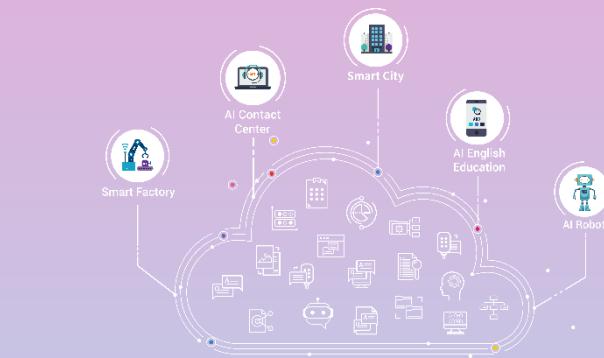


# Level 2. How to customize AI engines



<http://mindslab.ai>

## 1. HMD란

## 2. HMD Syntax

### 3. HMD 모델 생성

## 4. HMD 분류 수행



## HMD

- 패턴 매칭 분류 엔진
- 카테고리 와 탐지식(HMD Expression, Rule) 한 쌍이 하나의 HMD 데이터 (탐지식은 HMD Syntax에 따름)
- HMD 모델은 HMD 데이터의 집합으로 구성
- 사용자가 모델을 만들고 매칭되는 문장의 분류 결과를 얻을 수 있음
- Python으로 개발된 버전[brain-hmd]과 C++로 개발된 버전[brain-full-hmd]이 존재

1. HMD 란

2. HMD Syntax

3. HMD 모델 생성

4. HMD 분류 수행



### HMD Syntax

#### - HMD 탐지식 문법

규칙	설명
(A단어)(B단어)	A단어와 B단어 포함
(A단어 B단어)	A단어나 B단어 포함
(A단어)(@B단어)	A단어 뒤 B단어가 위치
(!A단어)	A단어 미포함
(A단어)(+NB단어)	A단어 위치부터 N번째 띄어쓰기 뒤에 B단어 위치 (단, $0 < N < 10$ )
(A단어)(-NB단어)	A단어 위치부터 N번째 띄어쓰기 앞에 B단어 위치 (단, $0 < N < 10$ )
(%A문자열)	A문자열 포함됩니다.

# 1. HMD 란

## 2. HMD Syntax

### 3. HMD 모델 생성

## 4. HMD 분류 수행



### HMD 모델 생성

- HMD 모델 데이터 구조

```
message HmdRule {  
    string rule = 1; // 탐지식  
    repeated string categories = 2; // 카테고리값  
}  
  
message HmdModel {  
    maum.common.LangCode lang = 1; // 언어  
    string model = 2; // HMD 모델명  
    repeated HmdRule rules = 3; // HMD 데이터 집합  
}
```

### HMD 모델 생성

- HMD 모델 생성 rpc

```
service HmdClassifier {  
  
    rpc SetModel (HmdModel) returns (google.protobuf.Empty); // python 버전 hmd 모델 생성  
  
    rpc SetMatrix (HmdModel) returns (google.protobuf.Empty); // c++ 버전 hmd 모델 생성  
  
}
```



# 1. HMD 란

## 2. HMD Syntax

### 3. HMD 모델 생성

## 4. HMD 분류 수행



### HMD 분류 수행

- HMD 분류 수행 요청 데이터 구조
- HMD 분류 수행 결과 데이터 구조

```
message HmdInputText {  
    string model = 1; // 분류 수행할 HMD 모델명  
    maum.common.LangCode lang = 2; // 언어  
    string text = 3; // 분류할 문장  
}
```

```
message HmdClassified {  
    int32 sent_seq = 1; // 문장 시퀀스  
    string category = 2; // 분류 카테고리  
    string pattern = 3; // 분류 기준이된 매트릭스화 HMD Syntax 결과  
    string search_key = 4; // not used  
    string sentence = 5; // 문장 내용  
}
```

```
message HmdResult {  
    repeated HmdClassified cls = 1; // 결과 집합  
}
```

### HMD 분류 수행

- HMD 분류 rpc
- 분류할 문장을 NLP형태소 분석을 통해 재구성하여 분류 수행  
(문장내 품사가 형용사, 동사인 단어에 대해서는 원형 추가해 재구성)

```
service HmdClassifier {  
    // python, C++ 공통 hmd 분류 수행  
    rpc GetClassByText (HmdInputText) returns (HmdResultDocument);  
}
```

## HMD 분류 수행 예시

### - 문장

이번달 사용한 인터넷 전화 요금이 얼마 청구되는지 메일로 발송해주세요.

### - HMD 모델 데이터

카테고리 : 요금조회

탐지식 : (인터넷)(@%청구)

### - 형태소 분석 후 재구성된 문장

이번달 사용하 ㄴ 인터넷 전화 요금 이 얼마 청구되 는지 메일 로 발송하 어 주 세요

### - 분류 결과

```
"cls": [{  
  "category": "요금조회",  
  "pattern": "인터넷$@%청구",  
  "sentSeq": 0,  
  "sentence": "이번달 사용한 인터넷 전화 요금이 얼마 청구되는지 메일로 발송해주세요."  
}]
```

### HMD 실습 방법

#### 1. Brain-ta가 설치된 서버에서 HMD 엔진 실행

```
$MAUM_ROOT/bin/brain-hmdd -> python hmd  
$MAUM_ROOT/bin/brain-full-hmdd -> c++ hmd
```

#### 2. 샘플 클라이언트 코드에서 아래 python, c++ 중 포트하나를 선택해 모델 생성, 분류 테스트를 진행

```
$MAUM_ROOT/samples/test_hmd.py  
# brain-hmdd(python) port  
remote = 'localhost:' + conf.get('brain-  
ta.hmd.front.port')  
# brain-full-hmdd(c++) port  
remote = 'localhost:' + conf.get('brain-ta.hmd.cpp  
.port')
```

감사합니다.

MINDs Lab. AI Platform Company