

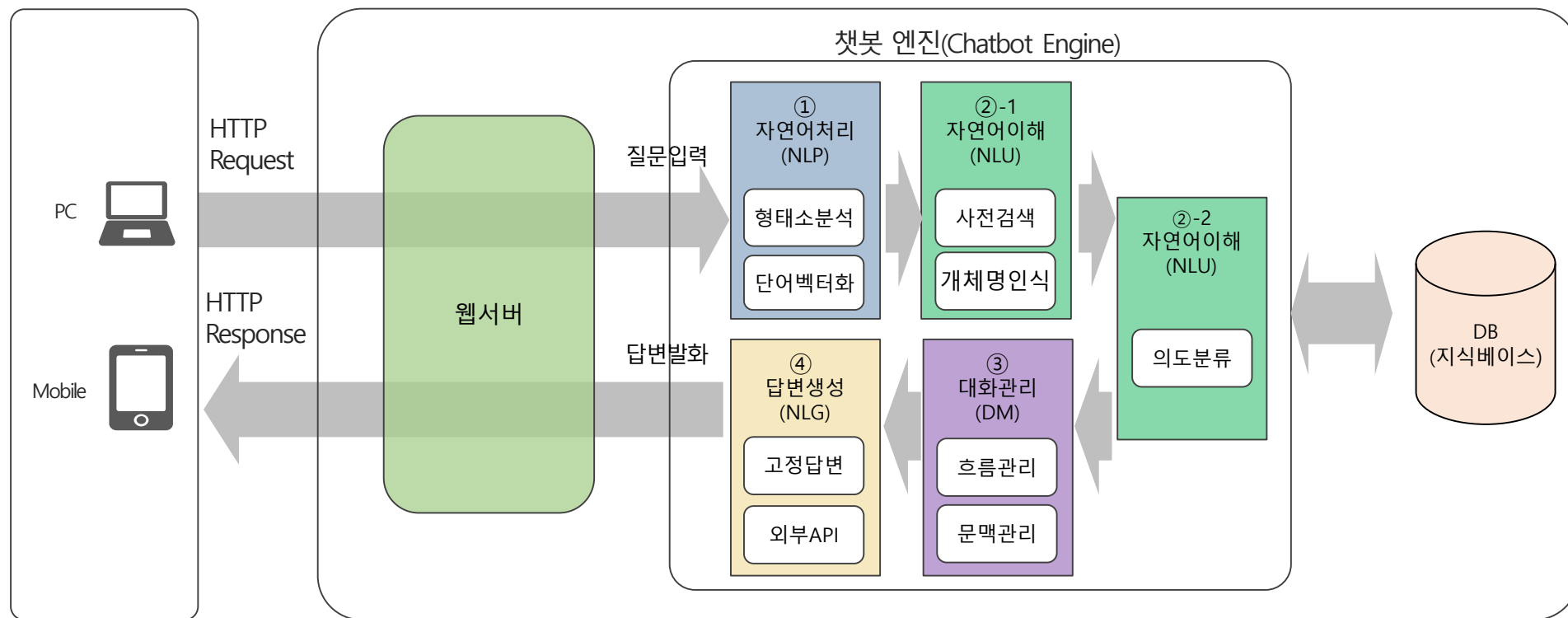
챗봇 시스템은 챗봇 클라이언트, 챗봇 서버로 구성

챗봇 클라이언트는 상용 메신저(카카오톡, 텔레그램 등)를 사용하거나 챗봇 전용 메신저를 직접 개발하여 사용 가능

챗봇 서버는 웹서버(Apache, NGINX 등)와 챗봇 엔진(Chatbot Engine), 지식베이스(Knowledge Base) 등으로 구성

챗봇 클라이언트(Chatbot Client)

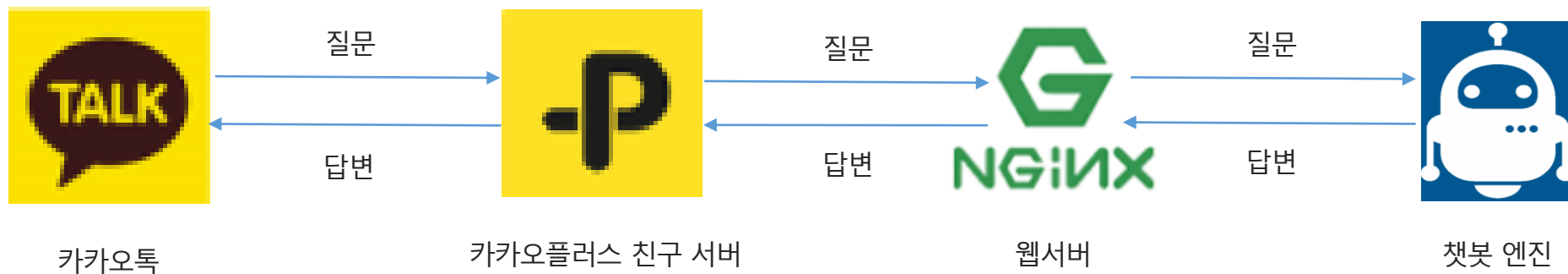
챗봇 서버 (Chatbot Server)



챗봇 클라이언트로 카카오톡 메신저를 사용하기 위해 카카오플러스 친구 서버 신청

카카오톡에서 입력한 질문은 카카오플러스친구 서버에서 챗봇서비스를 위한 웹서버와 챗봇 엔진으로 전달

챗봇 엔진은 다양한 처리 후 답변을 웹서버, 카카오플러스친구 서버를 통해 카카오톡으로 전달



https://center-pf.kakao.com/ 에 접속하여 채널 관리자 가입
관리자 이름, 휴대폰 인증 및 이용약관 동의 후 가입 신청

카카오톡 채널 관리자 가입하기

채널 운영을 위한 관리자 정보를 입력해주세요.

• 이름 ②

김복주

• 전화번호

대한민국(82)

인증

채널 중요 공지 및 알림을 받기 위한 용도로만 사용됩니다.

전체 동의

☐ 모두 확인, 동의합니다.

채널 이용약관, 개인정보 수집 및 이용동의, 개인정보 취급 위탁동의, 채널 추가 동의 (선택)에 모두 동의 합니다.

카카오톡 채널 이용약관

☐ 이용약관에 동의합니다. (필수)

제1장 총칙

제1조 목적

본 약관은 서비스 이용자가 주식회사 카카오(이하 "회사"라고 합니다)가 제공하는 카카오톡 채널 서비스(이하 "채널" 또는 "채널 서비스", "서비스"라 합니다)에 회원으로 가입하고 이를 이용함에 있어 회사와 회원(본 약관에 동의하고 회원등록을 완료한 서비스 회원을 말합니다. 이하 회원이라고 합니다)의 권리·의무 및 책임사항을 규정함을 목적으로 합니다.

개설할 채널의 이름, 검색용 아이디 입력 후 채널 개설

채널 개설하기

카카오톡에서 고객을 만나는 방법! 지금 무료로 만들어 새로운 비즈니스를 경험하세요.

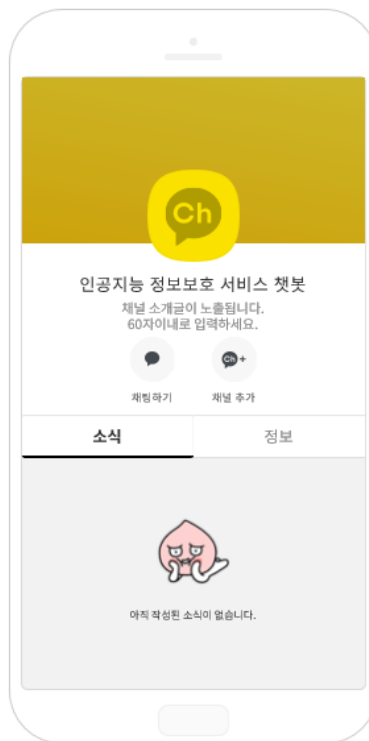
[정보 입력 가이드](#)

프로필 설정

프로필 사진	권장 사이즈 : 640 x 640px / 지원파일 : jpg, jpeg (최대 10MB)	업로드할 파일 선택	첨부
배경 사진	권장 사이즈 : 800 x 346px / 지원파일 : jpg, jpeg (최대 10MB)	업로드할 파일 선택	첨부
채널 이름	인공지능 정보보호 서비스 챗봇		
검색용 아이디	chatbot		
소개글	채널 홈에 노출될 소개 문구를 작성해주세요 (60자 이내)		

부가 정보 설정

카테고리1	IT	인터넷/컴퓨터
카테고리2	카테고리 선택	카테고리 선택



인공지능 정보보호
서비스 챗봇

월렛 상태 ● 미사용
총 캐시 0원 >
유상 0원 >
무상 0원 >
메시지 이용권 0개 >

대시보드

1:1채팅 +
스마트채팅 +
메시지 +
스마트메시지 Beta
친구그룹관리 +
포스트

비즈니스 채널

비즈니스 채널로 전환하여 신뢰도를 높이고 다양한 추가 기능을 사용해보세요.

혜택 4. 비즈메시지 사용 알림톡, 친구톡, 상담톡 사용 가능

비즈니스 채널은 이 모든 비즈메시지
사용이 가능해집니다!

[비즈메시지 자세히보기](#)

신청하기

4/5

최근 7일 비즈니스 성과 (5.23 ~ 5.29) ?

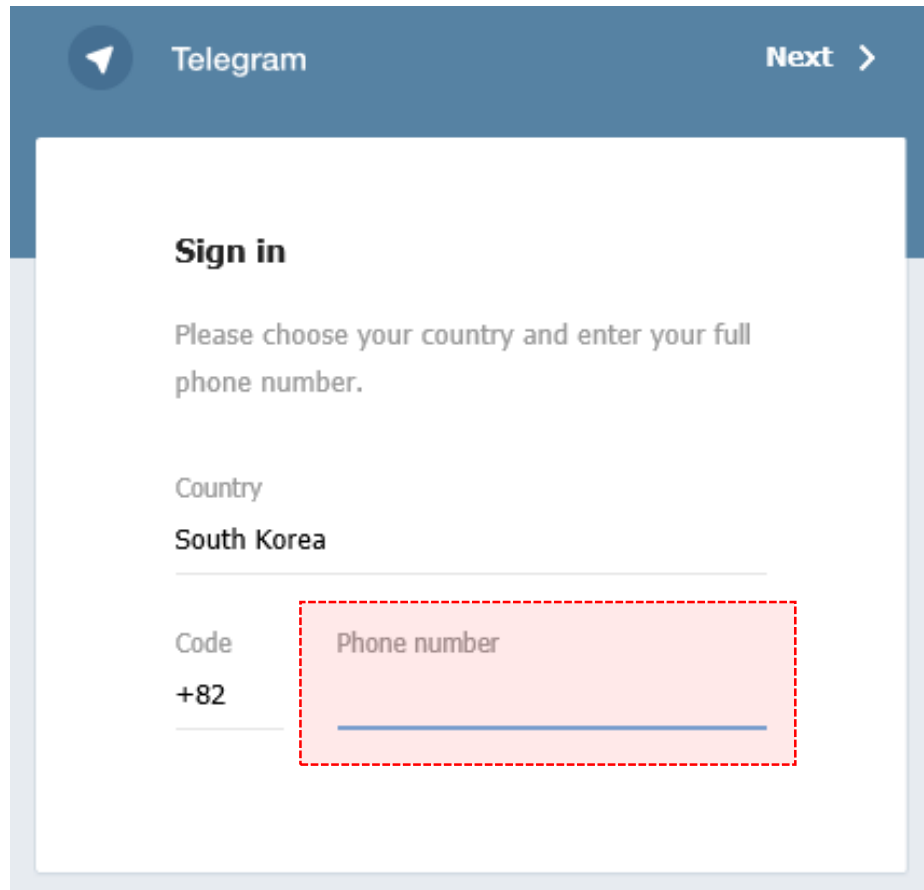
친구수 친구수가 0명 증가했습니다.
홈 사용자가 홈을 0번 조회했습니다.

<https://i.kakao.com/login>으로 접속하여 카카오톡 채널 챗봇 신청
"카카오톡 채널 챗봇 만들기"를 클릭하고 신청 화면에서 "개인" 선택 후 신청



출처 : <https://i.kakao.com/login>

<https://web.telegram.org/#/login> 에 접속하여 휴대폰 인증 및 텔레그램 가입



Telegram

Next >

Sign in

Please choose your country and enter your full phone number.

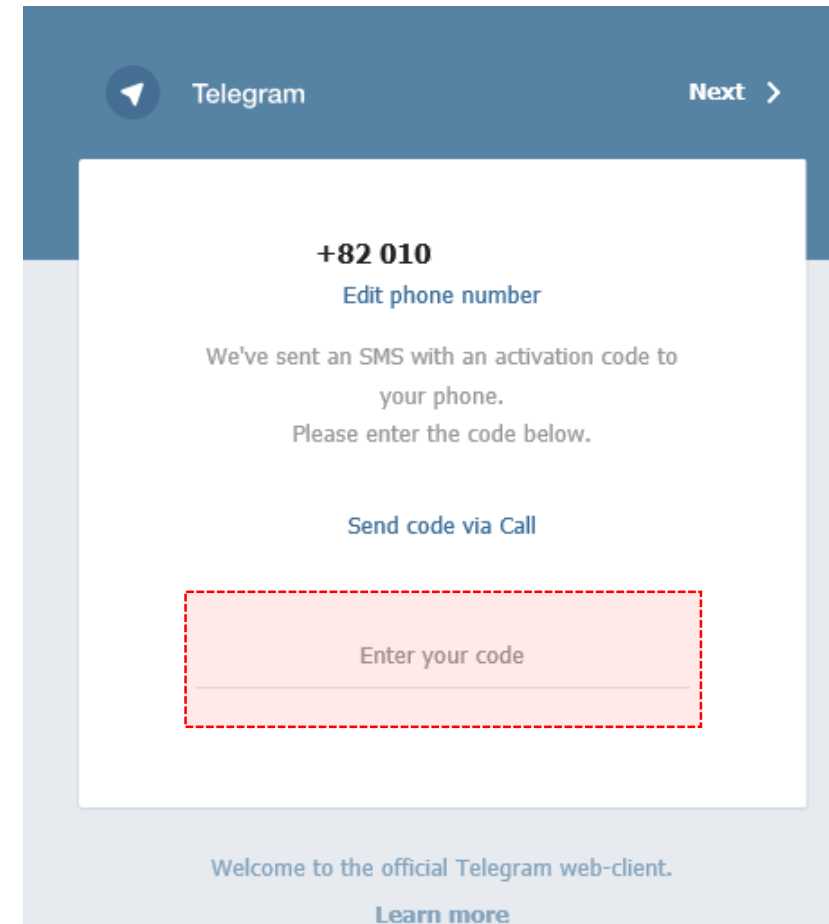
Country

South Korea

Code

+82

Phone number



Telegram

Next >

+82 010

Edit phone number

We've sent an SMS with an activation code to your phone.
Please enter the code below.

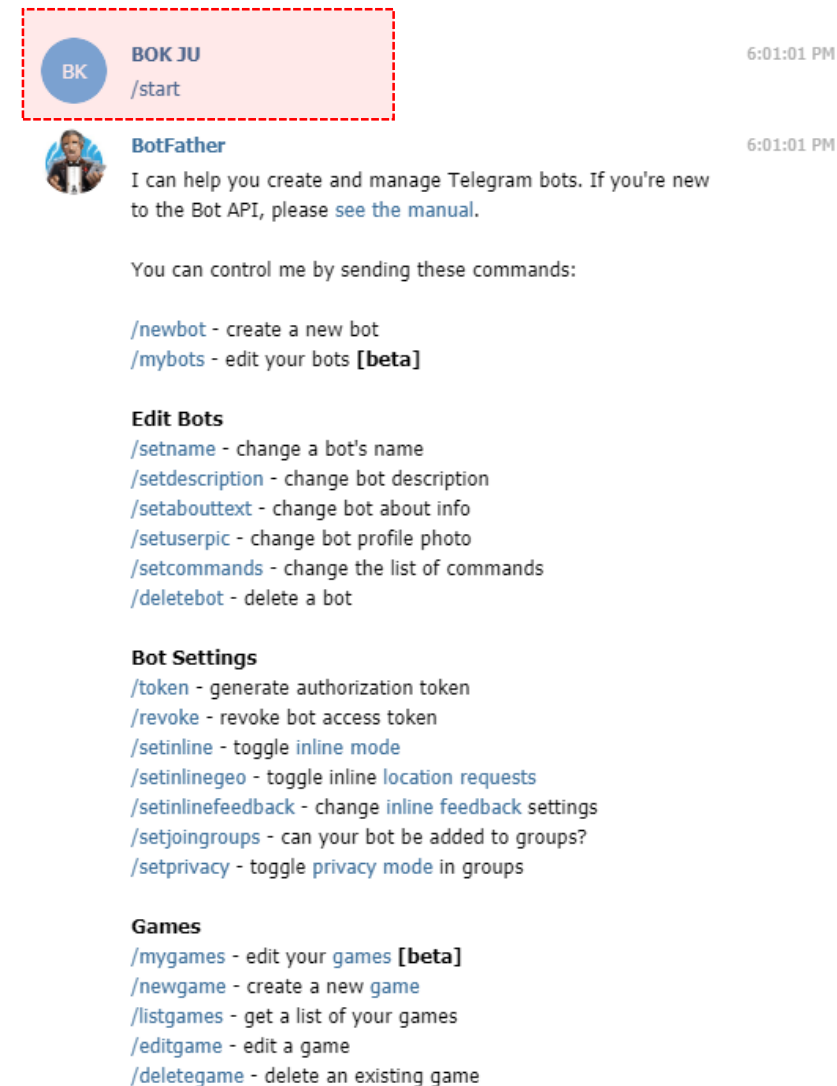
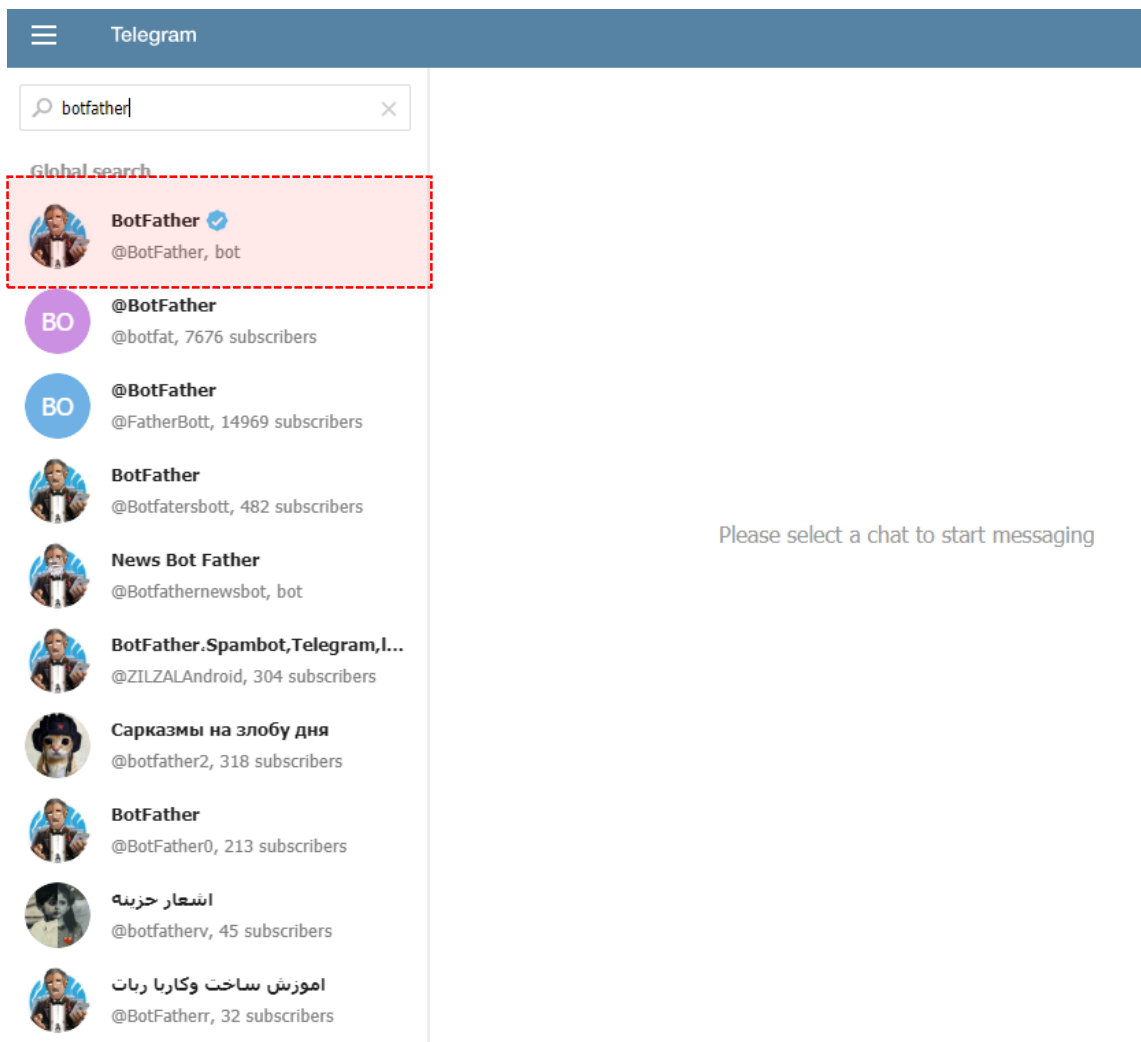
Send code via Call

Enter your code


Welcome to the official Telegram web-client.

[Learn more](#)


가입 완료 후 검색 창에서 "botfather" 검색 및 BotFather 클릭 후
오른쪽 대화 창에서 "/start" 입력




대화 창에서 "/newbot"을 입력
생성할 챗봇 클라이언트명을 입력하면 인증키 발급

 **BOK JU**
/newbot


6:01:17 PM

 **BotFather**
Alright, a new bot. How are we going to call it? Please choose a name for your bot.


6:01:17 PM

 **BOK JU**
secu chatbot


6:01:32 PM

 **BotFather**
Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.

6:01:32 PM

 **BOK JU**
new_secu_bot

6:02:24 PM

 **BotFather**
Done! Congratulations on your new bot. You will find it at t.me/new_secu_bot. You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

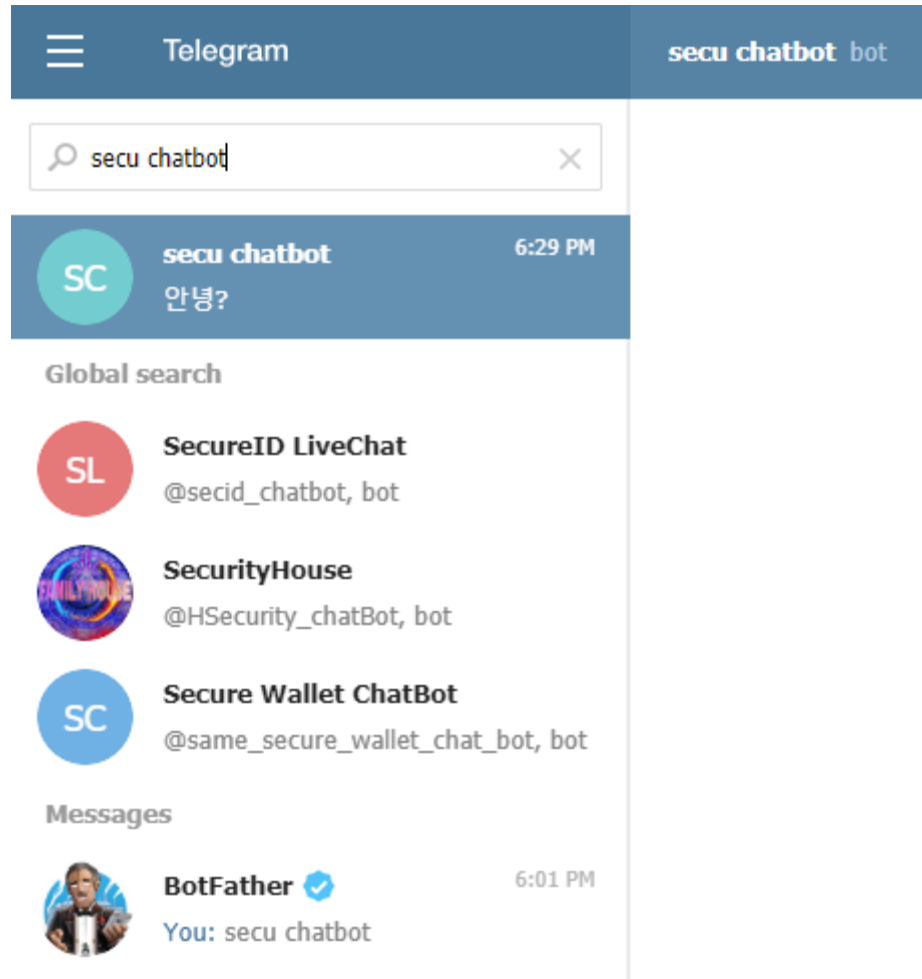
6:02:24 PM

Use this token to access the HTTP API:
`1267957310:AAFtibEGqYkBTx_6Ph7Q7bak`

Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:
<https://core.telegram.org/bots/api>

생성된 챗봇 명으로 검색 후 대화 입력



챗봇 클라이언트(텔레그램) 에 메시지를 전송

인증키 토큰

```
import requests
import os
import json
```

```
token = '1267957310:AAFtibEGqYkBTx_6Ph7Q7bak[redacted]'
```

```
url = 'https://api.telegram.org/bot{}/getUpdates'.format(token)
response = json.loads(requests.get(url).text)
```

```
response
```

```
{'ok': True,
 'result': [{'update_id': 934596913,
  'message': {'message_id': 3,
   'from': {'id': 1239692491,
    'is_bot': False,
    'first_name': 'BOK JU',
    'last_name': 'KIM',
    'language_code': 'ko'},
   'chat': {'id': 1239692491,
    'first_name': 'BOK JU',
    'last_name': 'KIM',
    'type': 'private'},
   'date': 1590830935,
   'text': '안녕?'}]}}
```

```
url = 'https://api.telegram.org/bot{}/sendMessage'.format(token)
```

```
chat_id = response["result"][-1]["message"]["from"]["id"]
msg = '나는 챗봇이야'
```

```
requests.get(url, params = {"chat_id" : chat_id, "text" : msg})
```

```
<Response [200]>
```

```
import requests
import os
import json
```

```
token = '1267957310:AAFtibEGqYkBTx_6Ph7Q7bak2'
```

```
url = 'https://api.telegram.org/bot{}/getUpdates'.format(token)
response = json.loads(requests.get(url).text)
```

```
response
```

bot_telegram.ipynb 실습

```
{
  'last_name': 'KIM',
  'language_code': 'ko',
  'chat': {
    'id': 1239692491,
    'first_name': 'BOK JU',
    'last_name': 'KIM',
    'type': 'private'
  },
  'date': 1590830935,
  'text': '안녕?'}]}
```

```
url = 'https://api.telegram.org/bot{}/sendMessage'.format(token)
```

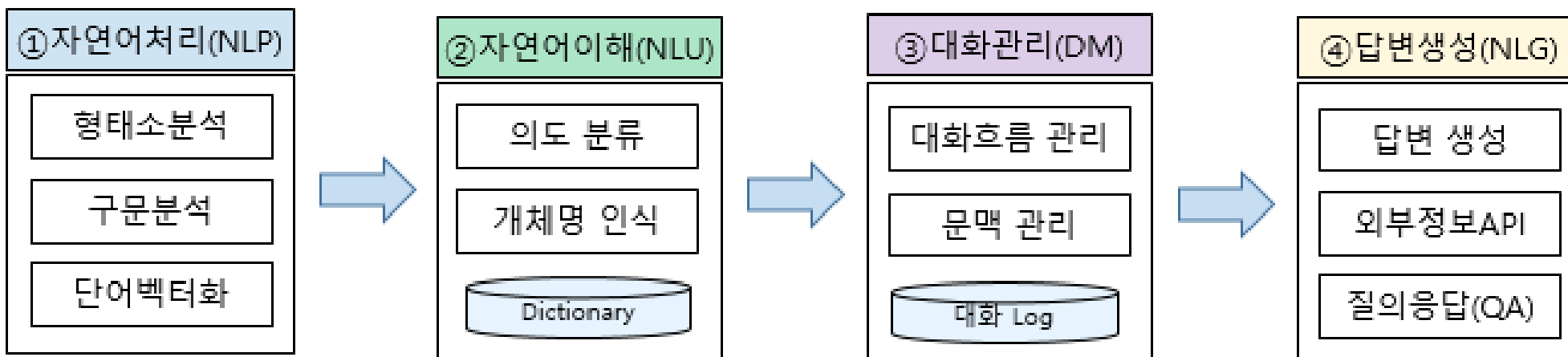
```
chat_id = response["result"][-1]["message"]["from"]["id"]
msg = '나는 챗봇이야'
```

```
requests.get(url, params = {"chat_id" : chat_id, "text" : msg})
```

```
<Response [200]>
```

▪ 챗봇 엔진의 구성은

①자연어 처리(NLP) ②자연어 이해(NLU) ③대화관리(Dialog Management) ④ 자연어(답변) 생성(NLG) 등으로 구성
본격적인 자연어 처리에 앞서 텍스트 전처리 과정 선행



- 텍스트 전처리(preprocessing)란?

정제, 불용어 제거, 어간 추출, 토큰화 및 문서 표현 등을 진행하는 과정

참고) 파이썬 pytorch의 torchtext는 텍스트 전처리를 위한 파이썬 라이브러리

- 정제(Cleaning)

특수문자 등과 같은 불필요한 노이즈 텍스트 제거 및 대소문자 통일

특수문자 : '!"#\$%&₩'()*+,-./:;<=>?@[₩₩]^_`{|}~'

대소문자 통일 : korea, Korea → KOREA

- 불용어 제거 (Stop word elimination)

전치사, 관사 등 문장이나 문서의 특징을 표현하는데 불필요한 단어를 제거하는 단계

- (3) 어간 추출 (Stemming)

단어의 기본 형태를 추출하는 단계

stem, stemming, stems, stemmed, stemmer → stem

- 어간 추출 (Stemming)

단어의 기본 형태를 추출하는 단계

stem, stemming, stems, stemmed, stemmer → stem

- 토큰화 (Tokenization)

코퍼스(corpus)에서 분리자(Separator)를 포함하지 않는 연속적인 문자열 단위로 분리

한글 토큰화 결과입니다. → ['한글', '토큰', '화', '결과', '입니다', '.']

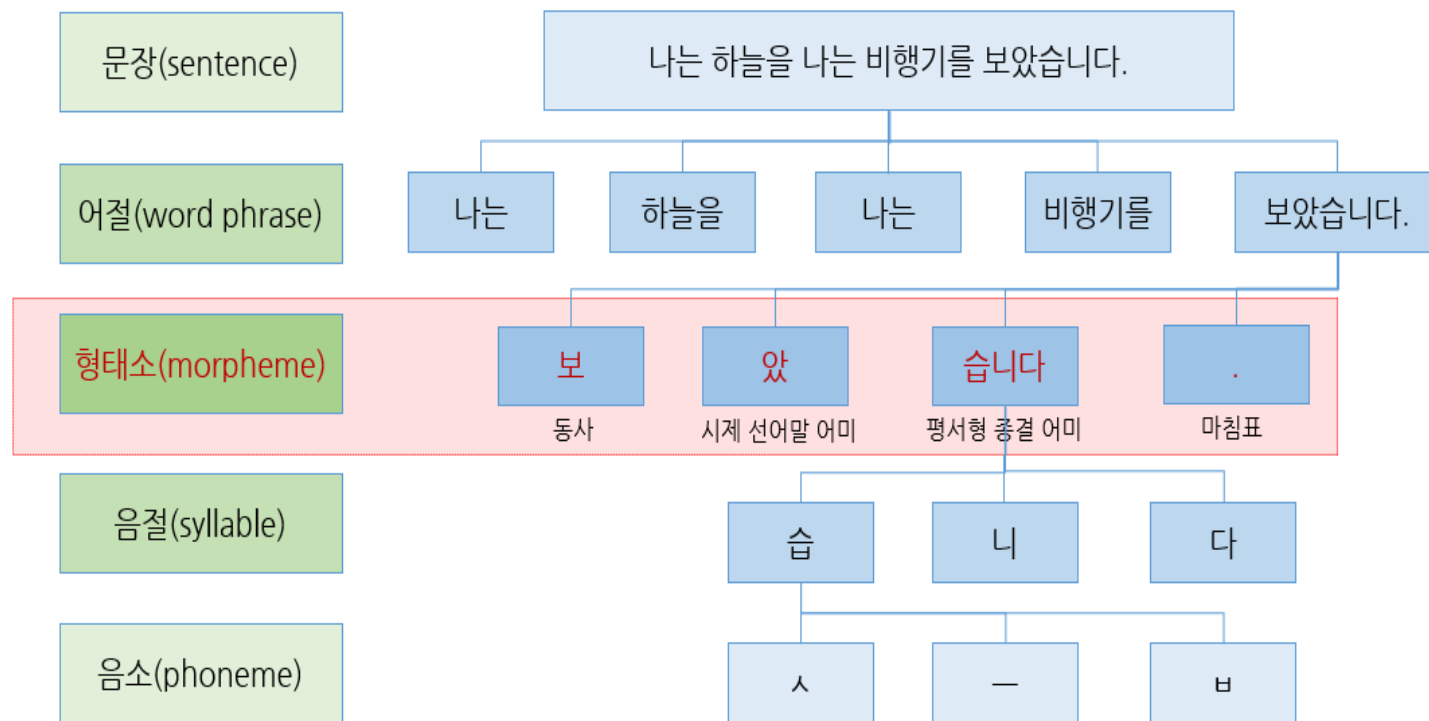
참고) 파이썬에서 영문 토큰화는 nltk를 사용하고 한글 토큰화는 konlpy를 사용

- 형태소란?

자연어에서 최소의 의미 단위

- 형태소 분석(Morphological Analysis) 이란

형태소 분석은 자연어 문장에서 의미를 가진 최소 단위인 형태소(명사, 동사, 형용사, 부사, 조사, 어미 등)를 분석
자연어 처리를 위해 수행해야 하는 가장 첫 단계의 분석



- 형태소 분석기를 이용하여 형태소를 분석

영문 형태소 분석기 : nltk(<http://www.nltk.org>)

한글 형태소 분석기 : konlpy(<http://konlpy.org/ko/latest>)

형태소 분석기

```
from eunjeon import Mecab  
tagger = Mecab()
```

형태소 분석 문장

```
pos = tagger.pos('나는 하늘을 나는 비행기를 보았다')
```

형태소 분석
결과

pos

```
[('나', 'NP'),  
 ('는', 'JX'),  
 ('하늘', 'NNG'),  
 ('을', 'JKO'),  
 ('나', 'NP'),  
 ('는', 'JX'),  
 ('비행기', 'NNG'),  
 ('를', 'JKO'),  
 ('보', 'VV'),  
 ('았', 'EP'),  
 ('다', 'EC')]
```


문장의 구조적 성질을 규칙화한 문법을 통해 문장의 구조를 분석

나	는	하늘	을	나	는	비행기	를	보	았	다
NP	JX	NNG	JKO	VV	ETM	NNG	JKO	VV	EP	EC
주어		목적어		술어		목적어		술어		

- 임베딩(Embedding)이란?

범주형 자료를 연속형 벡터 형태로 변환시키는 것

- 워드 임베딩(Word Embedding)이란?

인간의 언어를 컴퓨터가 이해할 수 있는 언어로 변환

인간이 이해하고 사용하는 언어(문자열)를 컴퓨터로 하여금 효과적으로 인식할 수 있도록 하기 위해 숫자 형태로 변환하는 방법

문자열을 숫자로 변환하여 벡터(Vector) 공간에 표현

- 워드 임베딩(Word Embedding)의 목적?

인간의 언어를 컴퓨터가 이해할 수 있는 언어로 변환하여 벡터 공간에 표현함으로써

단어와 단어, 문장(문서)과 문장(문서) 간의 유사도 계산 가능

벡터간 연산을 통해 의미적 관계 도출 가능

사전에 대량데이터로 학습한 모델(pre-trained model)을 재사용하는 전이학습(Transfer Learning) 가능

워드 임베딩에는 빈도 기반, 토픽 기반, 예측 기반 워드 임베딩 방법 등

- 빈도 기반 워드 임베딩

다수 문서에 등장하는 각 단어들의 빈도를 행렬로 표현하거나 가중치를 부여하여 단어의 중요도나 문서간 유사도를 측정하기 위한 임베딩

(DTM, TF-IDF)

- 토픽 기반 워드 임베딩

주어진 문서에 잠재된 주제(latent topic)를 추론(inference)하기 위한 임베딩

(LDA, Latent Dirichlet Allocation)

- 예측 기반 워드 임베딩

주어진 문장이나 단어의 다음 단어 예측, 주변 단어에 대한 예측, Masking 된 단어의 예측등을 위한 임베딩

(Word2Vec, FastText, BERT, ELMo, GPT)

워드 임베딩은 단어 단위 워드 임베딩에서 문장 단위 워드 임베딩으로 발전

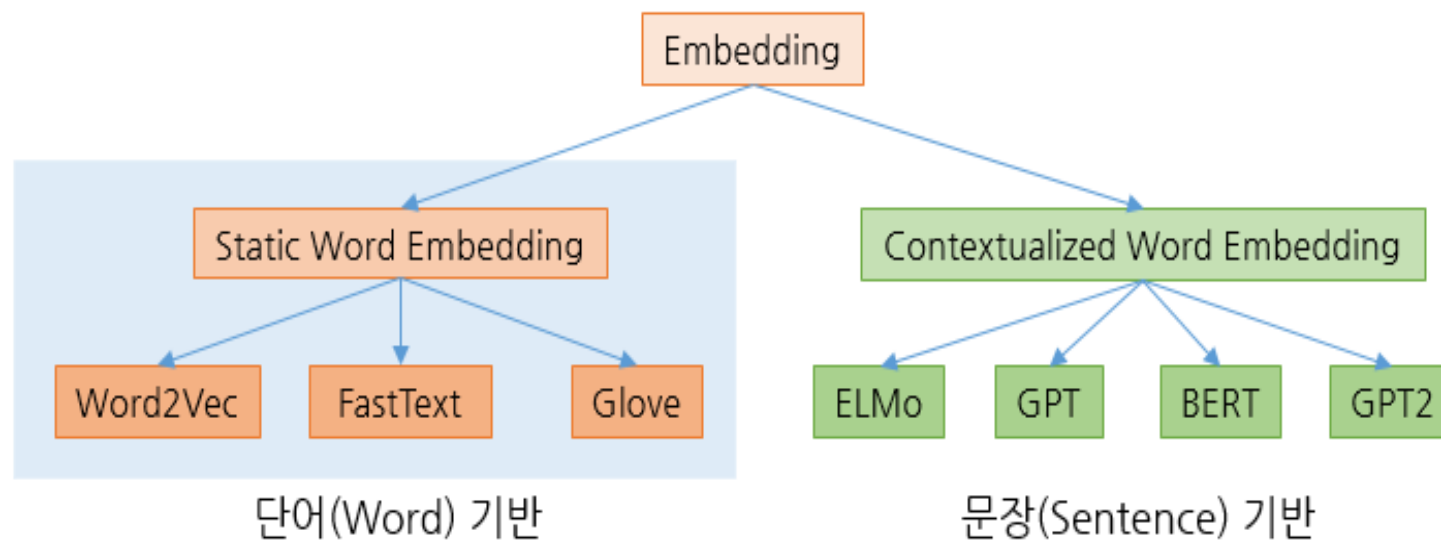
- 단어 단위 워드 임베딩

단어(Word) 기반으로 임베딩 수행

문맥을 고려하지 않은 상태에서 워드 임베딩을 수행

Word2Vec, FastText, Glove 등의 임베딩 방법

서로 다른 문맥의 동음이의어가 동일하게 임베딩되는 문제점

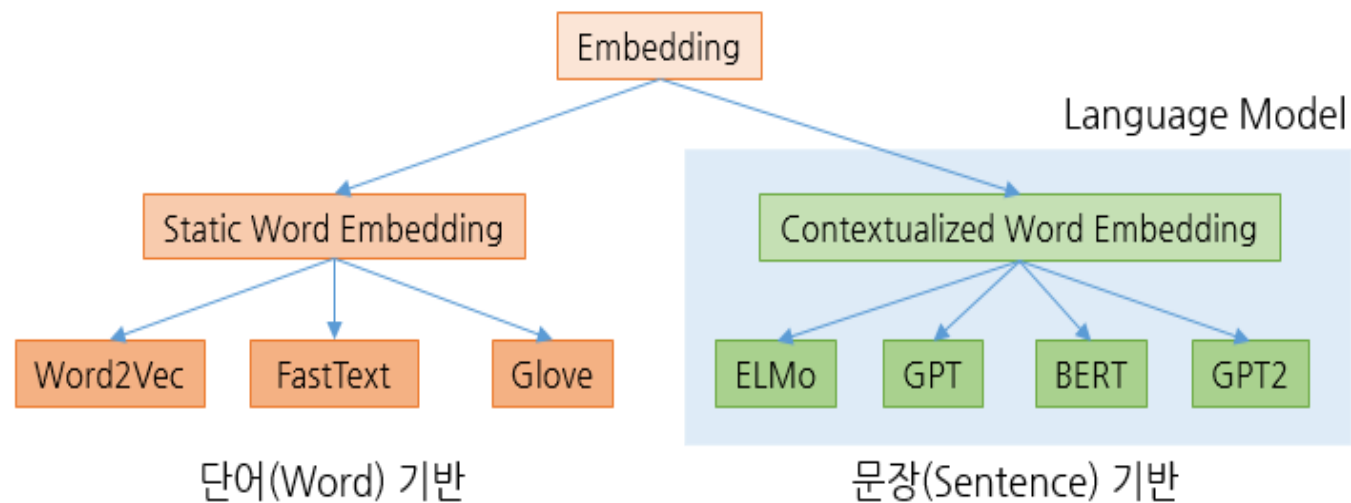


- 문장 단위 워드 임베딩

문맥을 고려하여 문장(Sentence) 기반으로 임베딩을 수행

ELMo, GPT, BERT, GTP2 등의 임베딩 방법

문장(Sentence) 기반으로 임베딩을 수행하여 언어 모델(Language Model)로도 불리움



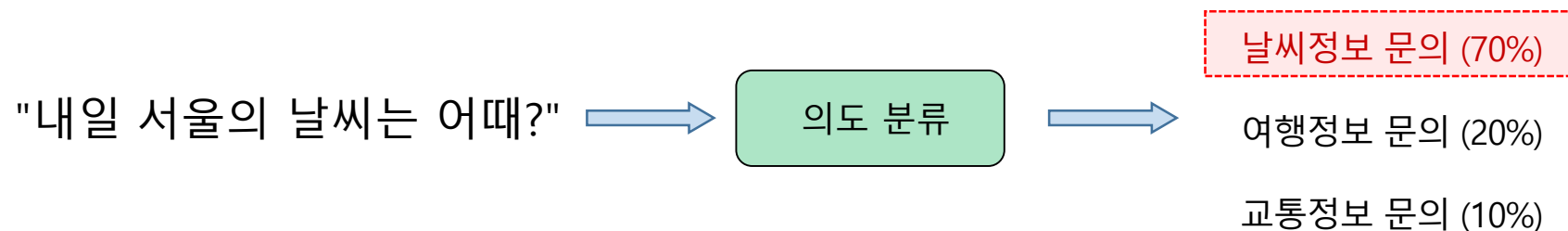
- 인텐트(intent)란?

대화에서 발화자의 의도

- 인텐트(intent) 분류

대화에서 발화자의 의도를 파악하기 위한 분석 과정

Word CNN과 Bi-directional LSTM을 이용한 의도 분류 방법이 많이 사용



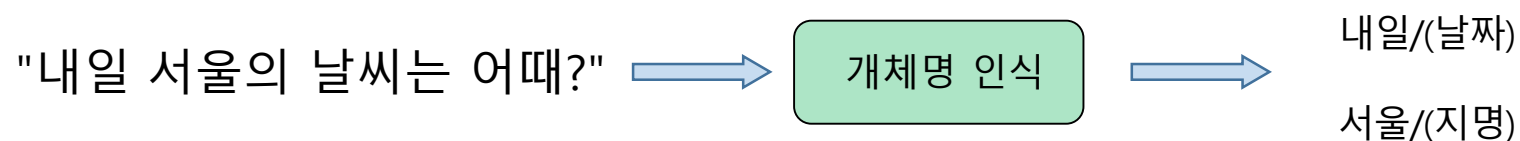
- 개체명 인식(Named Entity Recognition)

사람, 회사, 장소, 시간, 단위 등에 해당하는 단어(개체명)를 문서에서 인식하여 추출 분류

추출된 개체명은 인명(person), 지명(location), 기관명(organization), 시간(time) 등으로 분류

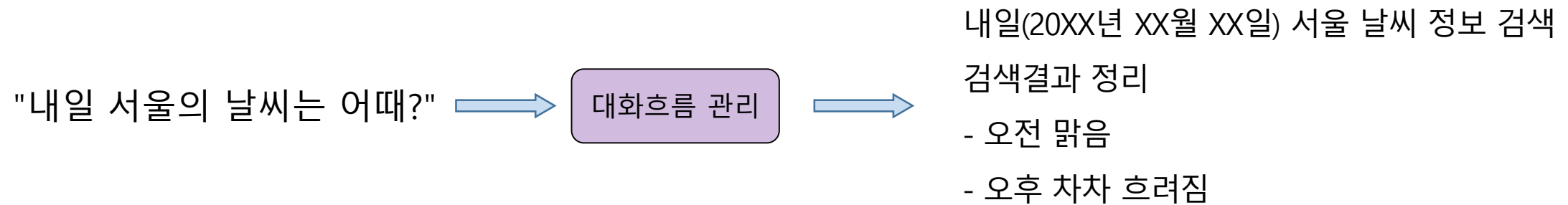
개체명 인식은 사전 기반으로 구현하기도 하지만

최근에는 Bi-directional LSTM과 CRF를 이용한 개체명 인식 방법이 많이 사용



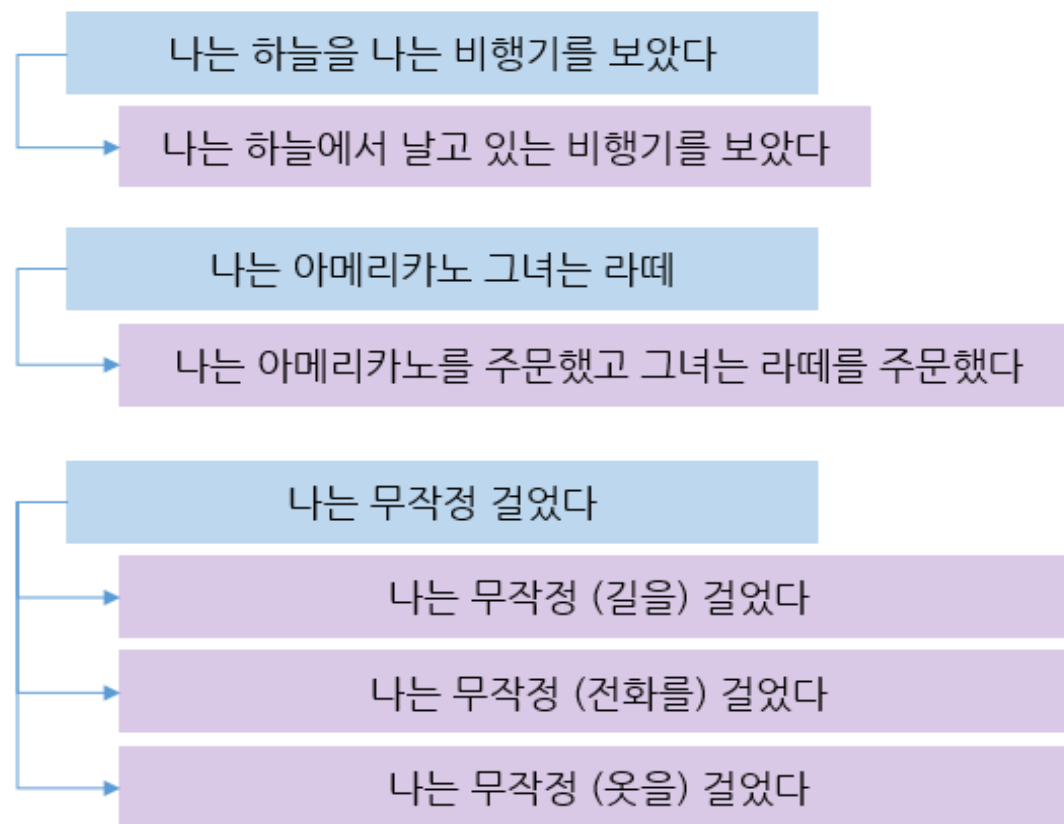
- 대화흐름 관리(Dialog Flow Management)

대화 흐름과 상황을 고려해서 발화자의 의도에 대한 최선의 대화 전략 결정



- 문맥 관리(Context Management)

대화의 맥락(context)과 의도(intent)를 파악하고 문맥과 의도에 맞게 문장의 의미를 분석



- 자연어(답변)생성을 위한 QA(Question Answer)

대화의 맥락(context)과 의도(intent)를 파악하고 문맥과 의도에 맞게 질문에 대한 답변을 생성

QA 알고리즘에 따라 어텐션(Attention)을 적용한 QA, Memory Network를 적용한 QA 등의 방법이 존재

챗봇 엔진 개발을 QA의 종류로 지식 기반(Knowledge Base) QA, 정보검색 기반(Information Retrieval) QA

기계독해 기반(Machine Read Comprehension) QA, 대화학습 기반(Dialog Learning) QA 등이 존재

