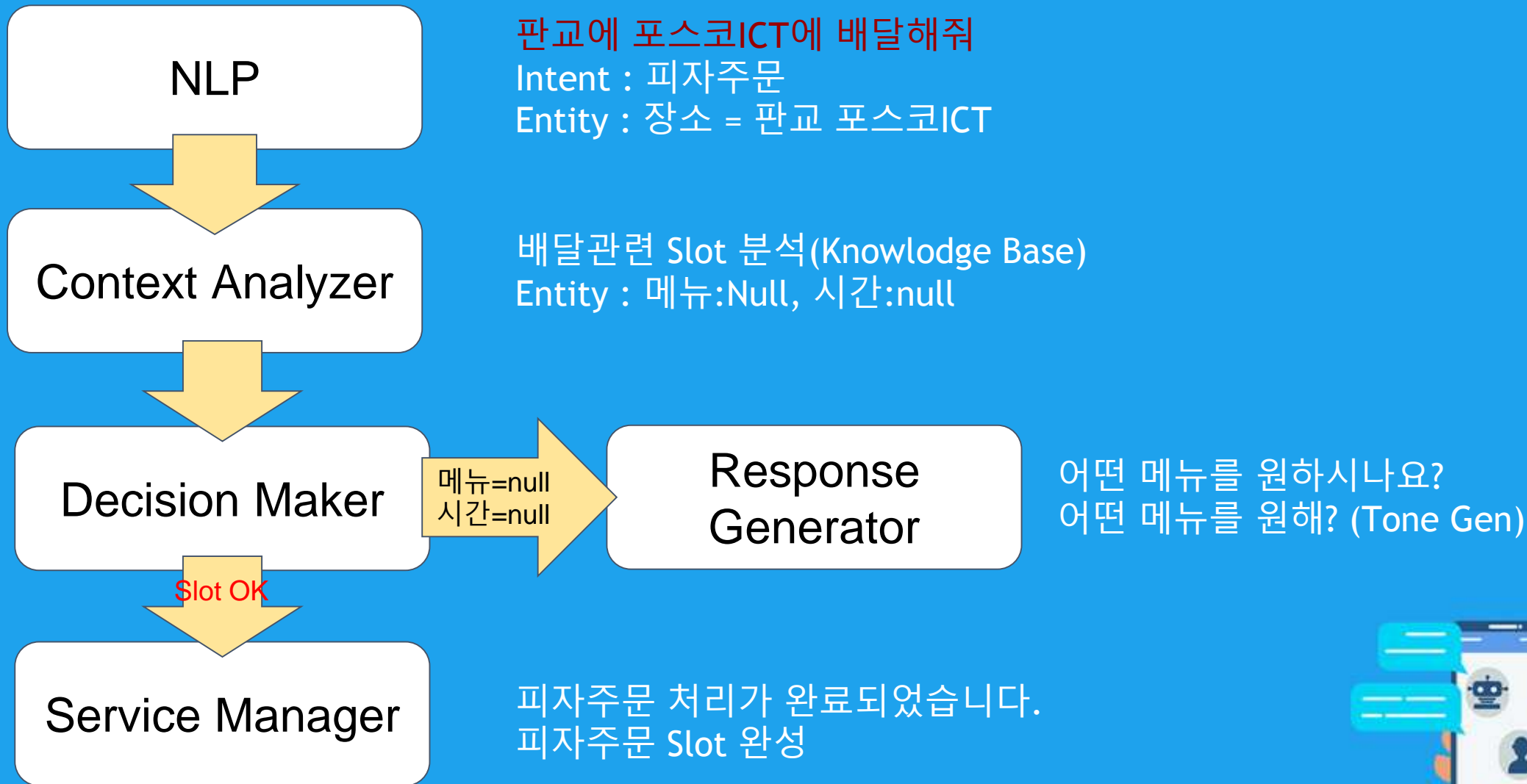


라벨링된 Data만으로 결과를 구할 수 있는 모델을 만들 수 있다
- 비슷한 Data들도 잘찾는편(W2V,GloVe)

intent = 판교에 근무하는 김수상 찾아줘 => Intent : 특정 지역 사람 찾아줘
NER = 판교에 근무하는 김수상 찾아줘 => B-Loc O O B-Name O



Chatbot Interface Flow



Story slot의 구성 (Frame-based DM)

피자 주문하고 싶어 → 피자 주문 의도 파악 →

Pizza Slot	
Size	
Type	
Side menu	

피자 Bot의 스토리 구성
1) 어떤 사이즈를 원하시나요?
2) 어떤 종류를 원하시나요?
3) 사이드 메뉴는 필요하신가요?



사용자 답변

- 페파로니 피자라지 사이즈에
콜라추가해주세요

서비스 연결
(Slot API Call)

NER처리 및 Slot 구성

Pizza Slot	
Size	Large
Type	Pepperoni
Side menu	cola

처리를 위해 Slot를
보여주는 것도 방법





새로운 쇼핑도우미
챗봇 바로에게 물어보세요.

네~ MacBook Pro로 찾아볼게요.

오후 10:37 API

찾으시는 MacBook Pro 최신 상품을 보여드릴
게요. 추천해 드린 상품 중 바로할인가라고 표시
된 가격은 11월에서만 추가 할인해 드리는 특별
한 혜택 🎁 이에요.

오후 10:37

Slot

Trigger

1. 맥북 프로 검색해줘
2. 전처리 -> 맥북 프로 NER
3. 맥북프로 -> 대표 Entity처리 -> MacBook Pro API Call
4. 검색결과 출력
5. 상세 서비스 조회를 위한 Slot 출력
6. 새상담 원할 경우 새상담 클릭

Slot를 선택할 수 있게 화면에 출력함으로써 챗봇의
정확도를 대폭 향상 시킬 수 있음
(해당 Frame안에서만 선택할 수 있기에...)

ex) “삼성 노트북” 쳐보면 Slot별 선택



바로봇

<http://www.11st.co.kr/toc/bridge.tmall?method=chatPage>

원하는 상품을 바로 찾아주는 디지털 컨시어지

AI기반의 학습을 위한 Data 구성방법



Data를 어떻게 얻는가?

일반적으로 Biz에 따른 Text는 존재하나 Deep Learning를 구현하기 위해서는 정제된 Text Tagging이 가능한 매우 많은 Data가 있어야함

한국어 Corpus를 일반적으로 세종 말뭉치를 사용하여 추가적인 Biz 어휘는 새로 학습시킴(노가다)

- Corpus (annotation) 세종말뭉치(2007) <https://ithub.korean.go.kr/user/main.do>
- 물결21 (2001~2014) 소스는없음 <http://corpus.korea.ac.kr/>
- Web Crawling or down (Wiki, Namu Wiki)
- Domain Specific의 경우엔 Text Data는 만들어야함

특화된 단어의 경우 새로 학습시켜야함 (ㅎㅇ? , 방가방가)

※ 고유명사등 새로운 어휘가 생성될때 새로 등록을 해주어야함



Train을 위한 Word Representation

Word Representation의 정의 (컴퓨터가 잘 이해할수 있게)

- One Hot은 단어별 강한 신호적 특성으로 Train 에 효과적 (Scope가 작을경우-Sparse)
- Word 단위 Embedding 은 단어를 잘 기억함 (But Sparse) / W2V (유사도)
- GloVe는 단어의 세부 종류까지도 구분 (Global Vector : 카라칼-고양이)
- Char 단위 Embedding 은 미훈련 단어 처리에 용이 (Vector을 줄이기위한 영어변환)
- 한글을 변환한 영어 Char 단위 Embedding는 벡터 수를 줄이면서 영어 처리도 가능



학습시킬 Data의 구성

Train Vector를 정한 후 Feature를 뽑아야함

Cleansing -> Feature Engineering -> Train

(상황별 특수문자 제거, 의미 있는 단어 도출 - Tagging)

의도나 객체와 상관있는 단어만 추출해내어

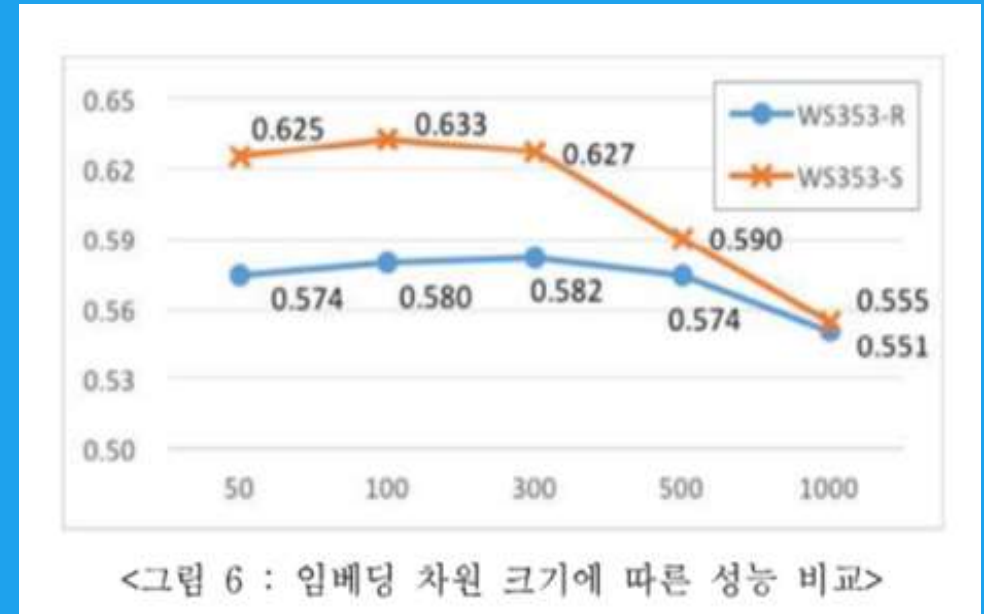
성능을 향상시킴 Train Cost를 줄이고 모델의 성능을 향상)

임베딩 차원도 줄이는 효과 (Dense Representation-SVD)

abcd~z, 0~9, ?, !, (,),',', 공백등 약 70여개

초중종성으로 글자를 쪼개기에는 어려움

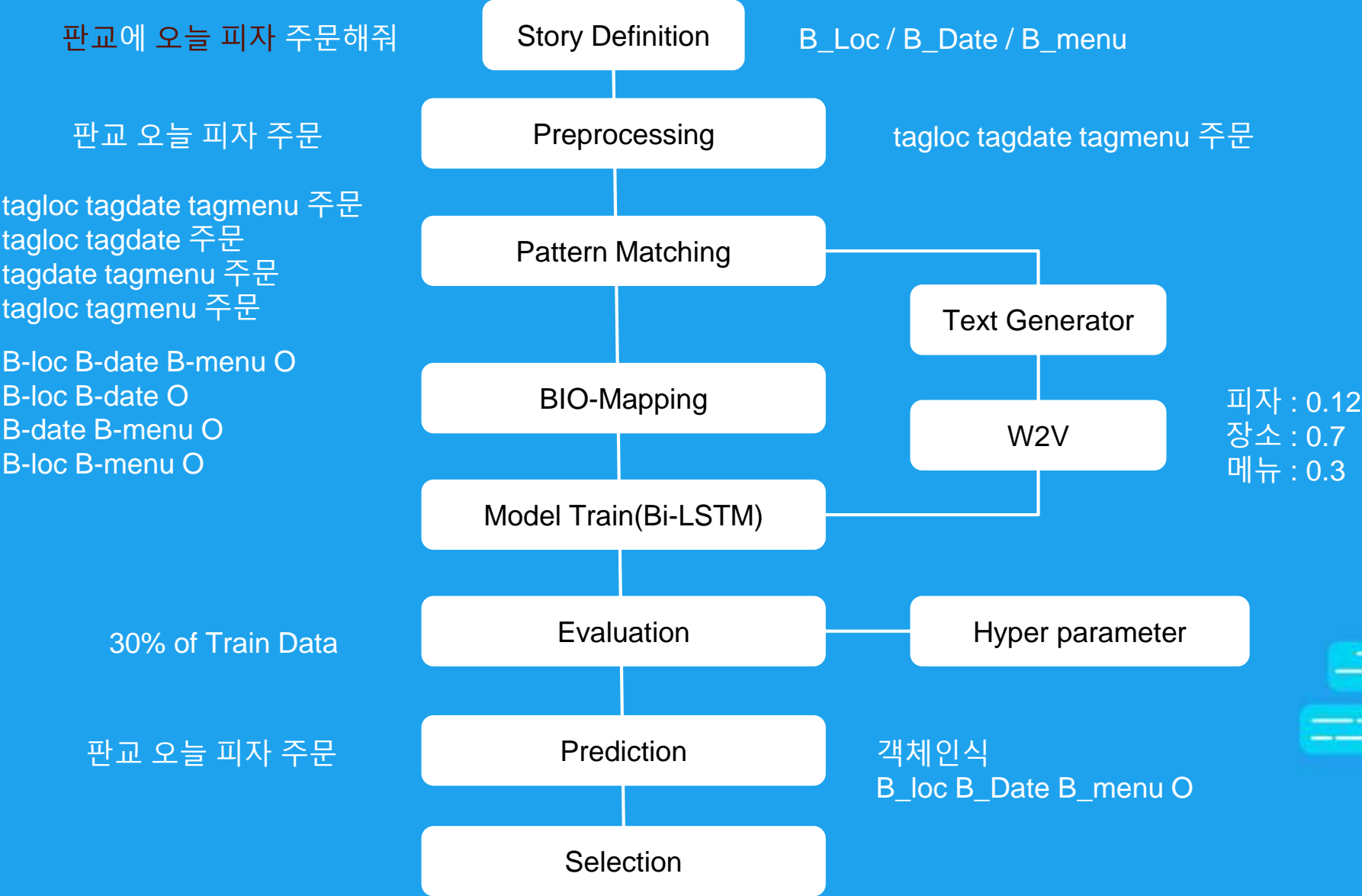
.lower()를 활용하는것도 방법 백터 줄이기



Data Augmentation for Deep Learning (Intent - tag)



Data flow for Model in Deep Learning (NER - BIO)



자연어처리 위한 AI 적용 방안



Intent를 알아내는법 (Text Classification)

피자주문 하고 싶어 / 여행 정보 알려줘 / 호텔 예약해줘

주문, 정보, 예약의 3가지 의도

문장 내 Word검색으로 일일이 파악할 수도 있으나 한계가 있음
ex) 피자 시켜먹고 싶어 / 여행 좋은데 알려줘....

Deeplearning를 활용하면 이런 문제들을 해결 할 수 있음

Char + CNN으로 분류해보자
(CNN - Feature 주문, 정보, 예약)
(Word Similarity 피자, 피자 / 정보, 갈만한데)



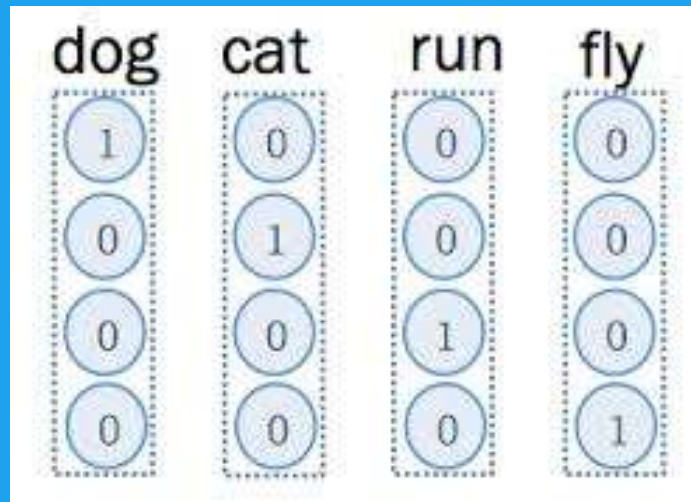
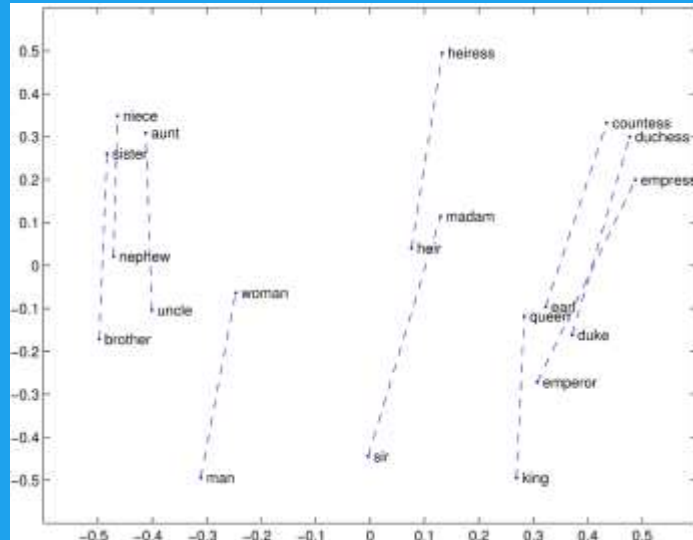
Intent를 알아내는법 (Text Classification - Data 구성)

Word
피자
주문
하고
싶어

Vector가 많다면

영어발음변환
PIJA
JUMUN
HAGO
SIPO

숫자, 특수문자, 공백등
모두 고려해야함



W2V(Pretrained)
피자 (0.12, 0.54, 0.72)
주문(0.56, 0.65, 0.64)
하고(0.67, 0.91, 0.13)
싶어(0.89, 0.14, 0.11)

Ont Hotencoding (Word단위 or 글자단위)

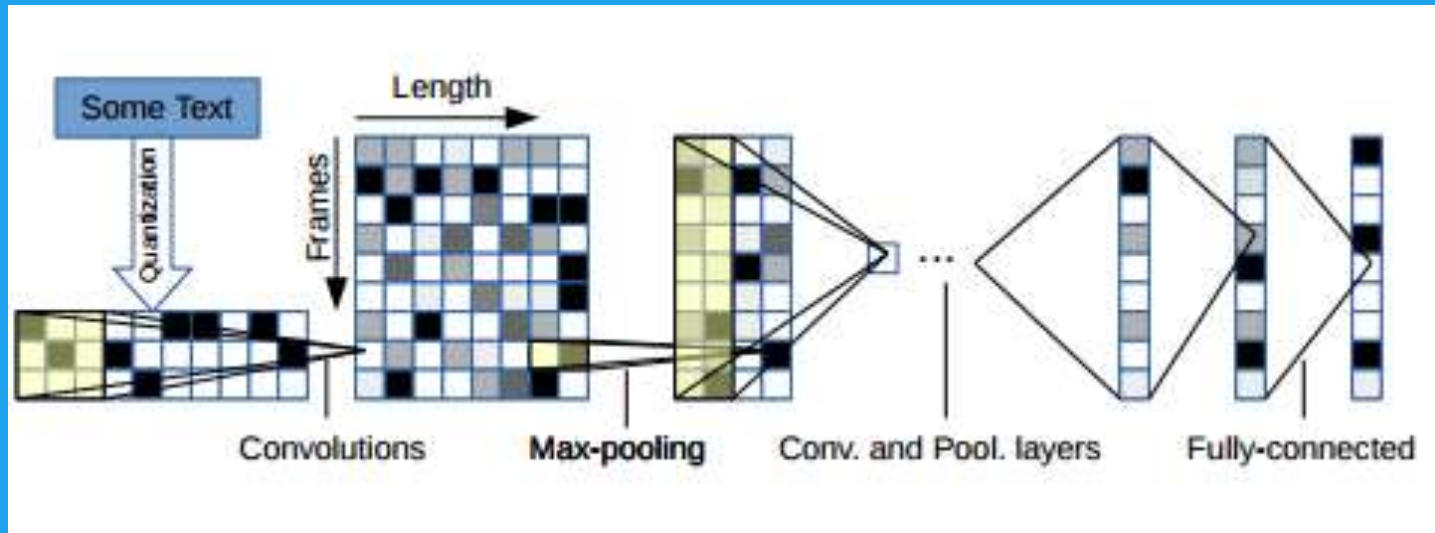
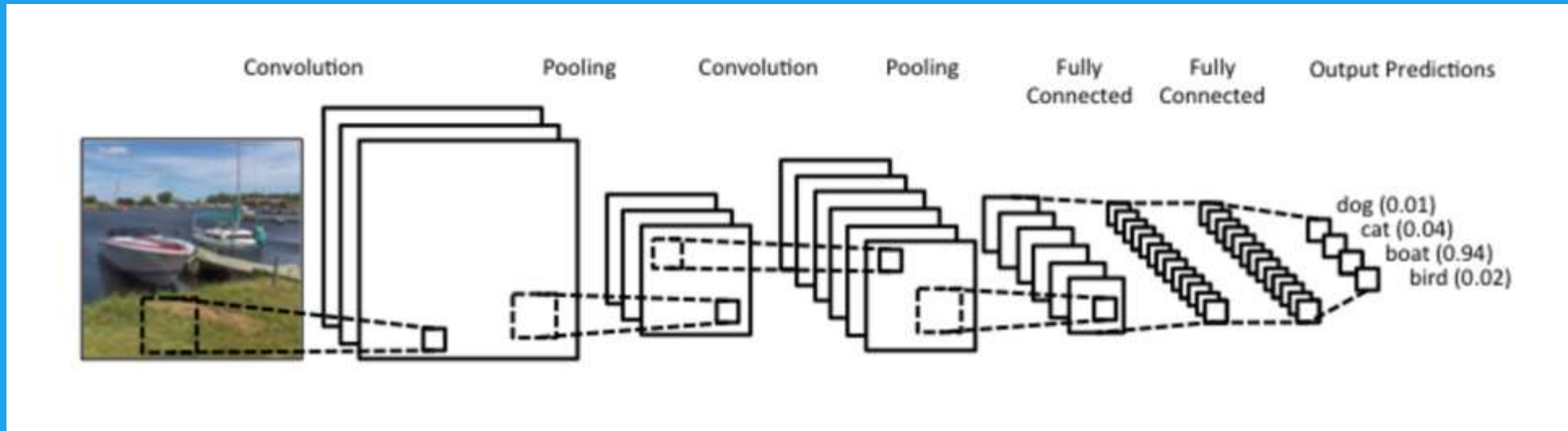
(0100000000)
(0000010000)
(0010000000)
(0000000100)

Ont Hotencoding (A~Z Vector)

(0100000000)
(0000010000)
(0010000000)
(0000000100)



Char CNN?



CNN은 일반적으로 이미지의 특징을 추출하여 인식하는데 많이 쓰이나
이미지도 결국은 Vector이고
텍스트도 Vector를 감안하면
텍스트의 Feature를
뽑아낼 수 있음

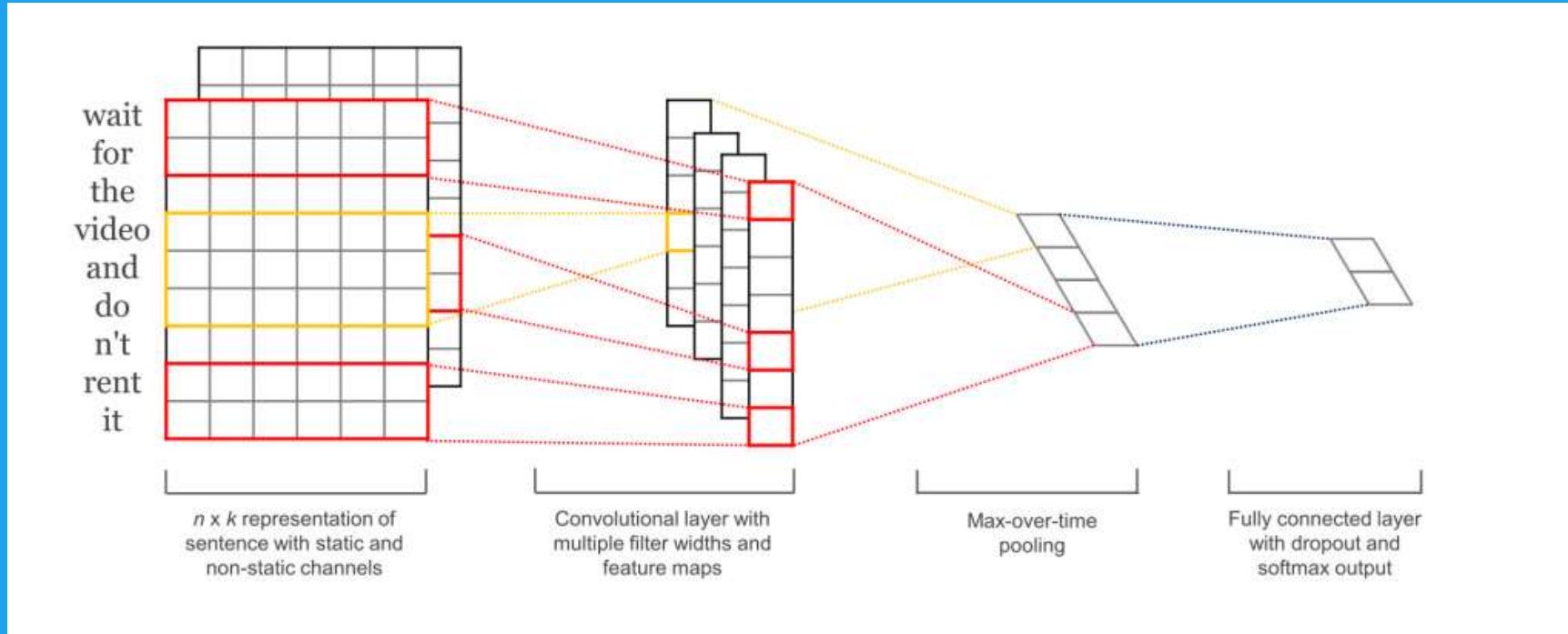


Text Classification - Char CNN

Char-CNN을 활용하여 의도를 파악해보자

지금
피자
주문
하고
싶어

예약
주문
정보



Vector (W2V)
길이/차원/윈도우
Static / Non Static
/ Random

Feature
바라볼 단어수
[3,4,5 filter]

pooling
추상화

classification
분류



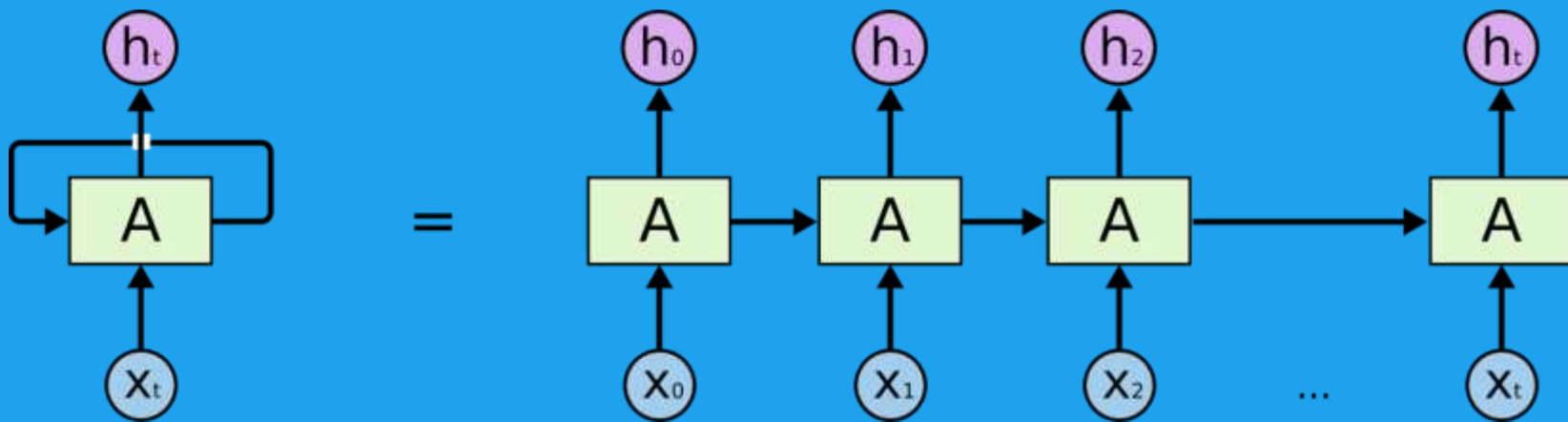
Why Char-CNN??

Char-CNN이 일반적인 다른 알고리즘과 비교하여 압도적 성능을 보임

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	89.6
CNN-non-static	81.5	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	88.1	93.2	92.2	85.0	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	—	—	—	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	—	—	—	—
RNTN (Socher et al., 2013)	—	45.7	85.4	—	—	—	—
DCNN (Kalchbrenner et al., 2014)	—	48.5	86.8	—	93.0	—	—
Paragraph-Vec (Le and Mikolov, 2014)	—	48.7	87.8	—	—	—	—
CCAE (Hermann and Blunsom, 2013)	77.8	—	—	—	—	—	87.2
Sent-Parser (Dong et al., 2014)	79.5	—	—	—	—	—	86.3
NBSVM (Wang and Manning, 2012)	79.4	—	—	93.2	—	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	—	—	93.6	—	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	—	—	93.4	—	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	—	—	93.6	—	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	—	—	—	—	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	—	—	—	—	—	82.7	—
SVM _S (Silva et al., 2011)	—	—	—	—	95.0	—	—



RNN에 대한 이해

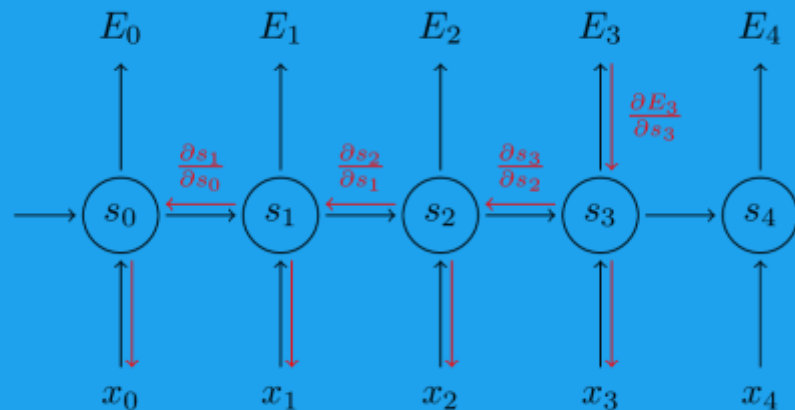


$$h_t = f_W(h_{t-1}, x_t)$$

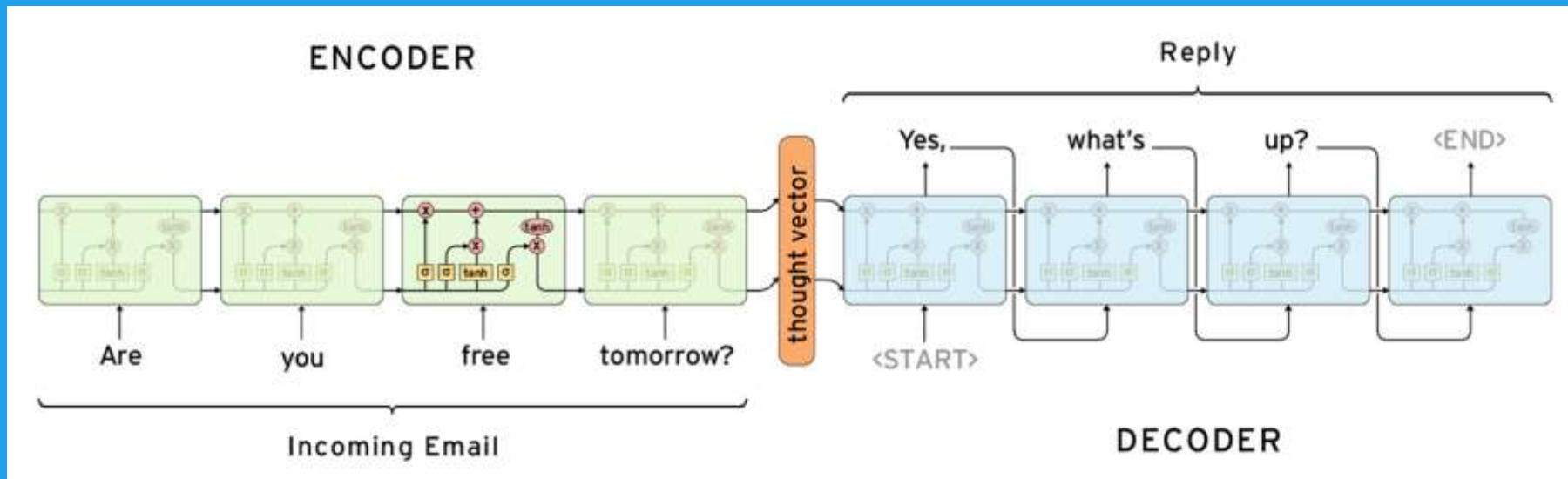
$$h_t = \tanh(W_{hh}h_{t-1} + W_{hx}x_t)$$

$$y_t = W_{hy}h_t$$

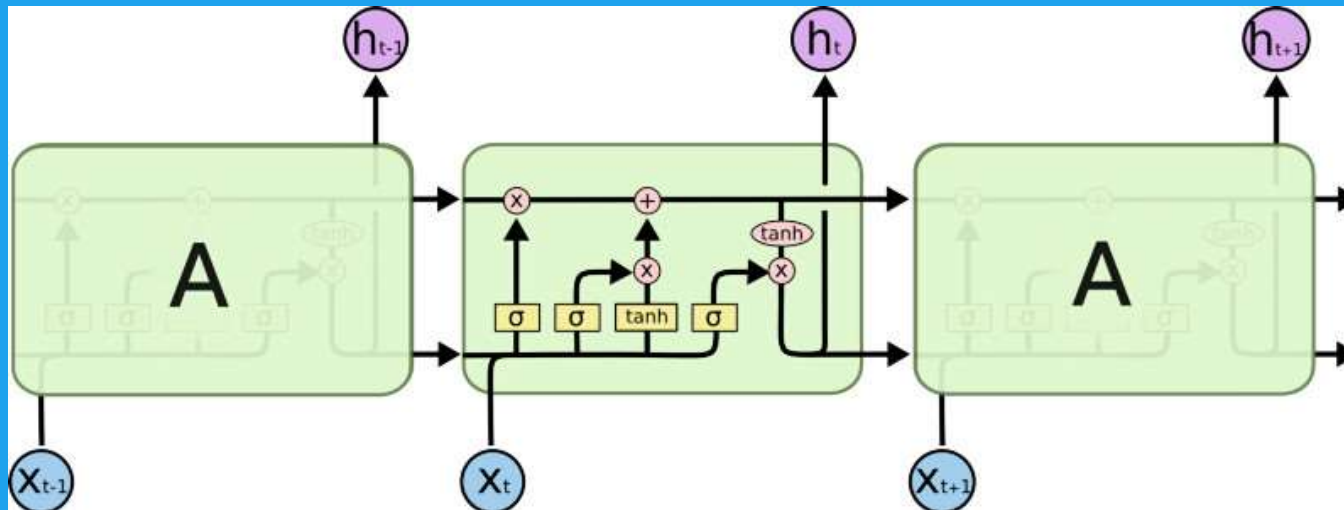
연속된 Data에 대한 모델링에 유용
시퀀스를 입력으로 받기 때문에
Backpropagation을 시간에도 대해서도 수행(BPTT)



Seq2Seq (RNN+RNN) 이해



Chatbot에서는
Generator의 역할
Sentence Generator



영화 자막이나 소설책을 활용하여
학습시킬 수 있음
(형태소 분석기로 input/output정의)



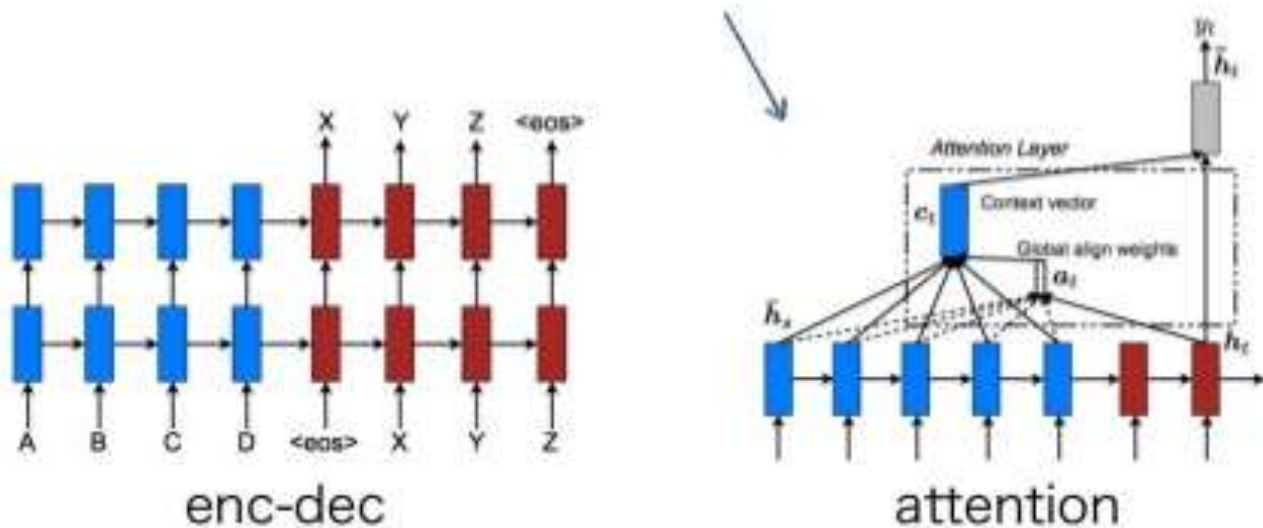
Attention Mechanism 이해

Figures from [Luong+2015] for comparison

Encoder compresses input series into one vector

Decoder uses this vector to generate output

Attention Mechanism predicts the output y_t with a weighted average context vector c_t , not just the last state.



Attention을 통해 Computing Cost를 줄이고 문장의 길이에 따른 복잡도를 줄임

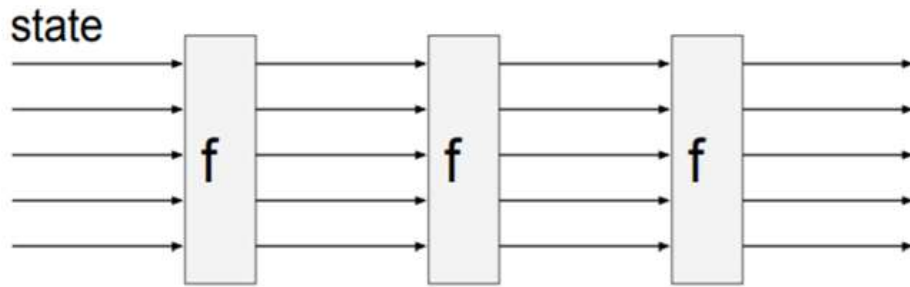
Attention Mechanism predicts the output y_t with a weighted average context vector c_t , not just the last state.

$$a_t(s) = \frac{\exp(\text{score}(h_t, \tilde{h}_s))}{\sum_{s'} \exp(\text{score}(h_t, \tilde{h}_{s'}))}$$
$$c_t = \sum_s a_t(s) \tilde{h}_s$$
$$\tilde{h}_t = \tanh(W_c[c_t; h_t])$$
$$p(y_t | y_{<t}, x) = \text{softmax}(W_s \tilde{h}_t)$$



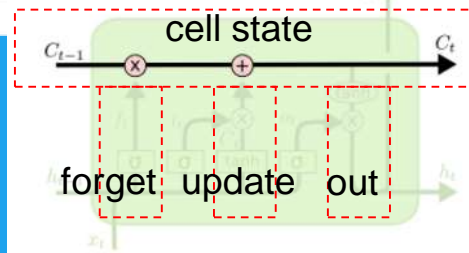
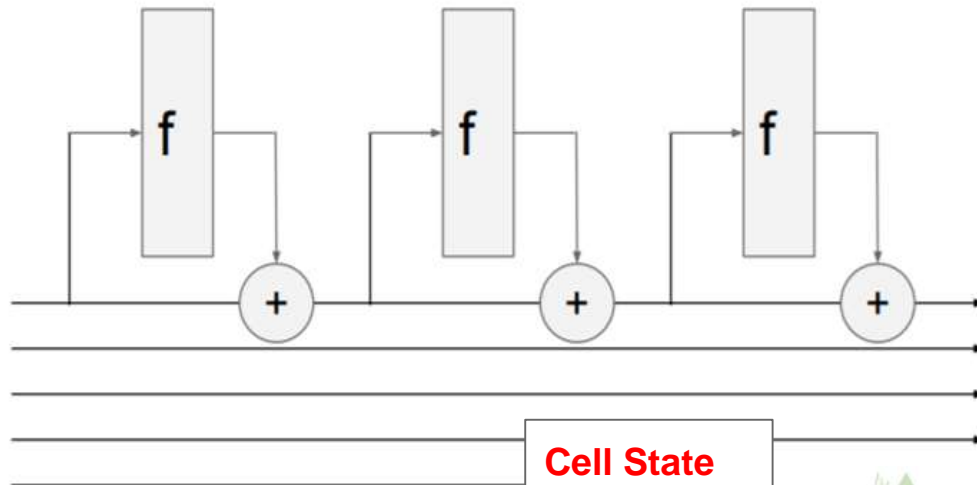
LSTM에 대한 이해

RNN

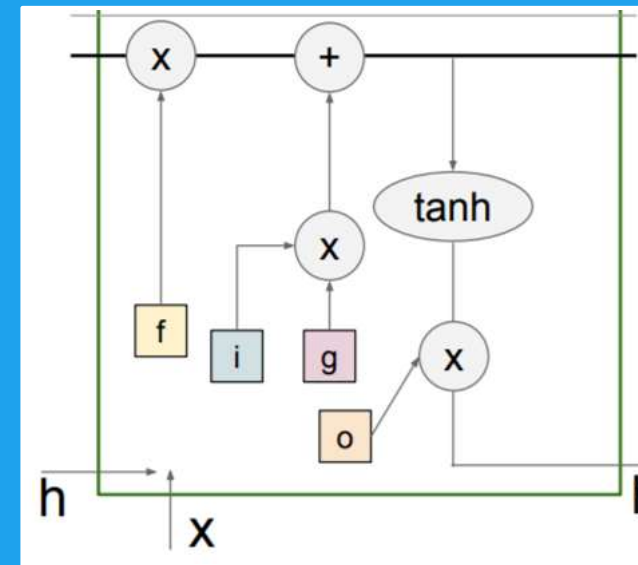


LSTM

(ignoring
forget gates)



<https://brunch.co.kr/@chris-song/9>



RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_{t-1}^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$h \in \mathbb{R}^n$ $W^l [n \times 2n]$

LSTM:

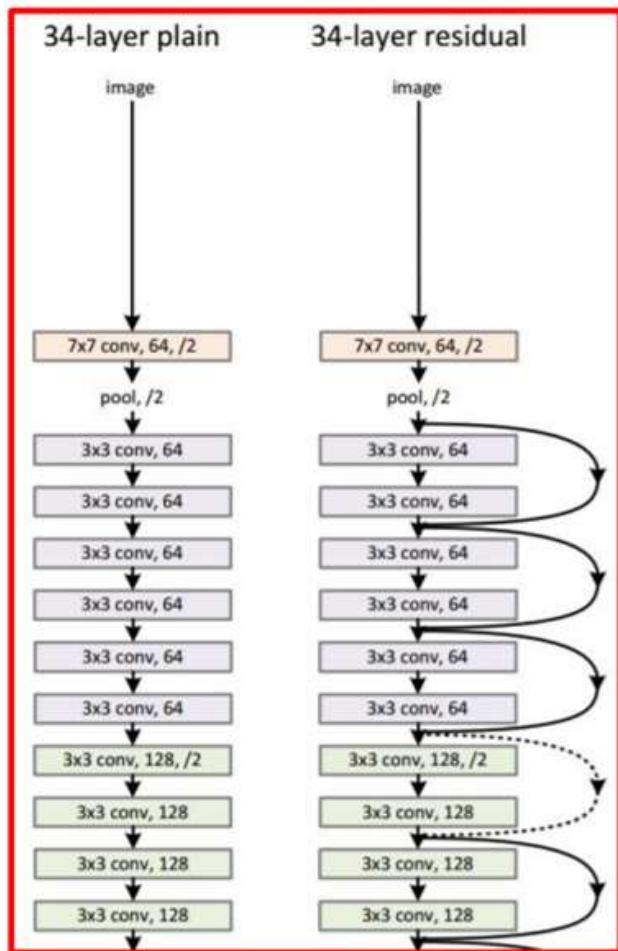
$W^l [4n \times 2n]$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} W^l \begin{pmatrix} h_{t-1}^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

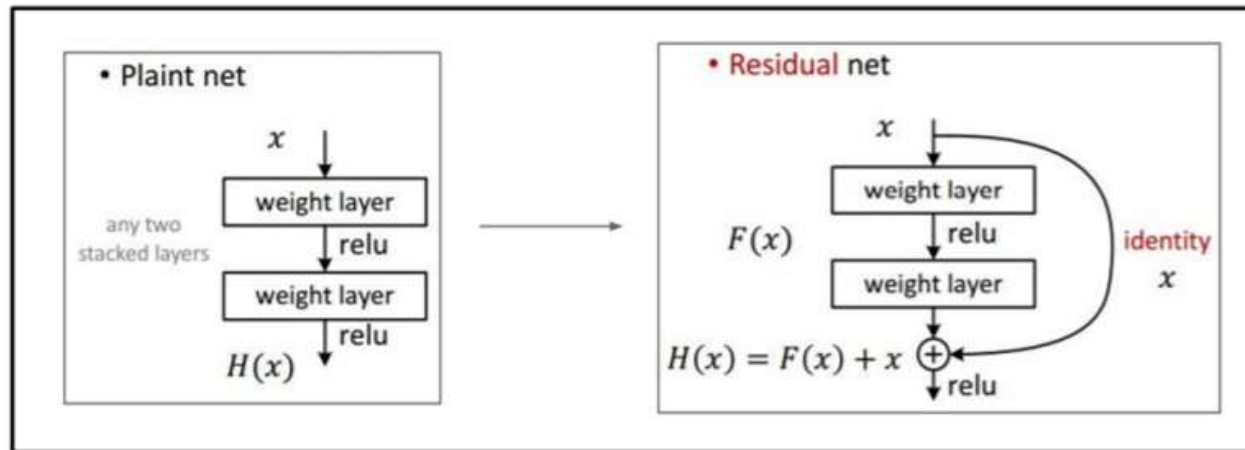
$$h_t^l = o \odot \tanh(c_t^l)$$

ResNet과 RNN의 LSTM은 비슷한 개념



Recall: “PlainNets” vs. ResNets

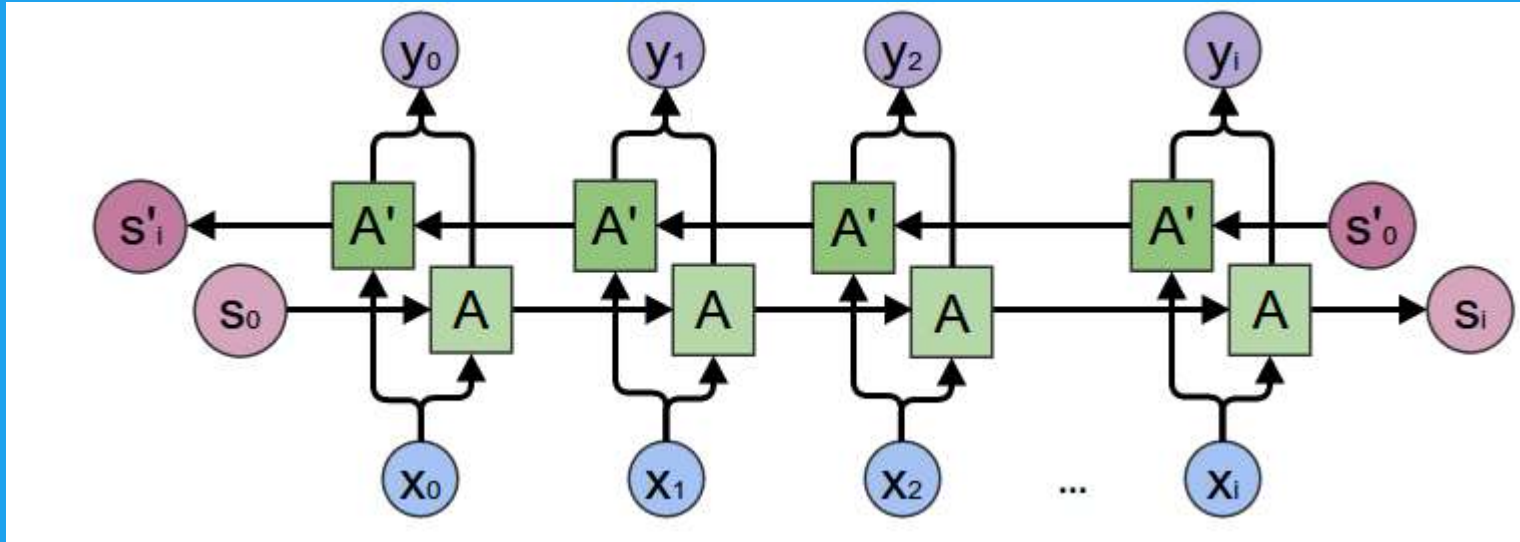
ResNet is to PlainNet what LSTM is to RNN, kind of.



Named Entity Recognition 알아내기

Bidirectional LSTM (양방향 Layer)

- RNN기반의 모델
- 특정위치에 있는 단어의 태깅에 유용



문장내 단어 위치에 따른 의미 처리하는 효과적인 방법



Why Bi-LSTM CRF ?

Table 3: Tagging performance on POS, chunking and NER tasks with only word features.

		POS	CoNLL2000	CoNLL2003
Senna	LSTM	94.63 (-2.66)	90.11 (-2.88)	75.31 (-8.43)
	BI-LSTM	96.04 (-1.36)	93.80 (-0.12)	83.52 (-1.65)
	CRF	94.23 (-3.22)	85.34 (-8.49)	77.41 (-8.72)
	LSTM-CRF	95.62 (-1.92)	93.13 (-1.14)	81.45 (-6.91)
	BI-LSTM-CRF	96.11 (-1.44)	94.40 (-0.06)	84.74 (-4.09)

Table 4: Comparison of tagging accuracy of different models for POS.

System	accuracy	extra data
Maximum entropy cyclic dependency network (Toutanova et al., 2003)	97.24	No
SVM-based tagger (Gimenez and Marquez, 2004)	97.16	No
Bidirectional perceptron learning (Shen et al., 2007)	97.33	No
Semi-supervised condensed nearest neighbor (Soegaard, 2011)	97.50	Yes
CRFs with structure regularization (Sun, 2014)	97.36	No
Conv network tagger (Collobert et al., 2011)	96.37	No
Conv network tagger (senna) (Collobert et al., 2011)	97.29	Yes
BI-LSTM-CRF (ours)	97.43	No
BI-LSTM-CRF (Senna) (ours)	97.55	Yes



Named Entity Recognition 알아내기

B-시작어휘
I-이어지는 어휘
O-어휘아님, 공백(OUT)
U-Unknown
(Word Embedding이 없을시)

피자 주문하고 싶어
B-Pizza B-Order O O

여행 정보 알려줘
B-Travel B-Information O

호텔 예약해줘
B-Hotel B-Reserve O

※New York?,수상하다?



brat를 활용 BIO Tagging



Bi-LSTM으로 사전 강화 -> 모델 학습

피자 주문하고 싶어
B-Pizza B-Order O O



피이자 주문하고 싶어

여행 정보 알려줘
B-Travel B-Info O

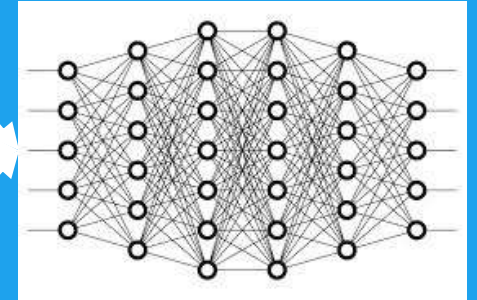


놀러갈 정보 알려줘

호텔 예약해줘
B-Hotel B-Reserve O



숙소 예약해줘



Bi-LSTM을 통해서 신규 어휘를 도출하고 학습Data에 반영하여 모델의 성능을 지속적으로 향상 시킴



엑소브레인 with Ontology



문장별 머신 러닝 기법을
적용하여
자연어를 이해한 후

웹크롤링으로 습득한
지식DB의 정보를 통해
가져옴

더지속적인 학습을 통해 진화

객관식이 더 어려움
지식구축에만 수년...



Ontology with syntaxnet



Ontology구축을 위한
Neo4j DB를 활용

구문분석기의 자체
제작이나 구글의
Syntaxnet를 활용

A는B이다 B는 C이다
추론 : A는 C이다

Neo4j Browser

1 // Get some data
2 MATCH (n) RETURN n LIMIT 100

CYPHER MATCH (n) RETURN n LIMIT 100

Movie [87]

Style

Caption: title

View stylesheet...

PLAY :play movies

Mini-app: The Movie Graph

Actors and their appearances. Click the code to load into the editor, then run it.

```
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})
CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})
CREATE (Carrie:Person {name:'Carrie-Anne Moss', born:1967})
CREATE (Laurence:Person {name:'Laurence Fishburne', born:1961})
CREATE (Hugo:Person {name:'Hugo Weaving', born:1960})
```

your graph-query workbench

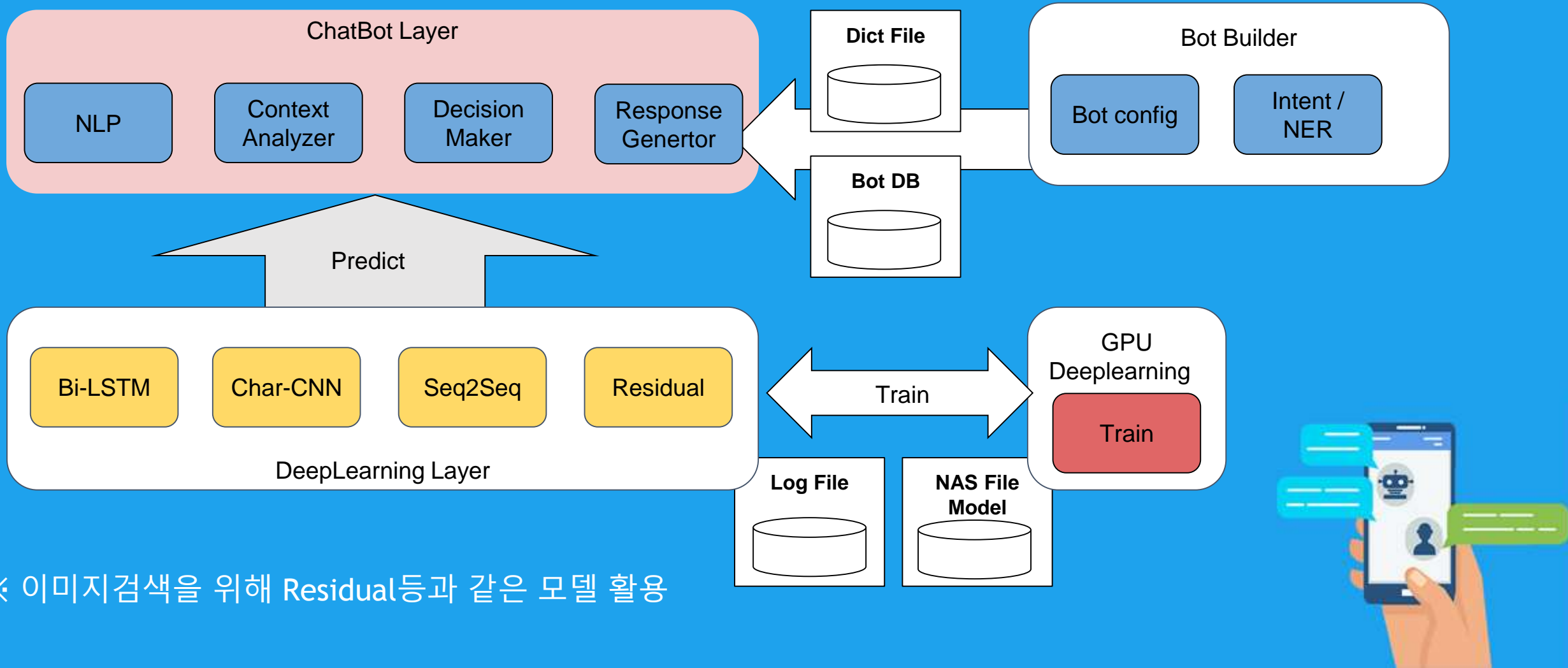


Chatbot Service를 위한 Architecture 구성



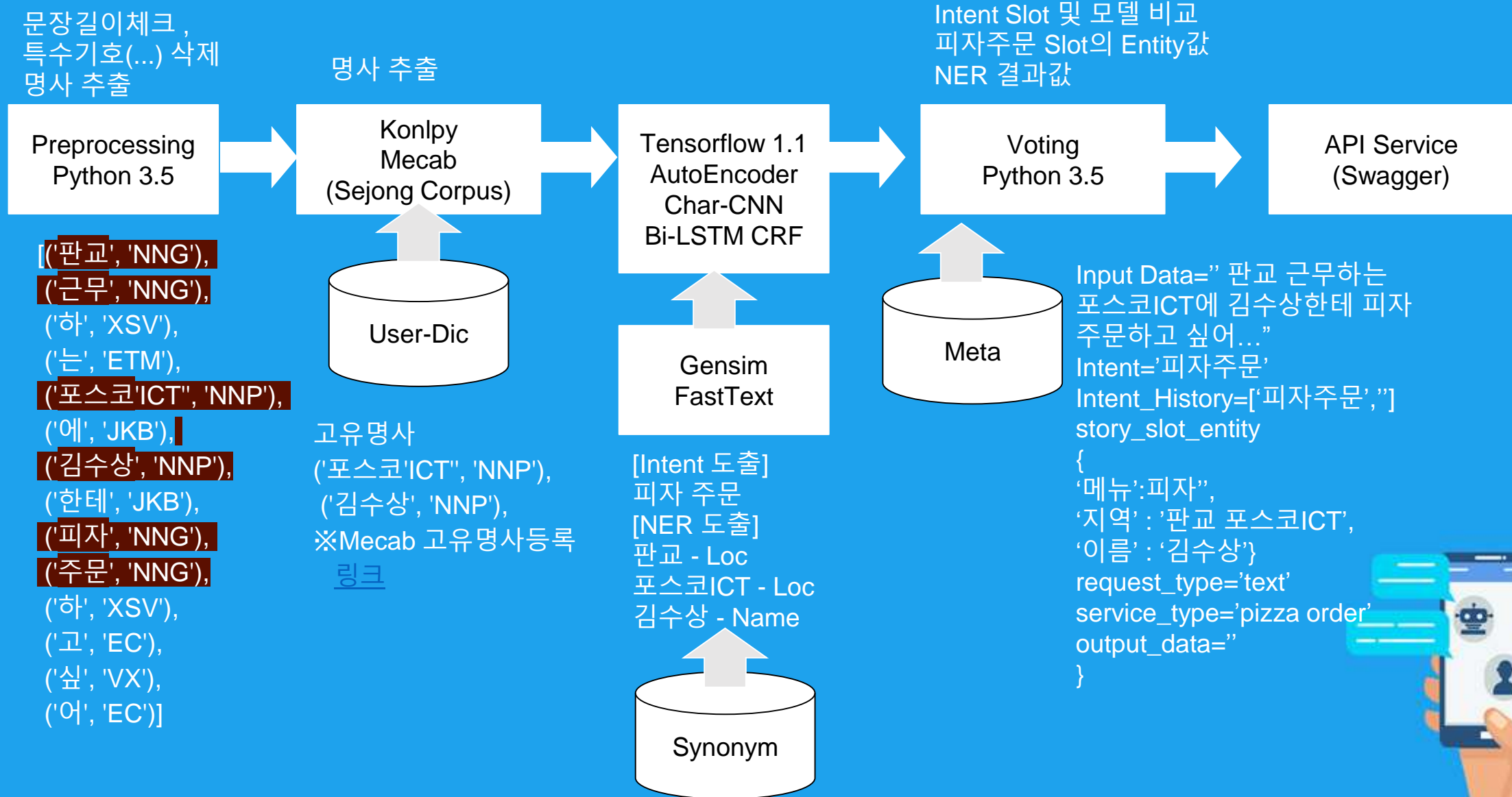
Chatbot Architecture

Deep Learning Layer 위에 ChatBot Layer 와 같은 Application Layer 를 구성하고 각 Application Layer 는 필요한 기능을 DL Layer 와 연동.

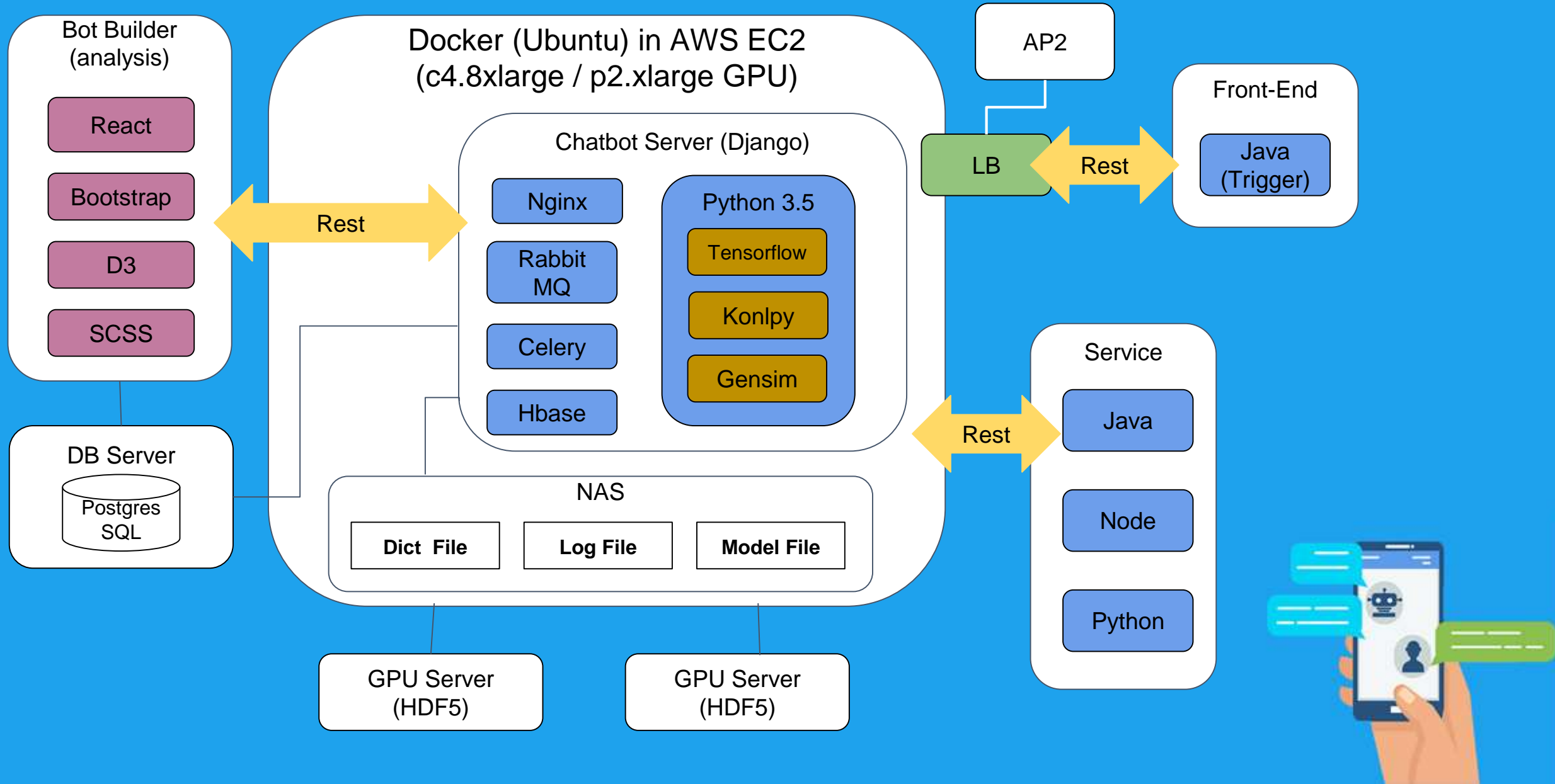


NLP Architecture

판교 근무하는 포스코ICT에 김수상한테 피자 주문하고 싶어...

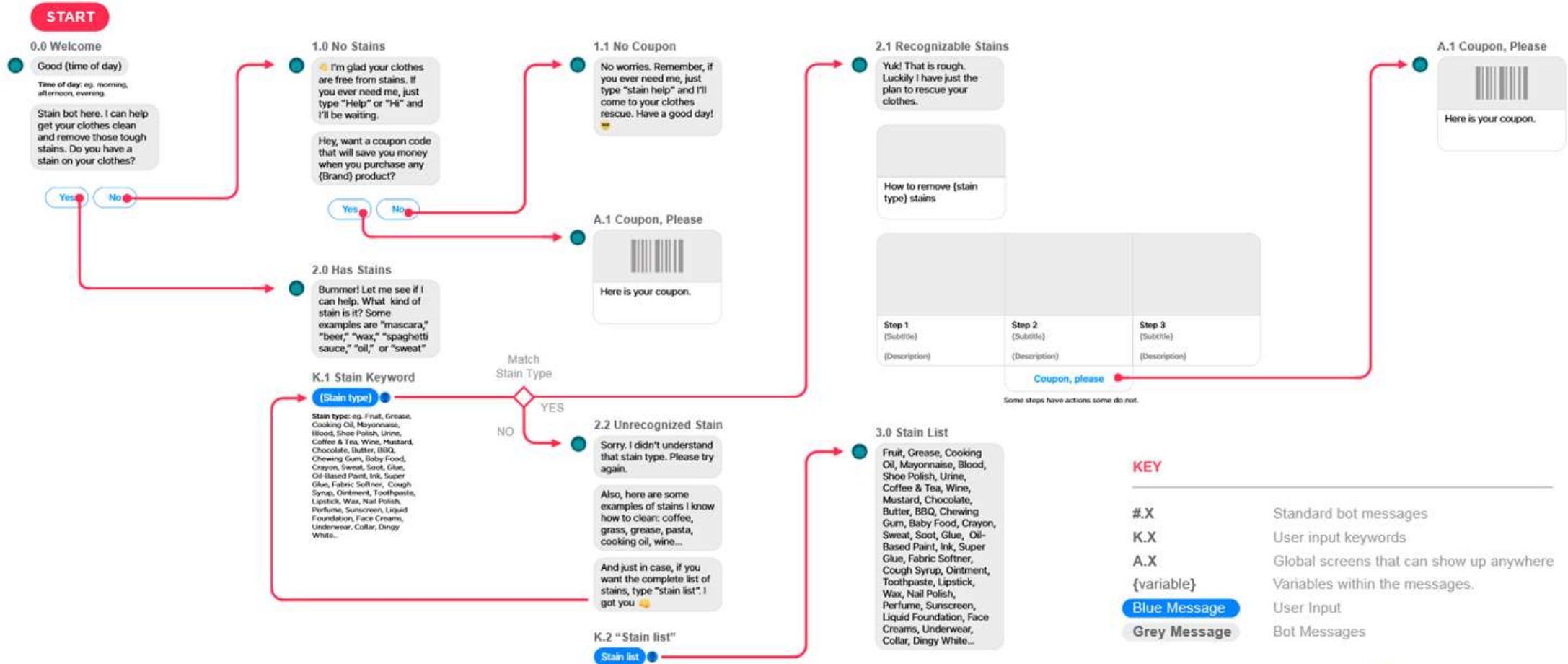


Web Service Architecture



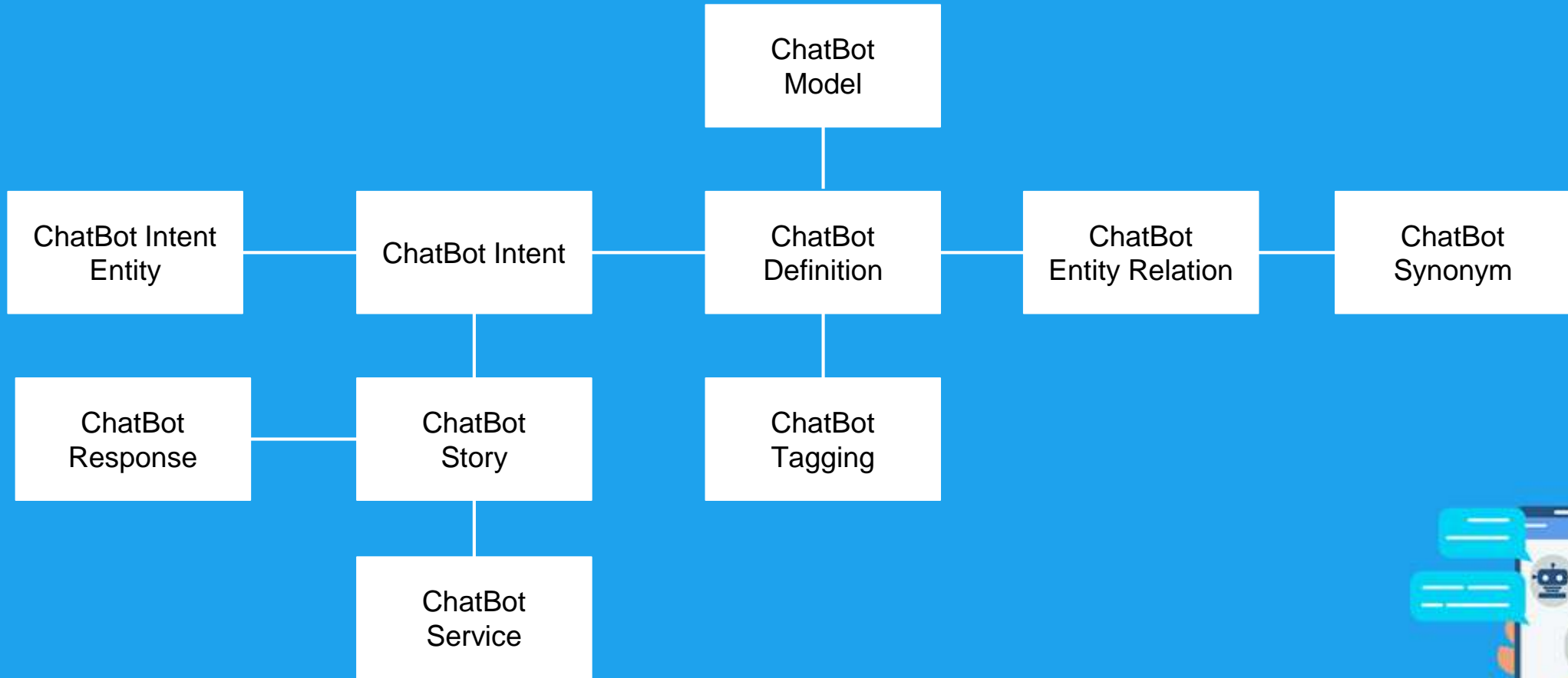
Bot Builder and UX

BOT CONVERSATION FLOW STAIN REMOVAL HELPER



Bot Builder DB

Service의 확대를 위해 가능하면 Common하게 구성



Chatbot API

Client

```
Input Data=페파로니 피자 주문할게  
Intent=""  
Intent_History=["", ""]  
story_slot_entity  
{  
    메뉴:",  
    사이즈:",  
    사이드:"  
}  
request_type=text  
service_type=""  
output_data=""
```

Rest API

Server

```
Input Data= 페파로니 피자 주문할게  
Intent=피자주문  
Intent_History=["피자주문", ""]  
story_slot_entity  
{  
    메뉴:피자,  
    사이즈:라지,  
    사이드: 콜라  
}  
request_type=text  
service_type=""  
output_data=주문완료
```

※ json의 길이가 길어지면 log파일로 관리



Test Codes for Chatbot

Case별 Test Coverage 코드 구현

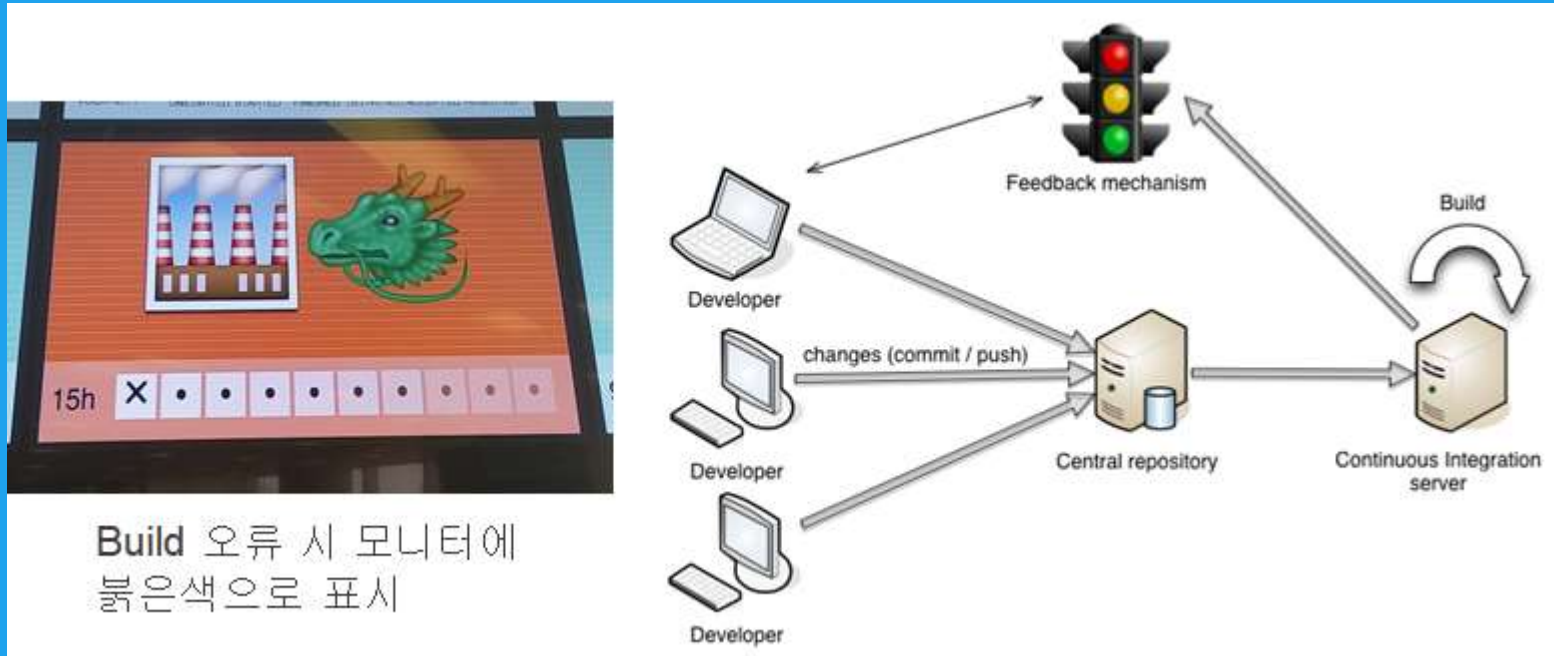
1. 로직 변경 (단위테스트)
2. Model 변경 (Hyper Parameter)
3. Data 변경 (Slot, Dict, Entity, 유의어)
4. 속성 값 변경 (Threshold, Rule기준)

피자주문
호텔예약
Slot점검
여행정보

의도점검->NER점검->

input 판교에 피자주문할게 -> intent : 피자주문
slot : {메뉴,크리,사이드-extra}

단순 로직 변경과는 다르게 Data와 Model의 변경사항을 지속적 검증 할 수 있는 방안 필요
가동상황에서 정확도를 올리기 위해선 Continuous Integration이 필수 (Jenkins / Travis CI등)



실무에서 발생하는 문제와 해결 Tips



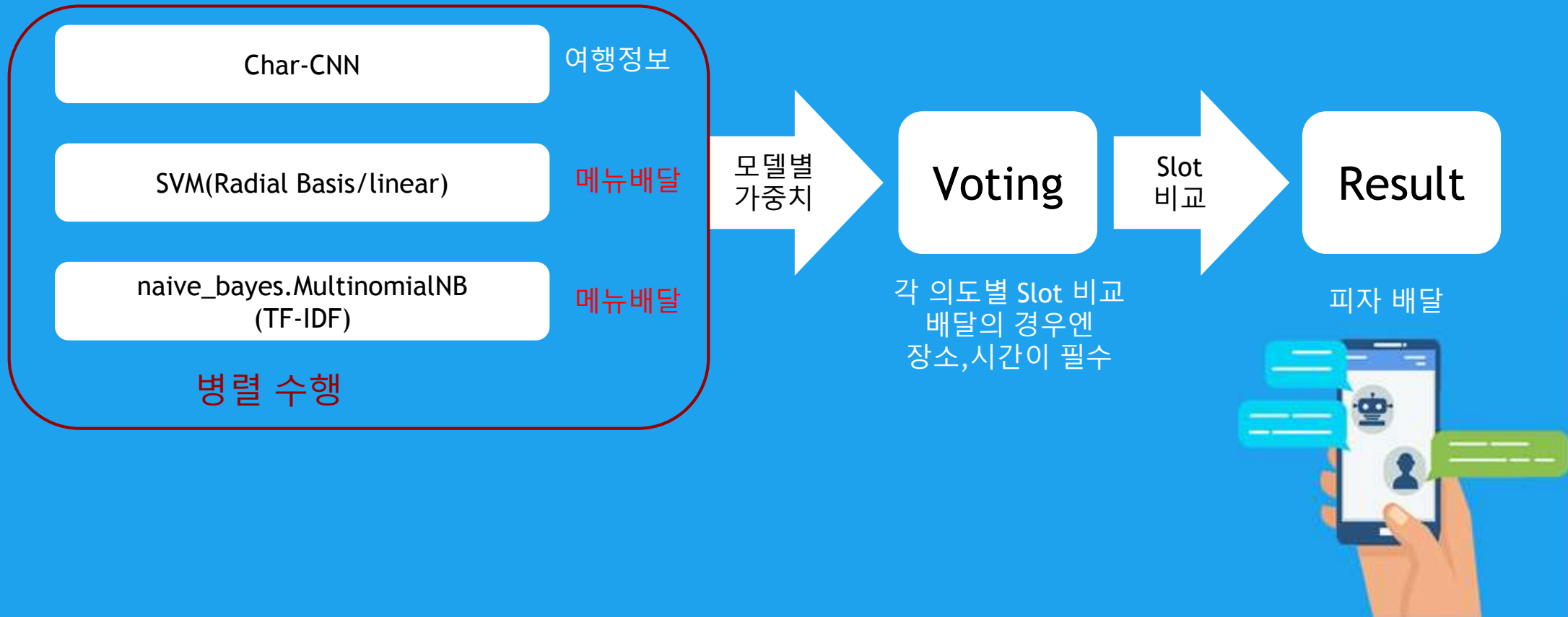
Ensemble and Voting

모델의 정합성을 올리기 위해 복수개의 모델과 로직으로 보완 (Scoring / Voting)

의도를 찾는 경우 여러모델을 비교하여 가장 근접한 값을 찾는다

Textmining과 앙상블의 조합으로 정합도르 올리자

포스코ICT에
지금 피자
배달해줘



Trigger 처리 (사랑, 이미지 검색)

1. 사랑단어가 포함될 경우 <실재 가동 사례>

직원 : XXX 사원에게 **사랑**한다고 포스톡 보내줘

챗봇 : 너무 쉽게 사랑하지 마세요.

직원 : 니가 언제 내 사랑을 논해

챗봇 : 학습중이라 아직 잘 모르는게 많아요.

직원 : ㅋㅋㅋㅋ

챗봇 : ㅋㅋㅋ

2. 이미지 검색 시(ResNet Model Call)



[안녕, 사랑, ㅋㅋㅋ] 등에 Trigger를 적용하고 이에 확보된 Data를 Seq2Seq모델에 학습시켜 NLP전처리 모델로 사용



필요시 Tone Generator을 쓰자

말투를 다르게만듬 (지역별, 존댓말, 부하톤)
주문이 완료되었습니다 (일반)
주문이 완료되었단다 (공손)
주문이 완료되었어요 (존대)
주문이 완료되었다니깐 (짜증)

Seq2Seq Model활용 - Encoder에 명사등 구성
Decoder에 명사+조사 구성

Response Generator의 경우 형태소 분석기의 응용

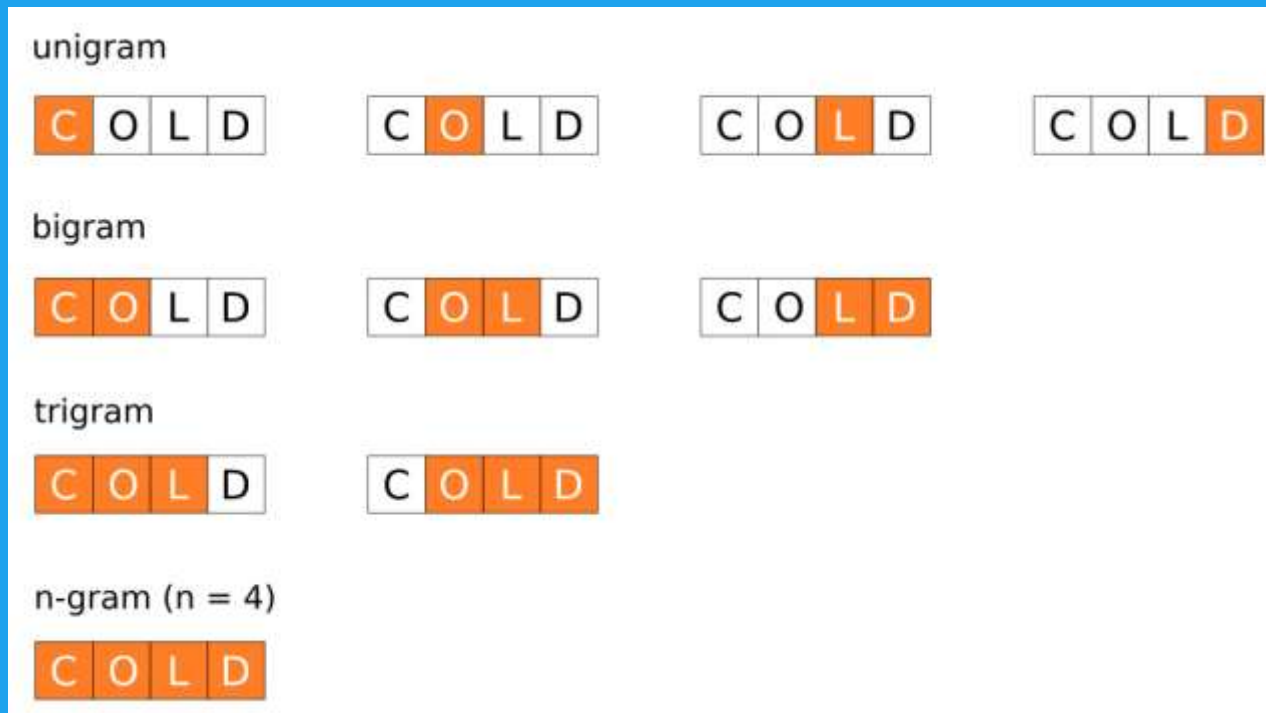


유의어 처리(N-Gram)

페파로니 - Pepperoni, 페파로니, 페파피자..... / Mac Book Pro - 맥프로, 맥북프로...

고객별로 다양한 단어를 사용하나 API호출시에는 지정 값으로 해야 함

N-Gram을 활용하여 유의어로 학습한 결과를 Dict에 찾는 방식 (일반적 trigram)



각 Entity별 N과
Threshold 값을 적절하게 조절

※ threshold :
작을수록 비슷하게 찾음



Response Speed

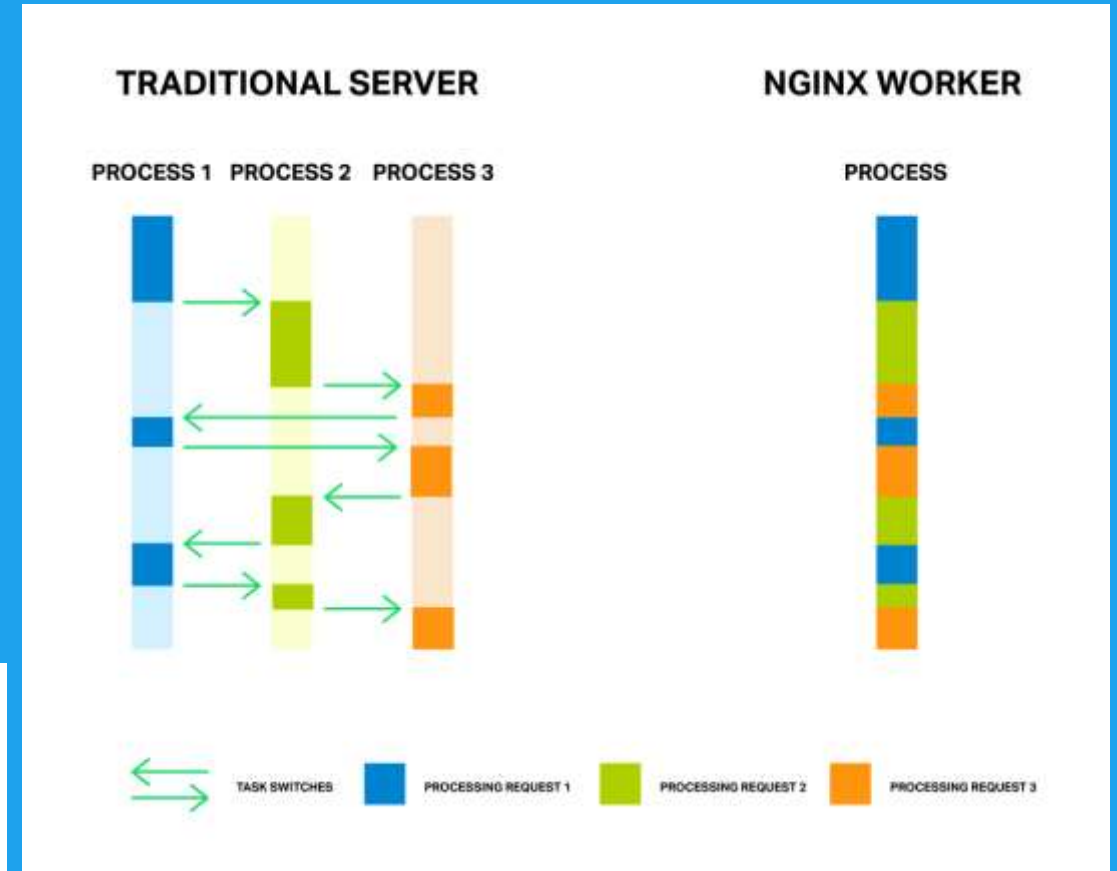
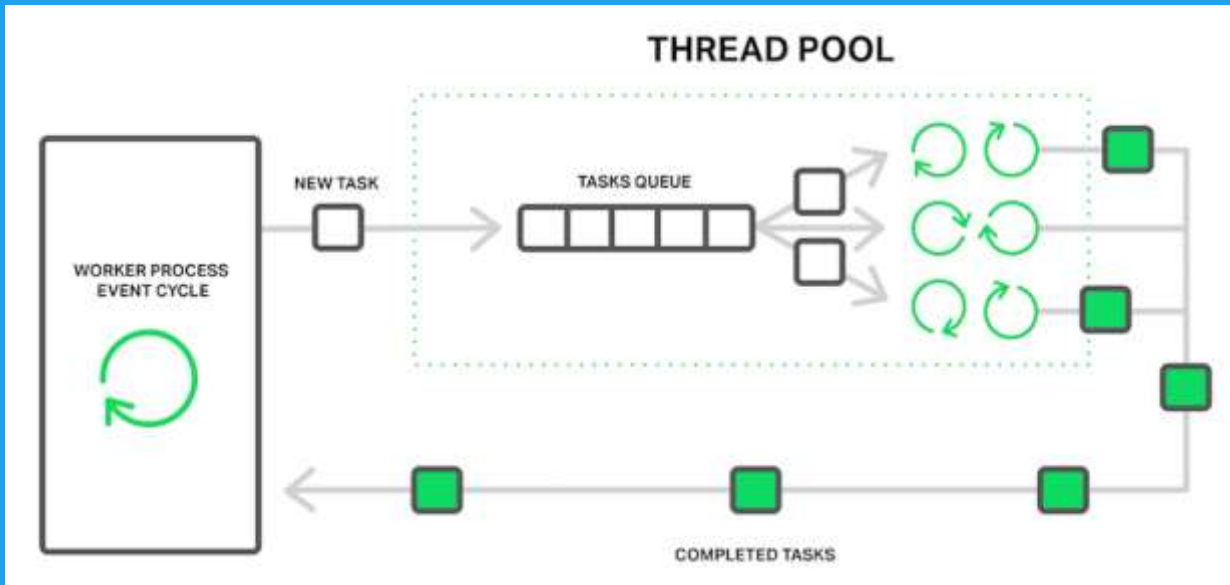
LB 구성

Nginx 사용

적절한 수의 Thread와 AP

Caching of Data (Memory - API사용)

Chatbot에서 수용할수 있는 MAX Time반영



학습시 병렬 처리를 위한 Coding

```
import tensorflow as tf
```

```
with tf.device('/gpu:2'):
```

```
    a = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3], name='a')
```

```
    b = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2], name='b')
```

```
    c = tf.matmul(a, b)
```

```
sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
```

```
print sess.run(c)
```

tf.device를 통해 연산할 Device를 지정
CPU와 GPU의 적절한 분배

```
import tensorflow as tf
```

```
c = []
```

```
for d in ['/gpu:2', '/gpu:3']:
```

```
    with tf.device(d):
```

```
        a = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3])
```

```
        b = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2])
```

```
        c.append(tf.matmul(a, b))
```

```
with tf.device('/cpu:0'):
```

```
    sum = tf.add_n(c)
```

```
# Creates a session with log_device_placement set to True.
```

```
sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
```

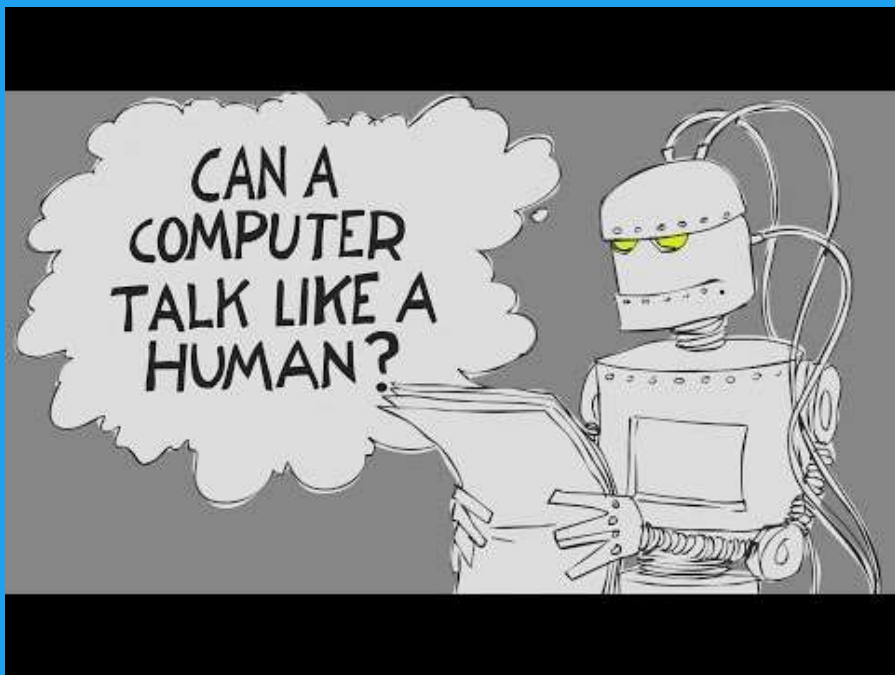
```
print sess.run(sum)
```

Instance Name	GPU Count
p2.xlarge	1
p2.8xlarge	8
p2.16xlarge	16

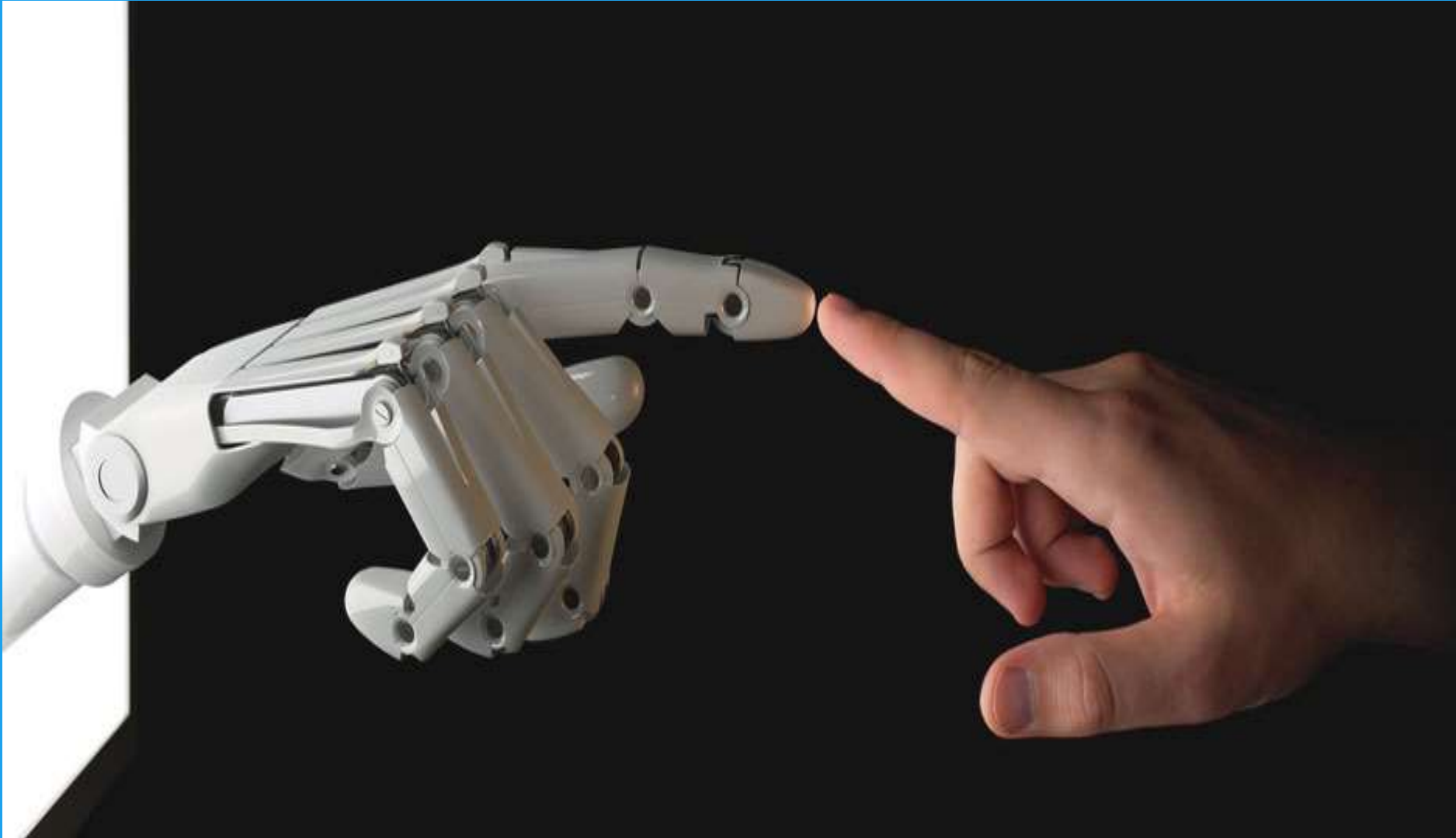
GPU가 많다고 무조건 빠른지는...

마무리

- 챗봇의 구현에 있어서 Hot한 기술의 사용도 중요하지만 무엇보다 Domain별 Data의 의미를 알고 컴퓨터가 잘 이해할 수 있게 해야함
- 학습할 Data와 예측 Data의 패턴을 일치화하는 것이 중요(일관성)
- 딥러닝은 대량의 정제된 Data와 확보가 중요함
- 딥러닝은 성능개선에 있어 충분한 해결 방안이 될 수 있음



When the singularity comes...



Reference

모두를 위한 딥러닝

<http://hunkim.github.io/ml/>

제28회 한글 및 한국어 정보처리 학술 대회

<https://sites.google.com/site/2016hclt/jalyosil>

Stanford University CS231n

<http://cs231n.stanford.edu/>

Creating AI chat bot with Python 3 and Tensorflow[신정규]

<https://speakerdeck.com/inureyes/building-ai-chat-bot-using-python-3-and-tensorflow>

파이썬으로 챗봇_만들기 [김선동]

https://www.slideshare.net/KimSungdong1/20170227-72644192?next_slideshow=1

딥러닝을 이용한 지역 컨텍스트 검색 [김진호]

<http://www.slideshare.net/devview/221-67605830>

Developing Korean Chatbot 101 [조재민]

<https://www.slideshare.net/JaeminCho6/developing-korean-chatbot-101-71013451>

Tensorflow-Tutorials

<https://github.com/golbin/TensorFlow-Tutorials>

