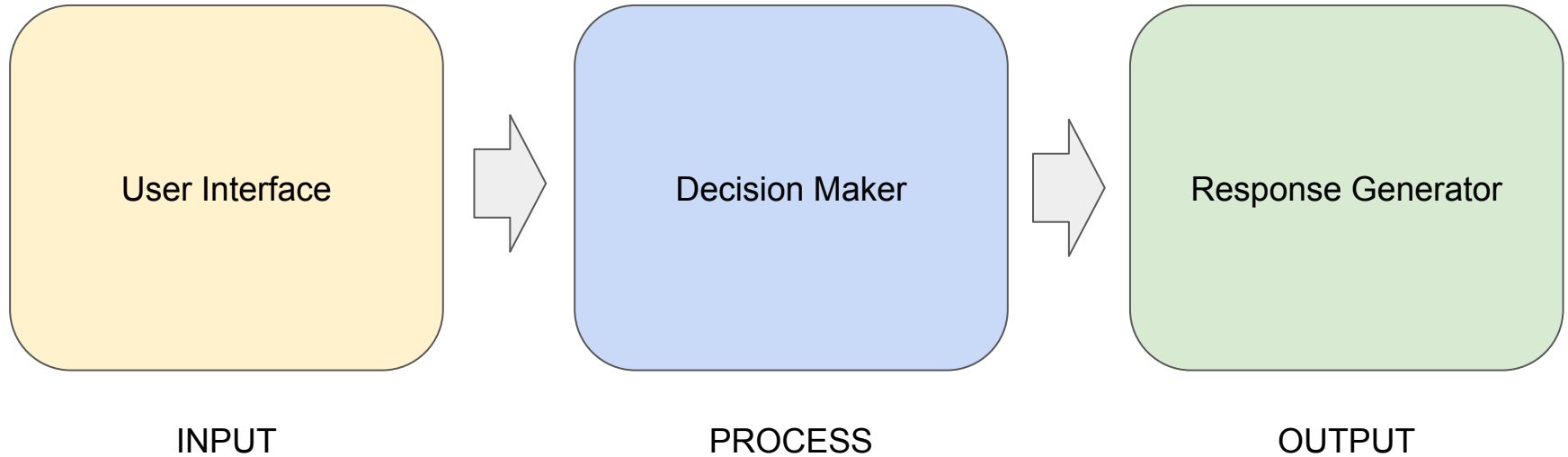


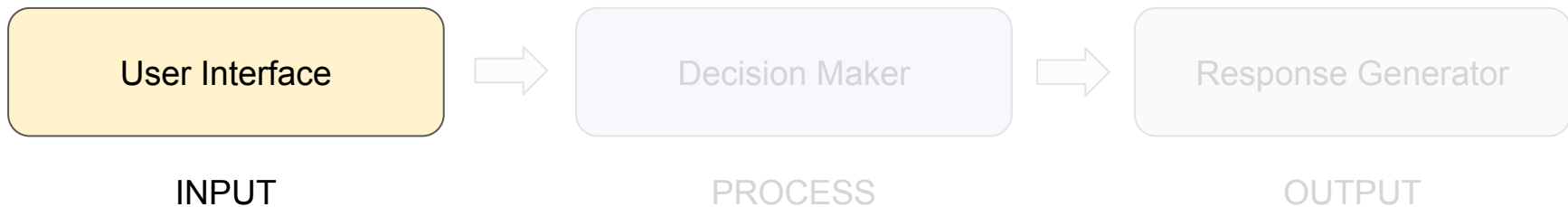
조금 멍청한 지능형 챗봇 만들기 1일차



How to work!



How to work!



사용자로부터 입력을 받아들이는 부분

간단한 명령어/키워드 : #시작, #사진전송

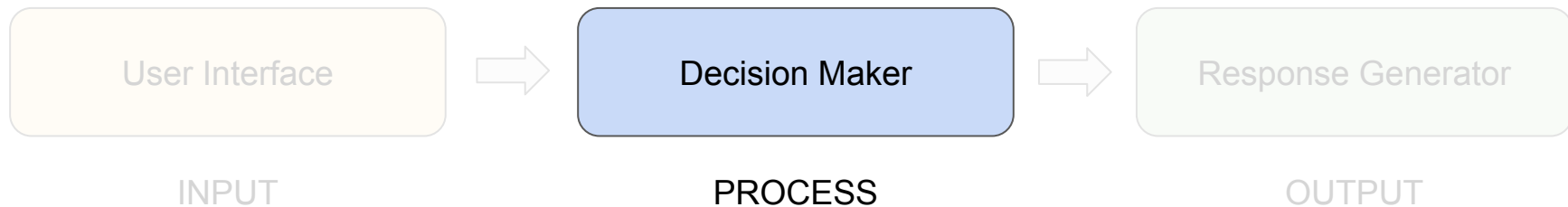
버튼 : 후보 버튼중 선택

자연어 처리 (Natural Language Process)

- 형태소 분석(Morpheme Analyzer)
- Deep Learning Model : Syntax Net, NLU)



How to work!



발화 의도를 파악하고 답변을 만드는 부분

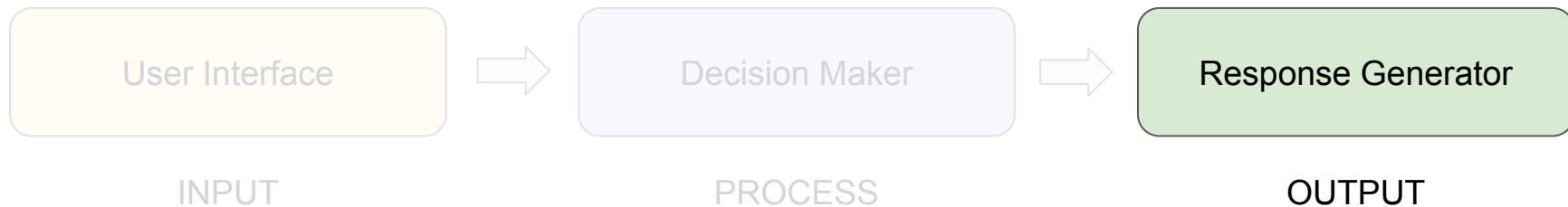
Rule Based: 경남 진주시 날씨 알려줘 [시도] [시군구] [날씨] [알려줘]

추론 (Inference): 오늘 날씨 어때? -> 3. Ask Weather

- Deep Learning Classification



How to work!



사용자에게 답변을 적절하게 가공해서 전달

버튼형 인터페이스

답변중 검색해서 하나 선택

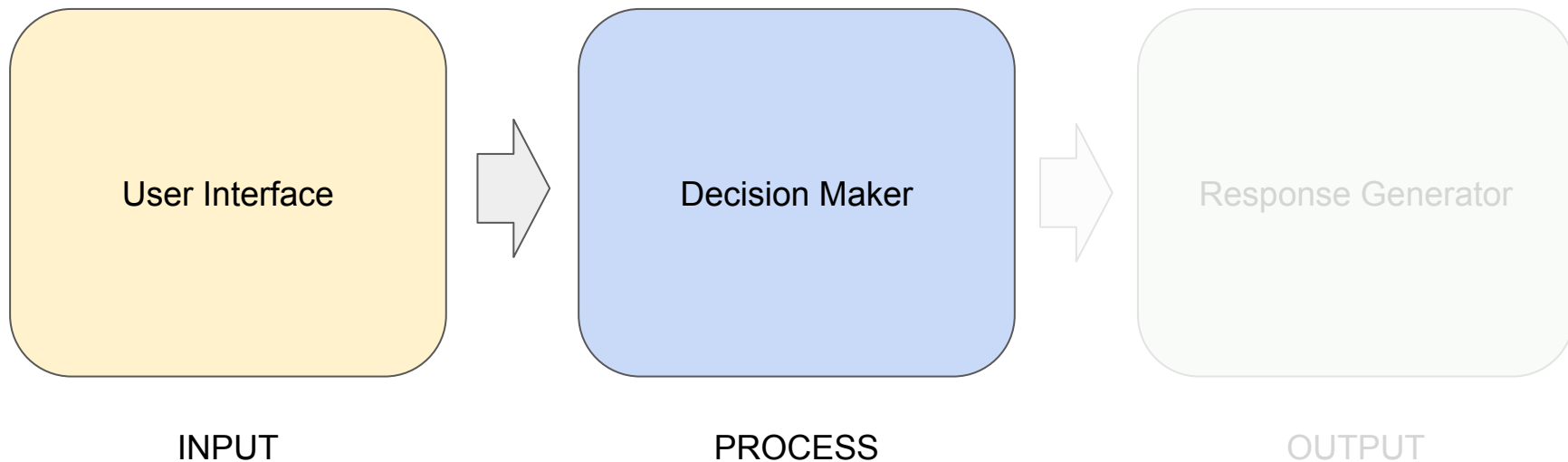
자연어 처리(Natural Language Generation)



단편

풍부

How to work! - 우리에게 중요한건?!



자연어로 입력을 받아서 가장 적절한 답변을 추론(예측) 한다.

So What?

자연어로 입력을 받아서 가장 적절한 답변을 **추론(예측)** 한다.

- 의도 파악(intent)
- 개체 파악(Entity)
- 맥락 고려(Context) : 앞의 내용에는 표시되어있지 않음.

So What?

의도 파악(Intent) - 가장 간단한 방법 Keyword Matching

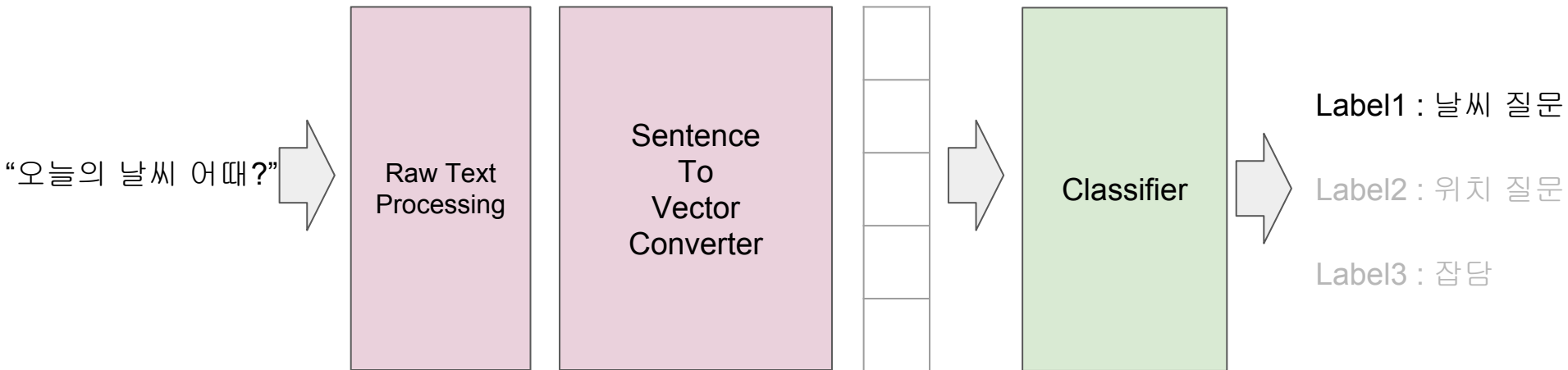
```
def Greeting(sentence):  
    #인사 Intent를 캐치하기 위한 키워드들  
    GREETING_KEYWORDS = ["ㅎㅇ", "하이", "안녕하세요", "하임"]  
  
    #후보 답변  
    GREETING_RESPONSES = ["ㅎㅇ", "반갑다", "안녕하세요"]  
  
    # 키워드를 캐치하면 답변을 한다  
    for word in sentence:  
        if word.lower() in GREETING_KEYWORDS:  
            return random.choice(GREETING_RESPONSES)
```

사용자의 어휘는 예상하는 것보다 훨씬 풍부할 수 있는데, 미리 키워드가 정의되어 있지 않으면 의도(Intent)를 캐치할 수 없다.

확실하지만 시간과 노력이 많이 든다. (Rule Based 일종)

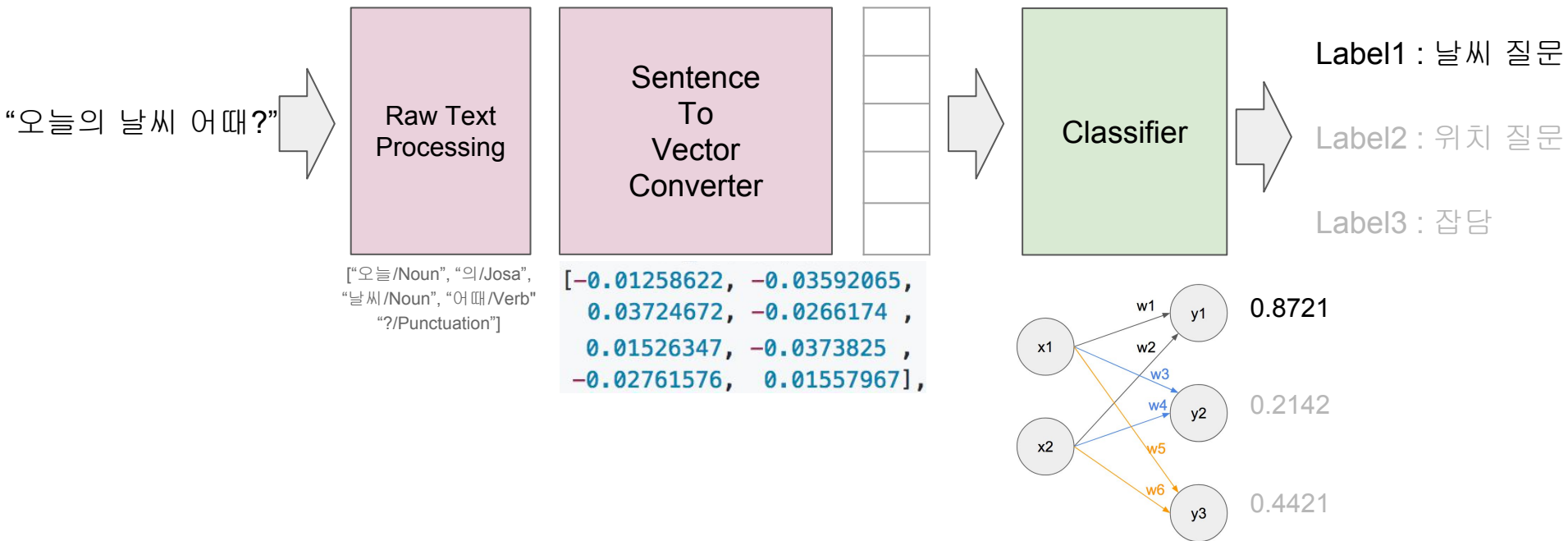
So What?

의도 파악(Intent) - 더 나은 방법 Intent Classifier



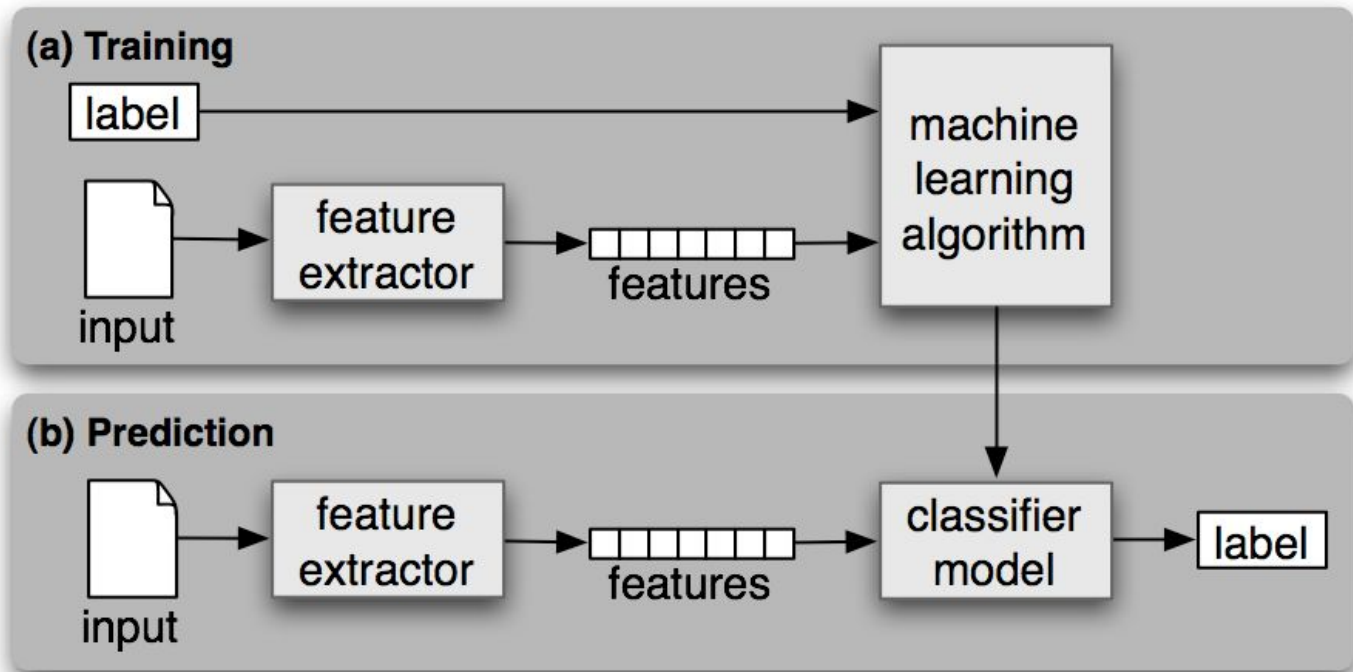
So What?

의도 파악(Intent) - 더 나은 방법 Intent Classifier



So What?

의도 파악(Intent) - 학습 및 추론 프로세스



So What?

의도 파악(Intent) - 테스트

목적: 긍정적인 영화리뷰인지 아닌지 분류한다.

사용 라이브러리 : konlpy, gensim.doc2vec, 등등..

학습 데이터: 영화리뷰 15만개의 자료

id	document	label
9976970	아 더빙.. 진짜 짜증나네요 목소리	0
3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1
10265843	너무재밌었다그래서보는것을추천한다	0
9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0

학습시간: 10~15분?

테스트: 아무 문장(단어 배열)이나 대충 생각해서 넣었을 때 얼추 파악이 됐다.

```
print(classifier.predict([doc_vectorizer.infer_vector(["김기철/Noun", "재미/Noun", '없다/Noun', '개판/Noun'])]))
print(classifier.predict([doc_vectorizer.infer_vector(["별로다/Noun", "재미없다/Noun", '쓰레기/Noun', '발연기/Noun'])]))
print(classifier.predict([doc_vectorizer.infer_vector(["재밌다/Noun", "추천/Noun", '영화/Noun', '최고다/Noun'])]))
print(classifier.predict([doc_vectorizer.infer_vector(["꿀잼/Noun", "명연기/Noun", '잼/Noun', '액션/Noun'])]))
print(classifier.score(test_words, test_tags))
```

```
['0']
['0']
['1']
['1']
```

So What?

개체 파악(Entity)

사용자

{오늘/Date} 날씨 어때?

어디건, {서울/Location}이지

챗봇

Ask Weather(Loc=?, Date=오늘)
어디요?

Ask Weather(Loc=서울, Date=오늘)
오늘 서울 날씨는 맑음입니다.

의도(Intent)를 파악하고 답변에 필요한 개체(Entity)가 모두 충족될 때 까지 질문한다.
모두 충족되면 API Call을 해서 필요한 정보를 전달한다.

So What?

맥락 고려(Context)

EX) 레스토랑 Slot filling (Context 관리 안될 경우)

B> 안녕하세요. 무엇을 도와드릴까요?

U> 오늘 저녁 7시에 6명으로 이탈리아 레스토랑 찾고 있어요.

B> 알겠습니다. 가격대는 어느 정도로 생각하고 계세요?

U> 1인당 2만원 이하요.

B> 시간대를 알려주시겠어요?

U> 오늘 저녁 7시라니깐요.

B> 알겠습니다. 더 추가할 사항은 없으신가요?

U> 위치는 어디로 하시려구요?

B> 죄송해요. 못찾겠어요.

U> ::

Entities

LOCATION : NULL

PRICE_RANGE : 2만원

NUM_OF_PEAPLE : 6

TYPE_OF_CUISINE : 이탈리아

TIME : NULL

Review

챗봇에서..

1. 사용자의 발화 의도(Intent)를 이해하는 부분을 이해하고 DeepLearning 기반으로 간단하게 테스트해볼 수 있었다. (다행히 좋은 데이터와 코드가 있었다;;)
2. 개체(Entity)를 파악하는 부분은 개념만 살펴봤고 조금 더 공부가 필요하다.
3. 문맥(Context)을 파악하는 부분은 감이 잘 오지 않는다. 공부가 더 필요하다.
닉이 하고 있는 State Machine 부분과 비슷해 보인다.

참고자료:

<https://rare-technologies.com/doc2vec-tutorial/>

<https://www.slideshare.net/KimSungdong1/20170227-72644192>

<https://www.lucypark.kr/slides/2015-pyconkr/#41>

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

<https://github.com/linanqiu/word2vec-sentiments/blob/master/word2vec-sentiment.ipynb> 기타 등등..