

## 캡스톤디자인 II 중간보고서(표지)

프로젝트명 : 자연어 처리 기반 영화 리뷰 필터링 및 시각화  
**캡스톤 디자인 II , 중간보고서**

Version 1.0

개발 팀원 명(팀리더): 한재혁  
서민석  
정인상  
대표 연락처: 010-5584-9805  
e-mail: hjh0633@gmail.com

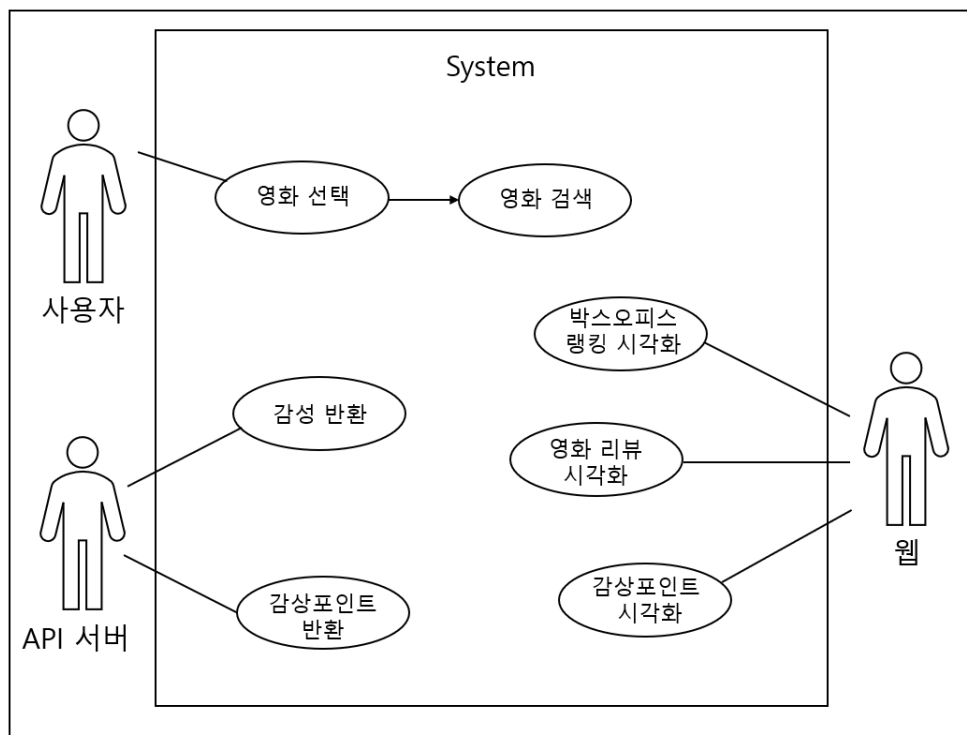
## 캡스톤 디자인 II 중간보고서 내용

1. 요구사항 정의서에 명시된 기능에 대하여 현재까지 진척된 결과 및 그 내용을 기술하시오.

### - 시스템(하드웨어, 시스템)구성도, 또는 소프트웨어 아키텍처

- 웹 : 사용자의 접근을 위한 웹을 제공한다. 해당 웹을 통해 사용자는 박스 오피스 랭킹의 상위 5개 영화 정보를 시각화하고, 사용자가 원하는 영화의 검색할 수 있는 UI를 제공하며 결과를 출력한다. 검색 결과에서 영화를 선택하면 해당 영화에 대한 상세한 정보를 열람할 수 있으며, 해당 영화의 리뷰들을 웹에서 크롤링한 후 필터링 및 분석하여 결과를 보여준다.
- API서버 : 사용자가 영화를 선택하게 되면 해당 영화의 영화명이 서버로 전달되면서 해당 영화의 리뷰들을 크롤링하는 API함수가 작성되어 있다. 추가적으로 크롤링된 리뷰들을 AI모델에 입력값으로 넣어 결과값을 반환하는 API함수 또한 작성되어 있다.
- 감상 포인트 추출 모델 개발: 영화 리뷰에서 사용자의 평가가 들어간 부분을 추출해 해당 평가가 긍정적인지 부정적인지 판단한다. 이를 위해 모두의 말뭉치에서 제공하는 속성 기반 감성 분석 데이터셋과 직접 구축한 데이터셋을 사용했으며, PyABSA 라이브러리를 활용하여 한국어 속성 기반 감성 분석 모델을 구현하였다. 모델은 입력받은 리뷰를 형태소 단위로 분리한 뒤, aspect term extraction 및 aspect polarity classification을 수행한다.  
실제 사용 환경에서는 은어, 비속어, 이모티콘 등이 포함된 다수의 리뷰가 존재할 것으로 예상되어 별도의 클렌징 작업을 하지 않은 데이터셋을 사용하였다.

### - 기능별 상세 요구사항(또는 유스케이스)



### - 소스 코드

모델 생성 및 훈련 코드

```
import os
import findfile
from pyabsa.functional import ATEPCModelList
```

```

from pyabsa.functional import Trainer, ATEPCTrainer
from pyabsa.functional import ABSADatasetList
from pyabsa.functional import ATEPCConfigManager
from pyabsa.functional.dataset import DatasetItem

atepc_config = ATEPCConfigManager.get_atepc_config_multilingual()
atepc_config.pretrained_bert = 'kykim/bert-kor-base'
atepc_config.model = ATEPCModelList.LCF_ATEPC
dataset_path = '/home/seominseok/ABSADatasets/datasets/atepc_datasets/100.CustomDataset'
aspect_extractor = ATEPCTrainer(config=atepc_config,
                                dataset=dataset_path,
                                from_checkpoint="", # set checkpoint to train on the checkpoint.
                                checkpoint_save_mode=1,
                                auto_device=True
                                ).load_trained_model()

```

## 모델 테스트 결과

The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 이런 <진개:Negative Confidence:0.9988879561424255> 는 누구 나 예상 할 수 있 는 전형 적 인 스토리 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 이런 식 이 라 앞뒤 <사정:Negative Confidence:0.9987070560455322> 설명 이나 사연 팔 이 , 인과관계 이런 게 넘 나 부족 하 다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 전설 의 <포켓몬 들:Negative Confidence:0.6877705454826355> 의 존재감 이 너무 열 어 지 지 않 았 나 싶 기 도 하 지만 <아기:Negative Confidence:0.5130007266998291> 가 너무 열길 로 새 지 않 고 일직선 으로 달린 것 은 괜 찮 았 다고 본다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 솔직히 이 영화 를 , 시나리오 나 스토리 전개 <면:Negative Confidence:0.9570225477218628> 에서 뛰어나 다거나 새로운 면 이 돋보인다 ( 참의 적 이 다 ) 고 보 기 에 는 힘들 다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 해당 함 의 근본 적 인 원인 은 <인물:Negative Confidence:0.9978147745132446> 들 에게 있 다고 본다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 상황 이 급진 하 다 보 니 , <에피소드:Negative Confidence:0.9975761771202087> 들 이 이어지 는 과정 역시 매끄럽 지 못하 다는 생각 이 들 었 다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 조선 시대 를 디운 영화 들 중 에서 는 꽤 나 신선 했 고 , 이 로 하여금 ` 조선 <관 액션 히어로 물:Positive Confidence:0.9996980428695679> ` 이 라는 새로운 <장르:Positive Confidence:0.9996249675750732> 가 탄생 한 것 같 아 기뻐 다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: < 퍼스트 <맨:Positive Confidence:0.9998030066490173> > 은 첫 <시퀀스:Positive Confidence:0.999869704246521> 부터 강렬 하 다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 바로 눈 앞 에 떨어지 는 폭탄 , 충격 으로 무분별 하 게 죽 어 나 가 는 병사 들 , 하늘 에서 내려다보 고 , 땅 , 바다 위 에서 올려다보 는 상황 의 긴박감 , 배 안 에서 생존 을 위해 고군분투 하 는 군인 들 의 생생 한 묘사 는 <놀란 감독:Positive Confidence:0.9948152899742126> 의 연출력 을 재 입증 하 는 요소 다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 적재적소 에 놓인 의미 가득 한 <소품:Positive Confidence:0.9997608065605164> 들 과 긴장감 을 배가 시키 는 <사운드 활용:Positive Confidence:0.9997732043266296> 은 높이 평가 할 만 하 다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: OST 와 이태리 <풍경:Positive Confidence:0.9999150037765503> 이 갑  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 할 말 을 잃 게 만드 는 명연기 였 던 가 같 아요 ~ ! !  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 이 영화 는 <풍경:Positive Confidence:0.9998577833175659> 이 무척 아 립답다 .  
 The results of aspect term extraction have been saved in /home/seominseok/PyABSA/atepc\_inference.result.json  
 Example 0: 완벽 하 게 깔끔 한 기 - 승 - 전 - 결의 스토리 와 최고 의 질료 한 연출 과 많 은 볼거리를 제공 하 는 영상미 .

## 형태소 분리 코드

```

# -*- coding: utf-8 -*-
# file: atepc_utils.py
# time: 2021/5/27 0027
# author: yangheng <hy345@exeter.ac.uk>
# github: https://github.com/yangheng95
# Copyright (C) 2021. All Rights Reserved.
# from transformers import AutoTokenizer
import re
import string
from konlpy.tag import Mecab
from pyabsa.core.apc.dataset_utils.apc_utils import (get_syntax_distance,
                                                    get_cdw_vec,
                                                    get_lca_ids_and_cdm_vec)
korean_re = re.compile(r "[ㄱ-ㅣ가-힣]")
def isKorean(text):
    return korean_re.search(text) is not None

```

```

mecab = Mecab()
def split_text (text ):
    # text = ' '.join(tokenizer.tokenize(text)[1:])
    # return text
    text = text .strip() # Remove white space
    Chinese_punctuation = '# $ % & ' ( ) * + , - / : ; < = > @ [ \ ] ^ _ ` { | } ~ ( ) 「 」 、 、 “ ”
    punctuation = string .punctuation + Chinese_punctuation # List of special characters
    defined
    for p in punctuation :
        text = text .replace('{}'.format (p ), ' {}'.format (p )) # Add spaces before and
    after special characters
    # text =
    '.join(re.compile(r'\w+[{]}'.format(re.escape(punctuation))).findall(text)).replace('$
    $', '$T$')
    # for non-latin Languages
    non_latin_unicode = [
        '\u4e00 -\u9fa5 ', # Chinese
        '\u0800 -\u4e00 ', # Japanese
        '\u0e00 -\u0e7f ', # Thai
        '\u1000 -\u109f ', # Myanmar
    ]
    # latin_lan = ([re.match(lan, text) for lan in non_latin_unicode])
    latin_lan = ([re .findall ('[{}]' .format (lan ), text ) for lan in non_latin_unicode ])
    if isKorean (text ):
        return mecab .morphs(text )
    # Modification: Split and return even if Korean exists
    if not any (latin_lan ):
        return text .split()

s = text
word_list = []
while len (s ) >0 :
    match_ch = re .match ('[{}]' .format (''.join (non_latin_unicode )), s )
    if match_ch : # character split
        word = s [0 :1 ]
    else : # If it is not matched, check if it is in English
        match_en = re .match (r '[a-zA-Z\d ]+', s )
        if match_en : # english
            word = match_en .group (0 ) # If it's English, group it and split it
        else :
            word = s [0 :1 ] # If it's not in English, just split character
    if word :
        word_list .append (word )

    # Remove word, white space from text
    s = s .replace(word , '', 1 ).strip(' ')
return word_list

def process_iob_tags (iob_tags : list ) -> list :
    for i in range (len (iob_tags ) - 1 ):
        if iob_tags [i ] == 'O' and 'ASP' in iob_tags [i + 1 ]:
            iob_tags [i + 1 ] = 'B-ASP'
        if 'ASP' in iob_tags [i ] and 'B-ASP' in iob_tags [i + 1 ]:
            iob_tags [i + 1 ] = 'I-ASP'
    return iob_tags

def prepare_input_for_atpc (opt , tokenizer , text_left , text_right , aspect ):
    if hasattr (opt , 'dynamic_truncate') and opt .dynamic_truncate:
        _max_seq_len = opt .max_seq_len - len (aspect .split())
        text_left = text_left .split(' ')

```

```

text_right = text_right .split(' ')
if _max_seq_len < len (text_left ) + len (text_right ):
    cut_len = len (text_left ) + len (text_right ) - _max_seq_len
    if len (text_left ) > len (text_right ):
        text_left = text_left [cut_len :]
    else :
        text_right = text_right [:len (text_right ) - cut_len ]
text_left = ' '.join (text_left )
text_right = ' '.join (text_right )
bos_token = tokenizer .bos_token if tokenizer .bos_token else '[CLS]'
eos_token = tokenizer .eos_token if tokenizer .eos_token else '[SEP]'
text_raw = text_left + ' ' + aspect + ' ' + text_right
text_spc = bos_token + ' ' + text_raw + ' ' + eos_token + ' ' + aspect + ' ' + eos_token
text_bert_tokens = tokenizer .tokenize(text_spc )
text_raw_bert_tokens = tokenizer .tokenize(bos_token + ' ' + text_raw + ' ' + eos_token
)
aspect_bert_tokens = tokenizer .tokenize(aspect )
text_bert_indices = tokenizer .convert_tokens_to_ids(text_bert_tokens )
text_raw_bert_indices = tokenizer .convert_tokens_to_ids(text_raw_bert_tokens )
aspect_bert_indices = tokenizer .convert_tokens_to_ids(aspect_bert_tokens )
aspect_begin = len (tokenizer .tokenize(bos_token + ' ' + text_left ))
if 'lcf' in opt .model_name or opt .use_syntax_based_SRD:
    syntactical_dist , _ = get_syntax_distance(text_raw , aspect , tokenizer , opt )
else :
    syntactical_dist = None
lcf_cdm_vec = get_lca_ids_and_cdm_vec(opt , text_bert_indices , aspect_bert_indices ,
    aspect_begin , syntactical_dist )
lcf_cdw_vec = get_cdw_vec(opt , text_bert_indices , aspect_bert_indices ,
    aspect_begin , syntactical_dist )
inputs = {
    'text_raw': text_raw ,
    'text_spc': text_spc ,
    'aspect': aspect ,
    'text_bert_indices': text_bert_indices ,
    'text_raw_bert_indices': text_raw_bert_indices ,
    'aspect_bert_indices': aspect_bert_indices ,
    'lcf_cdm_vec': lcf_cdm_vec ,
    'lcf_cdw_vec': lcf_cdw_vec ,
}
return inputs

def load_atepc_inference_datasets (fname ):
    lines = []
    if isinstance (fname , str ):
        fname = [fname ]
    for f in fname :
        print ('loading: {}'.format (f ))
        fin = open (f , 'r', encoding ='utf-8')
        lines .extend (fin .readlines ())
        fin .close ()
    for i in range (len (lines )):
        lines [i ] = lines [i ][:lines [i ].find('!sent!')].replace('[ASP]', '')
    return sorted (set (lines ) , key =lines .index )

```

2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

- 문제해결을 위해 적용한 방법(또는 기법) 결과, (문제점, 해결방안)

<p>[문제점]</p> <p>I. 진행 과정에서 오류 발생</p>
<p>[해결방안]</p> <p>II. 필요한 라이브러리나 패키지 등의 버전 정보를 확인하고 해당 오류의 정확한 원인을 파악</p>

- 팀원의 책임 및 역할 수행에 대한 결과, (문제점, 해결방안)

<p>[문제점]</p> <p>I. 딥러닝 및 자연어 처리 분야에 대한 지식 부족으로 인한 진행 차질</p> <p>II. PyABSA 라이브러리에 대한 이해 미숙</p>
<p>[해결방안]</p> <p>I. 추가적인 개인 공부 및 자연어 처리 전공 교수님과의 상담을 통해 확립된 방향성과 제시받은 아이디어를 토대로 진행</p> <p>II. 해당 라이브러리 설계자에게 문의 및 코드 해석을 통한 동작 방식 이해</p>

프로젝트명 : 자연어 처리 기반 영화 리뷰 필터링 및 시각화  
**소프트웨어 요구사항 정의서**

Version 1.0

개발 팀원 명(팀리더): 한재혁  
서민석  
정인상  
대표 연락처: 010-5584-9805  
e-mail: hjh0633@gmail.com

## 목차

1. 개요
2. 시스템 장비 구성요구사항
3. 기능 요구사항
4. 성능 요구사항
5. 인터페이스 요구사항
6. 데이터 요구사항
7. 테스트 요구사항
8. 보안 요구사항
9. 품질 요구사항
10. 제약 사항
11. 프로젝트 관리 요구사항



## 1. 시스템 개요

### 1) 목표

소비자가 영화를 선택하는 과정에 있어서 도움이 되지 않는 리뷰들(내용이 없는 리뷰, 영화의 내용과 관계없는 리뷰, 지나치게 편향된 리뷰 등)을 필터링하고 이를 관람 포인트별로 분석하여 어느 포인트에서 얼마만큼의 긍정도를 가지고 있는 시각화함으로써 소비자들의 영화 선택에 있어 도움이 되는 웹 어플리케이션을 제작한다.

### 2) 시스템 구성도



## 2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001		
요구사항 명칭		모델 개발 서버	응락수준	필수
요구사항 분류		시스템 장비 구성 요구사항		
요구사항 상세설명	정의	<ul style="list-style-type: none"> <li>- 모델 훈련 및 테스트</li> </ul>		
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목 : Ubuntu Server</li> <li>- 장비 수량 : 1개</li> <li>- 장비 기능 : 모델 학습 및 테스트 기능 수행</li> <li>- 장비 성능 및 특징               <ul style="list-style-type: none"> <li>- CPU : Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz</li> <li>- GPU : NVIDIA GeForce RTX 2080 Ti (3개)</li> <li>- RAM : 187GB</li> </ul> </li> <li>- 자원 제약사항 : 교내망으로만 접속할 수 있기 때문에 서버 사용을 위해선 교내망을 통한 접속이 이루어져야 한다.</li> </ul>		

요구사항 고유번호		ECR-002		
요구사항 명칭		인공지능	응락수준	필수
요구사항 분류		시스템 장비 구성 요구사항		
요구사항 상세설명	정의	- 리뷰 필터링 모델		
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목 : 머신러닝 모델</li> <li>- 장비 수량 : 1개</li> <li>- 장비 기능 : 영화 리뷰 필터링</li> <li>- 장비 성능 및 특징 : 크롤링 해온 리뷰들에 대해 실시간으로 해당 리뷰를 보여줄 것인지 보여주지 않을 것인지 여부를 결정한다. 근거 없는 비판, 칭찬성 리뷰, 비관람객이 작성한 리뷰, 편향성이 짙은 리뷰 등이 필터링 된다.</li> </ul>		

요구사항 고유번호		ECR-003		
요구사항 명칭		인공지능	응락수준	필수
요구사항 분류		시스템 장비 구성 요구사항		
요구사항 상세설명	정의	- 감상포인트 추출 모델		
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목 : 속성 기반 감성 분석 딥러닝 모델</li> <li>- 장비 수량 : 1개</li> <li>- 장비 기능 : 감상 포인트 추출</li> <li>- 장비 성능 및 특징 : 크롤링 해온 리뷰에 대해 각각의 리뷰별로 감상 포인트를 추출, 감성 분석을 진행한다. 감상 포인트별 감성 분석 결과를 종합해 영화별 감상 포인트를 반환한다.</li> </ul>		

요구사항 고유번호		ECR-004		
요구사항 명칭		감상 포인트 추출 모델용 학습 데이터셋	응락수준	필수
요구사항 분류		시스템 장비 구성 요구사항		
요구사항 상세설명	정의	- 인공지능 학습 데이터셋		
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목 : 속성 기반 감성 분석 데이터셋</li> <li>- 장비 수량 : 1개</li> <li>- 장비 기능 : 속성 기반 감성 분석 딥러닝 모델 학습 및 검증</li> <li>- 장비 성능 및 특징 <ul style="list-style-type: none"> <li>- 리뷰 텍스트 내에서 관람객의 의견이 들어간 substring을 from, to로 인덱싱</li> <li>- 인덱싱한 부분의 내용을 고려하여 해당 부분에 대한 감성을 2개(긍정, 부정)로 라벨링</li> </ul> </li> </ul>		

### 3. 기능 요구사항

요구사항 고유번호		SFR-001		
요구사항 명칭		박스 오피스 랭킹 시각화	응락수준	필수
요구사항 분류		기능 요구사항		
요구사항 상세 설명	정의	- 박스 오피스 랭킹 수집 및 시각화		
	세부 내용	<ul style="list-style-type: none"> <li>- 박스 오피스 랭킹 상위 5개 목록 수집</li> <li>- 수집 목록 시각화</li> <li>- 수집 목록 클릭 시 해당 목록에 대한 영화 정보 시각화</li> </ul>		

요구사항 고유번호		SFR-002		
요구사항 명칭		영화 검색	응락수준	필수
요구사항 분류		기능 요구사항		
요구사항 상세 설명	정의	- 사용자가 원하는 영화의 목록을 검색		
	세부 내용	<ul style="list-style-type: none"> <li>- 사용자가 원하는 영화의 목록을 검색</li> <li>- 검색하는 문자열의 길이는 2 이상이어야 함</li> <li>- 검색하는 문자열을 포함하고 있는 영화를 KMDB에서 검색해 결과 시각화</li> <li>- KMDB에서 받은 영화의 포스터, 제목, 감독, 출시년도, 출연배우 표시</li> </ul>		

요구사항 고유번호		SFR-003		
요구사항 명칭		영화 리뷰 시각화	응락수준	필수
요구사항 분류		기능 요구사항		
요구사항 상세 설명	정의	- 사용자가 선택한 영화의 리뷰를 시각화		
	세부 내용	<ul style="list-style-type: none"> <li>- 사용자가 선택한 영화의 리뷰를 시각화</li> <li>- 사용자가 선택한 영화의 제목을 인자로 자체 제작한 API함수 호출</li> <li>- API함수가 반환한 결과를 리뷰 페이지에 출력</li> </ul>		

요구사항 고유번호		SFR-004		
요구사항 명칭		영화 리뷰 분석 시각화	응락수준	필수
요구사항 분류		기능 요구사항		
요구사항 상세 설명	정의	- 사용자가 선택한 영화의 리뷰를 분석하여 결과를 시각화		
	세부 내용	<ul style="list-style-type: none"> <li>- 사용자가 선택한 영화의 리뷰를 분석하여 결과를 시각화</li> <li>- 사용자가 선택한 영화의 제목을 인자로 자체 제작한 API함수 호출</li> <li>- API함수가 반환한 결과를 감상 포인트 페이지에 출력</li> </ul>		

요구사항 고유번호		SFR-005		
요구사항 명칭		감상 포인트 추출 인공지능	응락수준	필수
요구사항 분류		기능 요구사항		
요구사항 상세설명	정의	속성 기반 감성 분석 딥러닝 모델 학습		
	세부 내용	<ul style="list-style-type: none"> <li>- 리뷰 내에서 감상 포인트에 해당하는 부분에 대해 라벨링한 데이터셋으로 모델을 학습한다.</li> <li>- 연계 기능 : 웹 표현</li> <li>- 산출물 : 학습된 딥러닝 모델</li> </ul>		

요구사항 고유번호		SFR-006		
요구사항 명칭		감상 포인트 추출 인공지능	응락수준	필수
요구사항 분류		기능 요구사항		
요구사항 상세설명	정의	속성 기반 감성 분석 딥러닝 모델 학습		
	세부 내용	<ul style="list-style-type: none"> <li>- 리뷰 내에서 감상 포인트에 해당하는 부분에 대해 라벨링한 데이터셋으로 모델을 학습한다.</li> <li>- 연계 기능 : 웹 표현</li> <li>- 산출물 : 학습된 딥러닝 모델</li> </ul>		

#### 4. 성능 요구사항

요구사항 고유번호		PER-001		
요구사항 명칭		리뷰 필터링	응락수준	필수
요구사항 분류		성능 요구사항		
요구사항 상세설명	정의	영화 리뷰 필터링 시간		
	세부 내용	<ul style="list-style-type: none"> <li>- 직접 설계한 크롤링 API를 통해 크롤링시 1000개당 약 11초가 소요될 것으로 예상된다.</li> <li>- 필터링 시간 및 웹 페이지 로드까지 고려했을 때, 필터링 모델은 1000개당 최소 5초 이내로 처리해야 한다.</li> </ul>		

#### 5. 인터페이스 요구사항

요구사항 고유번호		SIR001		
요구사항 명칭		메인 페이지 인터페이스	응락수준	필수
요구사항 분류		인터페이스 요구사항		
요구사항 상세 설명	정의	- 메인 페이지의 인터페이스 요구사항		
	세부 내용	<ul style="list-style-type: none"> <li>- 시스템명이 화면 중앙에 위치</li> <li>- 시스템명 아래 영화를 검색할 수 있는 input박스 위치</li> <li>- 스크롤시 박스 오피스 랭킹 출력</li> </ul>		

요구사항 고유번호		SIR002		
요구사항 명칭		박스 오피스 랭킹 인터페이스	응락수준	필수
요구사항 분류		인터페이스 요구사항		
요구사항 상세 설명	정의	- 메인 페이지의 박스오피스 랭킹 인터페이스 요구사항		
	세부 내용	<ul style="list-style-type: none"> <li>- 메인 페이지에서 스크롤시 출력</li> <li>- 시스템명과 검색 input박스가 화면 상단으로 이동</li> <li>- 화면의 세로 중앙에 가로 방향으로 박스 오피스 랭킹 위치</li> </ul>		

요구사항 고유번호		SIR003		
요구사항 명칭		박스 오피스 상세 인터페이스	응락수준	필수
요구사항 분류		인터페이스 요구사항		
요구사항 상세 설명	정의	- 박스 오피스 랭킹의 영화를 클릭하면 나오는 인터네이스 요구사항		
	세부 내용	<ul style="list-style-type: none"> <li>- 모달 형식으로 선택한 영화의 상세 정보 인터페이스 출력</li> <li>- 영화 포스터, 제목, 감독, 출연 배우, 줄거리 출력</li> <li>- 모달 창 하단에 '해당 영화의 리뷰 바로가기' 버튼 위치</li> </ul>		

요구사항 고유번호		SIR004		
요구사항 명칭		검색 결과 인터페이스	응락수준	필수
요구사항 분류		인터페이스 요구사항		
요구사항 상세 설명	정의	- 영화 검색 결과 페이지의 인터페이스		
	세부 내용	<ul style="list-style-type: none"> <li>- 시스템명과 검색 input 박스가 화면 상단에 위치</li> <li>- 총 검색 결과 수가 input 박스 아래에 위치</li> <li>- 영화 검색 결과가 총 검색 결과 수 아래 list 형식으로 출력</li> <li>- 각 결과는 영화 포스터, 제목, 감독, 출시연도, 출연 배우를 포함하여 출력</li> </ul>		

요구사항 고유번호		SIR005		
요구사항 명칭		영화 상세 인터페이스	응락수준	필수
요구사항 분류		인터페이스 요구사항		
요구사항 상세 설명	정의	- 검색 목록에서 영화 선택시 해당 영화 상세 페이지의 인터페이스		
	세부 내용	<ul style="list-style-type: none"> <li>- 시스템명이 화면 상단에 위치</li> <li>- 총 검색 결과 수가 input 박스 아래에 위치</li> <li>- 영화 포스터, 제목, 감독, 출연 배우, 줄거리 출력</li> <li>- 화면 하단에 해당 영화의 리뷰와 감상 포인트를 볼 수 있는 tab 위치</li> </ul>		

요구사항 고유번호		SIR006		
요구사항 명칭		영화 리뷰 데이터 인터페이스	응락수준	필수
요구사항 분류		인터페이스 요구사항		
요구사항 상세 설명	정의	- 선택한 영화의 필터링된 리뷰 데이터 출력 인터페이스		
	세부 내용	- 해당 리뷰를 작성한 사람의 닉네임, 내용, 리뷰를 가져온 플랫폼 출력		

요구사항 고유번호		SIR007		
요구사항 명칭		영화 감상 포인트 인터페이스	응락수준	필수
요구사항 분류		인터페이스 요구사항		
요구사항 상세 설명	정의	- 선택한 영화의 감상 포인트 출력 인터페이스		
	세부 내용	<ul style="list-style-type: none"> <li>- 선택한 영화를 분석해 해당 영상의 감상 포인트 비중을 “연기력”, “영상미”, “연출”, “OST”, “스토리”를 원 그래프를 통해 시각화</li> <li>- 각 감상 포인트에 대해 긍정적인 반응과 부정적인 반응의 정도를 가로 막대 그래프를 통해 시각화</li> </ul>		

## 6. 데이터 요구사항

요구사항 고유번호	DAR-001		
요구사항 명칭	데이터 구축		
요구사항 분류	데이터 요구사항	응락수준	필수
요구사항 상세 설명	<ul style="list-style-type: none"> <li>- 영화별 감상 포인트 추출 모델 실행 결과를 데이터베이스 테이블로 저장한다.</li> <li>- 테이블은 감상 포인트와 감상 포인트에 대한 긍·부정 개수로 구성된다.</li> </ul>		

## 7. 테스트 요구사항

요구사항 고유번호	TER-001		
요구사항 명칭	테스트 방안	응락수준	필수
요구사항 분류	테스트		
요구사항 상세 설명	정의	- 단위테스트 요구사항	
	세부 내용	<ul style="list-style-type: none"> <li>- 시스템은 최소 기능인 영화 리뷰 필터링을 베이스 라인으로 간주한다.</li> <li>- 사용자가 검색한 영화에 대한 리뷰를 크롤링해 필터링된 리뷰와 감상 포인트별 가이드라인을 제시하는 것을 목표로 한다.</li> <li>- 테스트를 위해서 임의의 영화를 선정하여 명세한 요구사항을 검증 활동을 통해 평가한다.</li> <li>- 기능 구현 정확성은 실제 사용자가 직접 테스트 수행 기간에 테스트를 수행함으로써 평가한다.</li> </ul>	

요구사항 고유번호	TER-002		
요구사항 명칭	단위 테스트	응락수준	필수
요구사항 분류	테스트 요구사항		
요구사항 상세 설명	정의	- 단위테스트 요구사항	
	세부 내용	<ul style="list-style-type: none"> <li>- 기능 요구사항에 기술한 요구사항에 대하여 단위 테스트 수행</li> <li>- 각 단위 테스트에 대하여 시나리오, 처리 절차, 수행 데이터, 예상 결과 등으로 사전에 정의하고 수행</li> </ul>	



## 8. 보안 요구사항

요구사항 고유번호		SER-001		
요구사항 명칭		API서버 보안	응락수준	필수
요구사항 분류		보안 요구사항		
요구사항 상세 설명	정의	- API서버 보안		
	세부 내용	- 해당 API서버를 사용함에 있어서 허가된 origin이 아닌 곳에서 정보를 요청할 경우 결과를 반환하지 않는다.		

요구사항 고유번호		SER-002		
요구사항 명칭		API키 보안	응락수준	필수
요구사항 분류		보안 요구사항		
요구사항 상세 설명	정의	- KMDB API키 보안		
	세부 내용	- KMDB의 API키를 코드 내부에 포함시켜 배포하지 않는다. - 해당 API키를 서버에 위치시켜 API형식으로 내려받도록 한다.		

## 9. 품질 요구사항

요구사항 고유번호		QUR-001		
요구사항 명칭		필터링 검출률	응락수준	필수
요구사항 분류		품질		
요구사항 상세설명	정의	인공지능의 성능(기술 관점)		
	세부 내용	- 인공지능을 통해 영화 리뷰를 필터링할 때, 정확도가 0.95 이상의 값을 갖도록 한다.		

요구사항 고유번호		QUR-002		
요구사항 명칭		감상포인트 검출률	응락수준	필수
요구사항 분류		품질		
요구사항 상세설명	정의	인공지능의 성능(기술 관점)		
	세부 내용	- 감상 포인트별 감성 분석을 수행할 때, Aspect Polarity Classification과 Aspect Term Extraction의 F1-score가 0.9 이상의 값을 갖도록 한다.		

## 10. 제약 사항

요구사항 고유번호		COR-001		
요구사항 명칭		시스템 개발	응락수준	필수
요구사항 분류		제약사항		
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"> <li>- 프로그래밍 언어: Python</li> <li>- 사용 라이브러리: PyABSA, Pytorch, Konlpy</li> <li>- 개발도구: Jupyter Notebook</li> </ul>		

요구사항 고유번호		COR-002		
요구사항 명칭		데이터셋 취득	응락수준	필수
요구사항 분류		제약사항		
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"> <li>- 쉽게 취득할 수 있는 일반적인 감성 분석 데이터셋과 다르게 한글로 이루어진 속성기반 감성 분석(ABSA) 데이터셋 취득 제한으로 인해 모델에 학습하기 충분한 양의 데이터 구축이 필요하다.</li> </ul>		

요구사항 고유번호		COR-003		
요구사항 명칭		업무 모듈화 및 자원 활용 방안	응락수준	필수
요구사항 분류		제약사항		
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"> <li>- 유연성, 확장성을 확보할 수 있도록 모듈화 개발전략을 반영한다.</li> <li>- 현재 활용 가능한 소프트웨어를 최대한 활용하여 수행한다.</li> </ul>		

## 11. 프로젝트 관리 요구사항

요구사항 고유번호		PMR-001		
요구사항 명칭		프로젝트 수행 조직	응락수준	필수
요구사항 분류		프로젝트 관리		
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"> <li>- 한재혁 : Vue.js를 이용한 웹 페이지 구축 및 node.js를 이용한 API 작성과 서버 구축</li> <li>- 서민석 : PyABSA를 활용한 한국어 감상 포인트 추출 모델 개발</li> <li>- 정인상 : 감상 포인트 추출 모델 학습을 위한 데이터셋 구축</li> </ul>		

요구사항 고유번호		PMR-002		
요구사항 명칭		프로젝트 일정계획	응락수준	필수
요구사항 분류		프로젝트 관리		
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"> <li>- 영화 리뷰 필터링 모델 개발 : 9월 이전</li> <li>- 모델 테스트 및 웹 디자인 개선 : 10월 이전</li> </ul>		