2020학년도 1학기 종합설계 교과목
캡스톤 디자인 결과보고서

소 속 (전공)	디지털콘텐츠학과	
팀 명	TMI	
지 도 교 수	권 순 일 (인)	
팀 장	학번: 15011139	이름: 심재경
팀 원	학번: 15011136	이름: 신현욱
	학번: 15011137	이름: 김지수
	학번:	이름:
	학번:	이름:
	학번:	이름:
제 출 일 자	2020. 06. 25	

세종대학교 공학교육센터

목 차

5. 시험 (시험 결과 기술)

5.1 최종 결과물에 대한 시험 결과

6. 평가

6.1 작품의 완성도 및 기능 평가
6.2 기대효과 및 영향
6.3 작품제작 후기
6.4 팀 개요 및 역할분담
6.5 참고문헌

[첨부 1] 작품 사진 첨부

볼드체의 각장과 각절의 제목은 되도록 유지하여 주시기 바라며 그 밖의 소제목은 지도 교수의 지도하에 가감 또는 조정이 가능합니다.

제 1 장 종합설계 개요

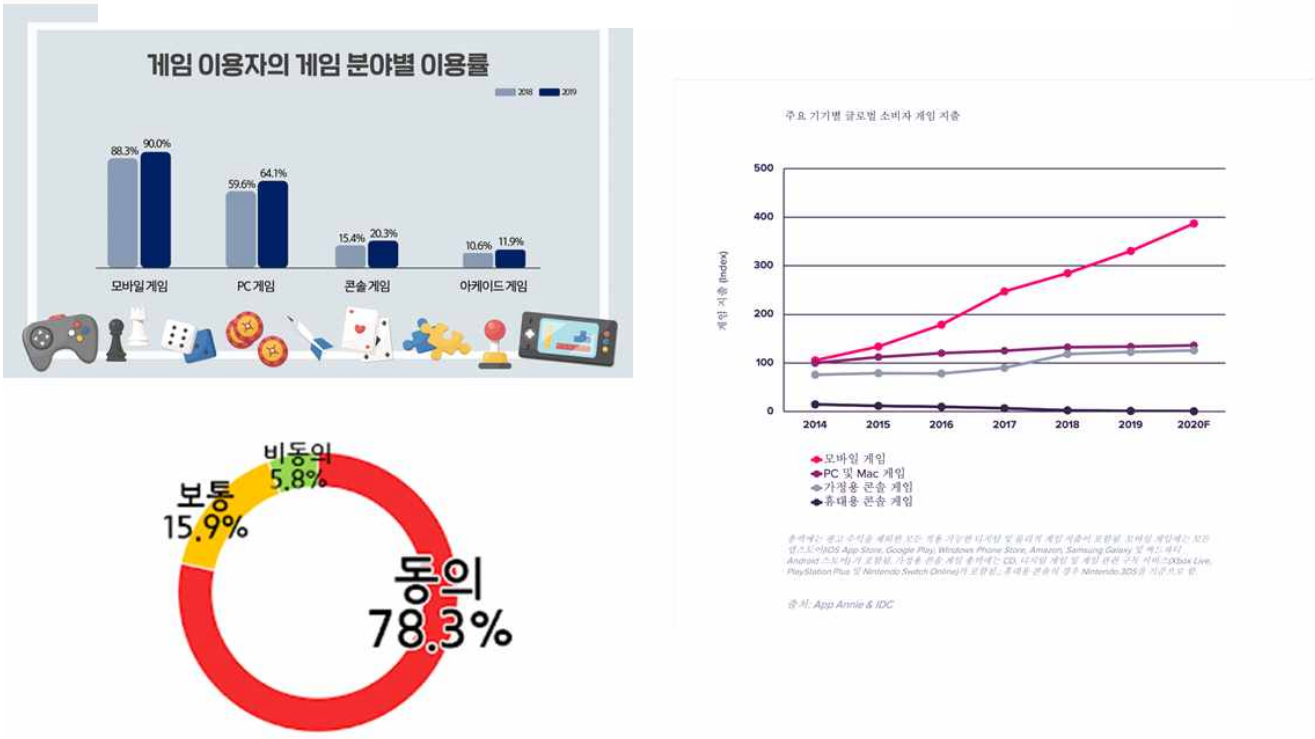
1. 종합 설계 제목

제목 : 단어 유사도 측정을 통한 게임 어플리케이션 리뷰 분석
(부제 : 리뷰 속 단어 유사도 측정을 통한 카테고리 분류 및 감정 분석)

2. 설계의 개요(Abstract or Concept)

현재 모바일 시장의 규모가 커지고 있는 상황에서 개발자는 리뷰를 통해서 사용자로부터 게임의 발전을 위한 피드백을 얻어야 합니다. 따라서 저희는 유사도에 따라 리뷰들을 정해 놓은 카테고리로 분류하여서 특정 카테고리와 높은 유사도를 보이는 리뷰를 보아서 볼 수 있는 기능과 각 리뷰마다 4단계의 감정(긍정, 약한 긍정, 약한 부정, 부정)으로 분석 되는 기능을 통해 게임 어플리케이션에 대한 리뷰를 분석해주는 프로젝트를 기획했습니다.

3. 설계의 배경 및 필요성



게임 이용자의 게임 분야별 이용률-한국콘텐츠진흥원(왼쪽 위),
제품 구매 시 항상 고객들의 리뷰를 확인한다.-트랜드모니터(왼쪽 아래),
주요 기기별 글로벌 소비자 게임 지출-APP Annie&IDC(오른쪽)

현재 게임시장은 시간이 지날수록 모바일 게임의 비중이 압도적으로 증가하고 있습니다. 특히 게임시장에서 2번째 비중을 차지하고 있는 PC게임들도 날이 갈수록 모바일 시장을 겨냥하여서 PC와 모바일간의 크로스 플레이[1]를 지원하는 게임의 비중도 커지고 있습니다. 특

히 모바일 게임 시장에서는 PC게임이나 콘솔 게임시장보다 더욱 손쉽게 게임에 대한 평가를 리뷰를 통해 쉽게 작성할 수 있고, 다른 사용자들의 리뷰도 쉽게 열람이 가능하며 개발자도 리뷰를 통해 소비자들의 평판을 쉽게 접할 수 있습니다. 여기서 게임의 리뷰는 사용자와 사용자간의 소통공간이기도 하지만 개발자와 사용자의 소통공간으로써 개발자가 게임의 발전을 위한 피드백을 얻을 수 있는 공간으로 게임 개발의 있어서 매우 중요한 의미를 가집니다. 특히 대부분의 소비자가 게임 구매 이전의 리뷰들을 확인하기 때문에 개발자 입장에서 리뷰에서 요구하는 사항들에 대해 빠른 피드백을 제공하지 않는다면 문제가 발생할 수 있습니다.

그러나 이러한 상황 속에서 리뷰의 수는 점점 많아지고 있고 현재 기존의 어플리케이션 분석도구들은 주로 사용자의 인적사항, 사용시간, 설치한 APP등의 <사용자 위주>의 분석만 제공하고 있고 리뷰에 대한 분석은 자세한 분석이 아닌 감정분석 정도로만 이루어지고 있습니다. 그래서 저희는 기존의 분석도구보다 리뷰자체의 집중한 분석도구를 만들고자 했고 수만 또는 수십만 개의 리뷰들에 대한 빠른 접근을 위해서 단어 유사도를 통한 특정 카테고리 분류방법을 통하여 개발자가 해당 카테고리에 대한 리뷰들을 좀 더 빠른 시간 안에 편리하게 열람할 수 있도록 하는 프로젝트를 기획하게 되었습니다.

---

1) 크로스 플레이(cross-play) : 플랫폼에 관계없이 온라인상에서 함께 게임을 즐길 수 있게 하는 것

제 2 장 설계의 현실적 제한 조건 기술

1. 현실적 제한 조건과 이에 따른 고려 내용의 기술(필요시 자료 첨부)

(본 과제 수행함에 있어 고려할 현실적 제한조건을 기술)

현실적 제한조건		
제한요소		**고려할 내용**
1. 산업표준	설계 제작품의 산업 표준 규격 참조	해당 사항 없음.
2. 경제성	가능한 한 저렴한 비용과 주어진 여건 아래에서 제작	해당 사항 없음.
3. 윤리성	참고 문헌/제품 인용 표시	(6.5 참고 문헌에서 명시)
4. 안전성	안전하게 구현	해당 사항 없음.
5. 신뢰성	지속적으로 구동	크롤링을 통해 지속적으로 자료 수집 가능하고, 수집 자료 기반으로 지속적인 분석 가능.
6. 미학	가급적 공학적 실용성을 갖춘 외형 구비	해당 사항 없음.
7. 환경에 미치는 영향	환경 유해 물질의 사용과 설계 제작품의 폐기 시 절차 규정	해당 사항 없음.
8. 사회에 미치는 영향	사회 전반에 유익한 영향을 미치는 설계 제작품 창작 및 적용 분야 명기	사용자들이 작성하는 리뷰들이 실제로 피드백으로서 이용될 수 있음을 상기시키고, 건전한 리뷰 문화가 형성되도록 장려.
9. 기타	지역 특성화 산업과 연계성 고려	해당 사항 없음.

제 3 장 설계 구성 요소에 따른 결과 기술

◉ 설계의 구성 요소 체크 항목

(본 과제 수행함에 있어 고려할 설계의 구성요소 기술, 필수 항목)

설계 구성요소		
구성요소		**실시여부**
1. 목표 설정	- 브레인스토밍 등의 아이디어 창출 도구를 이용하여 설계 목표를 설정 - 현실적인 제한 요소와 공학적인 제한 요소를 감안하여 설정	실시 완료
2. 합성	- 설계목표에 달성에 필요한 관련 기술을 조사 분석하여 제작 가능한 설계안 제시 (작품의 개념을 1차 합성함)	실시 완료
3. 분석	- 다양한 방법으로 자료를 수집하고, 포괄적인 문제에 대한 분석 또는 결과물에 대한 유용성 분석을 실시 - 다양한 도구를 이용하여 설계서 작성 및 주요 부분에 대한 해석 결과 제시	실시 완료
4. 제작	- 공학실무에 필요한 기술 방법, 도구들을 사용하여 설계서에 따른 제작, 혹은 프로그램 작성	실시 완료
5. 시험	- 최종 결과물에 대한 시험 - 안전하고 지속적으로 구동가능한가를 테스트	실시 완료
6. 평가	- 최종 시작품이 설계 가이드라인을 만족하고 결론이 일치하는지 평가하고 일치하지 않을 경우 개선 방안 고찰 - 발표 능력 평가	실시 완료

1. 목표 설정

1.1 문제 해결을 위한 아이디어 및 구체적인 방법

모바일 게임 개발자 혹은 관계자들에게 해당 게임 어플리케이션 사용자들의 리뷰를 분석 및 분류하여 제공해 리뷰에 대한 내용을 쉽게 파악할 수 있도록 해주고자 기획한 프로젝트입니다. 해당 분석 자료를 통해 개발 시에 적절한 피드백을 할 수 있고, 신규 개발자에겐 좋은 참고 자료가 될 수 있도록 하는 것이 주목적입니다.

리뷰에 대해서 분석하기 위해, 해당 프로젝트에서는 단어 유사도 측정 방식과 딥러닝 방식을 이용한 감정 분석을 이용하였습니다.

우선 단어 유사도 측정 방식은 Word2Vec모델을 이용해 그 결과로 나온 단어 벡터들을 사용하여 가중치 행렬로 변환하였고, 이를 통해 리뷰와 각 카테고리 간의 유사도를 계산하였습니다. 그 결과로 해당 리뷰가 가지고 있는 내용과 비슷한 카테고리로 분류하는 시스템을 도출해냈습니다.

또한 감정분석 방식은 딥러닝을 통한 분류 방식을 이용합니다. 분류 방식을 이용하기 위해서는 먼저 리뷰에 대해서 긍정 또는 부정을 숫자로 나타내주는 라벨링이 필요합니다. 따라서 (특정)기준을 통해 긍정 = 1, 부정 = 0으로 라벨링 작업을 거쳐서 Train Data로 만들고, 라벨링이 없는 리뷰들을 Test Data로 만듭니다. 이 Data들을 이용하여 딥러닝 학습을 통해 새로운 Input값에 대한 Score값을 계산해주고 이를 바탕으로 긍정, 부정을 판단해줍니다.

최종적으로 게임 어플리케이션 관계자가 사용자들이 작성한 각 리뷰들이 어떤 카테고리와 관련된 내용인지 알 수 있고, 해당 리뷰의 감정 상태가 어떠한 지 파악할 수 있어 사용자들의 니즈를 파악하는데 도움을 줄 것입니다.

1.2 수행목표

유사도 분석에 의한 카테고리 분류는 각 단어별 벡터를 이용해 단어 간의 거리를 측정할 수 있으며, 이를 통해 유사한 의미를 가진 단어들을 포함하는 리뷰들이 특정 카테고리에 정확히 분류되는 것을 목표로 하였습니다. 또한, 리뷰의 특성상 하나의 카테고리 이외에 여러 가지 카테고리로 분류될 수 있으므로 최대 세 개의 카테고리로 분류되는 것을 목표로 하였습니다.

감정 분석에 대해서는 0에서 1사이의 값의 스코어로서 나타낸 분석 결과를 적절한 값으로 나누어 4단계 감정 상태(긍정, 약한 긍정, 약한 부정, 부정)로 분석하는 것을 목표로 하였습니다.

2. 합성

2.1 기초 조사

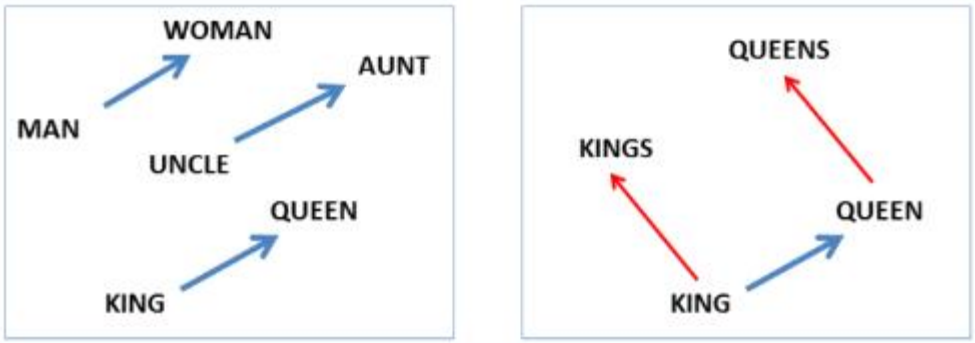
2.1.1 관련 분야의 이론 및 기술 현황 조사

단어 임베딩(Embedding)은 단어(Word)를 벡터(Vector)로 바꾸어주는 방법으로 크게 Word2Vec, GloVe, Fasttext 세 가지 방법이 존재합니다. 이 세 가지 방법들은 모두 단어 동시 등장 정보를 보존(Word's of Co-occurrence)하는 것이 특징입니다.

그 중 Word2Vec은 Distributional Hypothesis[2)]에 근거한 방법론으로, 비슷한 위치에 등장하는 단어들은 그 의미도 유사할 것이라는 의미를 가지고 있습니다.

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{W} \exp(u_w^T v_c)}$$

Word2Vec은 위의 식을 최대화하는 것을 목표로 하고 있는데, o는 주변 단어(context word), c는 중심 단어(center word)입니다. 다시 말해 p(o|c)는 중심 단어(c)가 주어졌을 때 주변 단어(o)가 등장할 조건부확률을 뜻합니다. 즉, 위의 식을 최대화하는 것은 중심 단어로 주변 단어를 잘 맞춘다는 의미를 가집니다.



(Mikolov et al., NAACL HLT, 2013)

GloVe는 2014년 미국 스탠포드 대학 연구팀에서 개발한 단어 임베딩 방법론으로, GloVe로 임베딩된 단어 벡터끼리의 내적은 동시 등장 확률의 로그 값을 나타내줍니다. 또한 2016년 페이스북이 발표한 Fasttext는 원래 단어를 부분 단어(Subword)의 벡터들로 표현한다는 점을 제외하고는 Word2Vec과 유사합니다.

이렇게 단어 임베딩을 하기 위한 토큰[3]화를 진행하기 전에는 단어들을 포함하고 있는 문장들의 적절한 정제 및 정규화 작업이 필요한데, 이를 데이터 전처리 과정이라고 합니다. 데이터 전처리 과정에는 표기가 다른 단어들을 통합하고, 영어 문장에서는 대소문자를 통합하며, 분석하고자 하는 목적에 맞지 않는 불필요한 단어를 제거하는 불용어 제거 과정을 거칩니다. 불용어 대상에는 등장 빈도가 적은 단어, 길이가 너무 짧거나 긴 단어, 단일 자모음, 제2외국어, 이모티콘과 같은 특수문자 등이 있습니다.

2.1.2 현 상황에서의 문제점 또는 해결이 필요한 사항

Word2Vec 방식은 영어를 기반으로 만들어진 방식이어서 한국어 리뷰에 대해 적용하려면 몇 가지 문제점이 존재합니다. 한국어 특성상 영어보다 띄어쓰기가 잘 지켜지지 않고 또한 한국어는 단어의 문법적 기능을 표현하고자 조사를 사용하며 이를 단어에 붙여 쓰지만 영어는 전치사를 따로 떼어 사용하는 모습을 볼 수 있습니다. 예를 들면 영어로 he/him 이 들어간 문장이 있다고 하면 이 경우 영어의 경우 he/him 뒤에 바로 띄어쓰기가 존재하지만 한

2) 분포 가설(Distributional Hypothesis)은 동일한 맥락에서 발생하는 단어들이 유사한 의미를 갖는 경향이 있다는 것이다.(Harris, 1954)
3) 토큰 : 문법적으로 더 이상 나눌 수 없는 언어요소

국어는 그라는 단어 뒤에 '그가', '그에게', '그를', '그는'과 같이 다양한 조사가 '그;라는 글자 뒤에 띄어쓰기 없이 바로 붙게 됩니다. 자연어 처리과정에서 같은 단어임에도 서로 다른 조사가 붙어서 다른 단어로 인식되어버리면 자연어 처리가 번거로워지기 때문에 이를 보완할 수 있는 Tokenizing 전처리 단계를 거쳐야합니다.

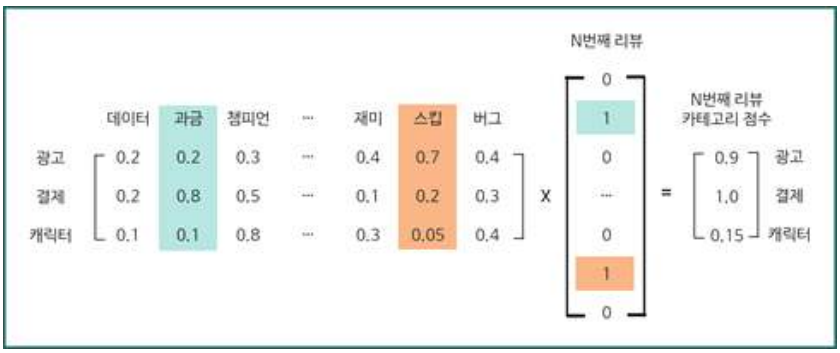
2.2 개념의 합성(개념 설계)

2.2.1 작동원리

우선 리뷰를 분석하기 전에, 리뷰에 대한 충분한 학습이 필요하기 때문에 충분히 많은 양의 리뷰가 필요하게 됩니다. 따라서 구글 플레이스토어에서 총 약 24만개의 리뷰의 (작성자, 리뷰 내용, 작성 날짜)에 대한 내용을 크롤링해서 가져옵니다. 이 과정에서 리뷰의 내용이 편향되지 않게 하기 위해 인기 게임 1위부터 200위까지의 게임 어플리케이션의 리뷰를 각각 약 2,000개씩 분할하여 수집하였고, 리뷰 개수가 2,000개가 되지 않는 경우 가져올 수 있는 모든 리뷰를 수집하는 방식을 이용했습니다.

그 다음으로는 Train Data로서 가져온 약 24만개의 리뷰를 전처리 과정을 거칩니다. Pandas 라이브러리를 통해 데이터를 분리시키고 문장 내에서 내용과 관계없는 Stopwords를 제거하였습니다. 그 후, Konlpy 한국어 형태소 분석기를 이용해 문장 단위의 리뷰를 형태소 단위로 토큰화를 진행하였습니다.

단어 토큰들을 이용해 Word2Vec 모델을 적용하여 나온 N개의 단어들을 100차원의 벡터로 표현하였고, 이를 (NX100) 단어 좌표 행렬로 표현했습니다. 그 이후 유클리디언 거리 방식을 이용해 (NXN) 단어 간 거리 행렬을 도출해냈고, 단어 간 거리 행렬에서 단어간의 거리가 가까우면 높은 가중치를 두고, 그렇지 않으면 낮은 가중치를 두는 가중치 행렬(Weight Matrix)로 표현을 하였습니다. 그 중에서 가중치 행렬에서 분류하고자 하는 카테고리를 뽑아내어 특정 쿼리 단어로만 구성된 (36XN) 가중치 행렬을 만들어 냈습니다.

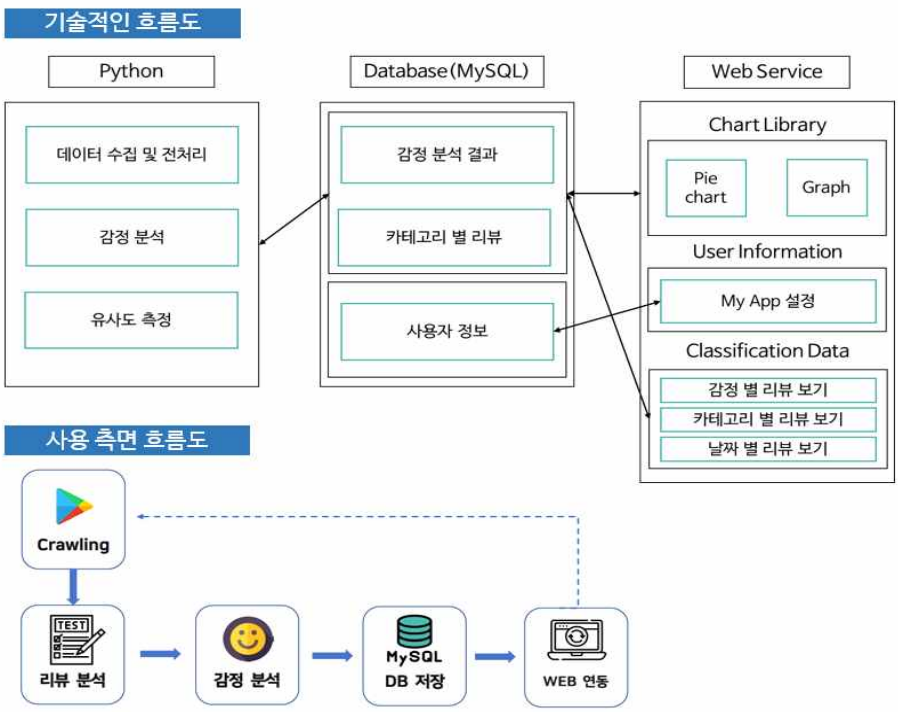


특정 쿼리 단어로 구성된 가중치 행렬과 TDM을 내적하여 카테고리 Score를 계산하는 모습

다음으로는 각 리뷰에서 N개의 단어 중 포함하는 단어가 있으면 1, 없으면 0으로 나타내는 Binary Matrix인 TDM(Term-Document Matrix)을 구축하였고, 이전에 구현한 가중치 행렬과 내적 하여 계산해 카테고리별 스코어를 계산하였습니다. 계산된 결과에서 특정 값을 넘는 카테고리 중 최댓값을 가지는 3개의 카테고리로 분류하여 유사도 분석에 따른 리뷰 분류를 제공하였습니다.

감정분석의 경우 원하는 리뷰를 전처리 과정을 거친 후 Input으로 입력했을 때 딥러닝 모델을 통해 학습을 하고 Output을 출력해주는데 이를 위해 전처리 과정에서 사용 될 불용어 사전과 전처리 과정을 끝낸 리뷰 데이터가 필요합니다. 사용자가 리뷰를 입력했을 때 전처리 과정을 끝낸 리뷰 데이터로 부터 가장 빈도수가 높은 N개의 토큰을 이용하여 딥러닝을 진행합니다. 딥러닝 과정은 3개의 층으로 구성해서 1번째 층에서는 N개의 Feature로 부터 64개의 Feature를 추출해내고 2번째 층에서는 동일하게 16개의 Feature로부터 새로운 16개의 Feature을 추출해내고 3번째 층에서 16개의 Feature로부터 1개의 확률(Score)를 출력해주는 신경망 모델을 통해 결과 값을 추출합니다. 추출된 결과 값 0에서 1사이의 값을 0.25씩 4단계로 나누어 감정 분석 4단계(긍정, 약한 긍정, 약한 부정, 부정)에 대한 분석을 제공하였습니다.

2.2.2 논리적인 구조도



2.2.3 주요 기능

가. 어플리케이션 리뷰에 대한 카테고리 분류 제공

분석하고자 하는 모바일 게임 어플리케이션에 대한 리뷰들에 대해 카테고리별로 분류를 제공해줍니다. 특정 카테고리로 분류할 때에는 카테고리 단어와 특정 단어와의 유사도 관계를 측정할 필요가 있습니다.

우선 유사도를 측정하기 위해서 특정 쿼리 단어와 단어사이의 가중치를 구해, 각 리뷰마다 얼마나 많은 가중값을 가지고 있는지 측정합니다. 이때 특정 쿼리 단어를 뽑아낸 기준은 Word2Vec모델 적용 시에 빈도수가 높은 단어 중 카테고리로 분류 가능한 단어들을 위주로 선택하였습니다. 이렇게 측정된 가중값을 통해 카테고리로 분류를 진행하였습니다.

이 중 모바일 리뷰의 특성상 특정 카테고리로 분류하기 어려운 단순 리뷰들이 많이 존재하게 되는데. 예를 들어, “그럭저럭이네요.”, “재밌어요”, “나쁘진 않아요.”와 같은 특별한 내용이 존재 하지 않고 감정만 포함하고 있는 리뷰들이 다수 존재합니다. 이와 같은 리뷰들을 가중값이 낮게 측정되는데, 따라서 어느 카테고리에도 높은 가중값을 보이지 않는 리뷰들은 ‘단순 리뷰’로 분류하여 카테고리 분류 대상에서 제외시켰습니다.

‘단순 리뷰’로 분류된 리뷰 모습(왼쪽), 최대 3개의 카테고리로 분류되는 모습(오른쪽)

또한, 하나의 리뷰에 여러 가지 카테고리를 포함하고 있을 수도 있는데, 이 점을 해결하기 위해 하나의 리뷰 당 3개의 카테고리까지 분류가 가능하도록 설정하였습니다. 특정 가중값(0.2)을 넘는 카테고리 중 오름차순으로 정렬하여 최댓값을 가지는 3개의 카테고리로 분류하였습니다.

그렇게 분류된 카테고리를 기준으로 각 카테고리 별로 리뷰의 개수를 카운팅하여 원형 차트와 바 그래프로 나타내었고, 각 카테고리로 분류된 리뷰들에 대해 자세히 살펴볼 수 있도록 리뷰 상세 보기 페이지와 연동하여 열람 가능하게 구현하였습니다.

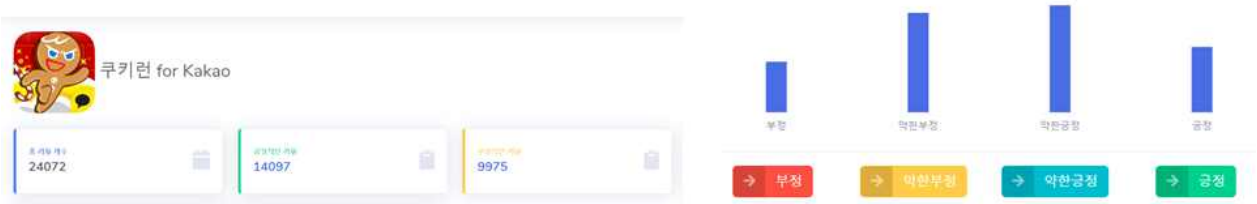
원형 차트로 나타낸 상위 카테고리(왼쪽), 바 그래프로 나타낸 하위 카테고리(오른쪽)

나. 리뷰에 대한 감정 분석 결과 제공

현재 해당 어플리케이션의 리뷰에 대한 감정 분석 결과를 제공해줍니다. 사용자들이 어플리케이션을 사용하고 남긴 리뷰가 긍정적이면 긍정의 상태로 분석해주고, 부정적이면 부정의 상태로 분석해 결과를 제공합니다.

딥러닝 모델을 통해 학습하여 나온 결과를 이용해 감정 상태를 나누었습니다. 이 과정에서 리뷰의 특성상 자유 양식이기 때문에. 한 리뷰 안의 여러 가지 감정을 포함할 수 있기에 긍정, 부정 2단계의 감정 분석을 넘어 약한 긍정과 약한 부정의 상태를 추가해 총 4단계로 감정 분석을 진행했습니다.

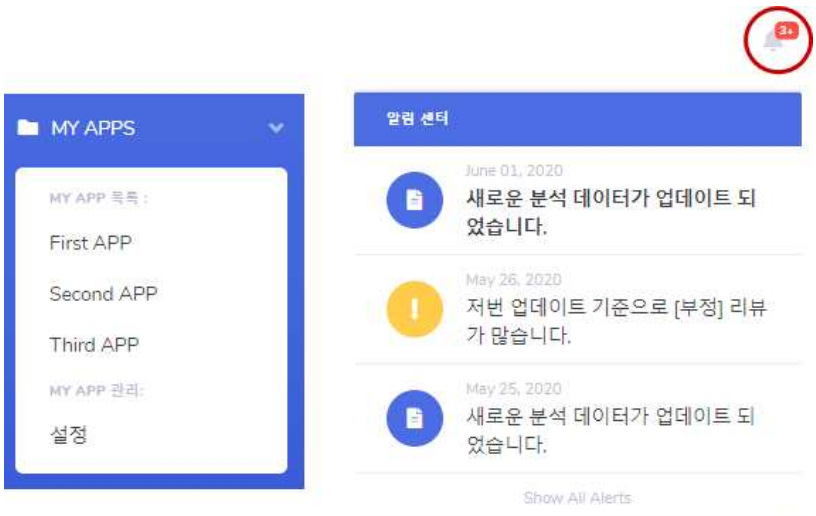
4단계의 감정 분석을 진행한 결과를 토대로, 어플 메인 화면에서 리뷰 요약 정보를 통해 총 긍정-부정 리뷰의 개수를 카운트하여 정보를 제공합니다. 또한, 하위 카테고리 리뷰 상세보기 페이지에서 감정 분석 보기 버튼을 통해 4단계의 감정 상태를 바 그래프 형태로 한눈에 보기 쉽게 정보를 제공하였습니다. 추가로, 감정 상태에 따른 리뷰 열람도 가능합니다.



전체 리뷰의 감정 분석 2단계 결과(왼쪽), 하위 카테고리의 감정 분석 4단계 결과(오른쪽)

다. 관심 앱 설정을 통한 빠른 정보 제공

프로그램 사용자가 관심 앱으로 설정해둔 어플의 리뷰에 대한 정보를 검색 없이 빠르고 쉽게 제공받을 수 있으며, 차트 페이지를 통해 더 자세한 분석 정보를 제공 받을 수 있습니다. 관심 앱 설정은 초기 회원가입 시에 설정할 수 있고, 회원가입 이후에도 마이 앱 설정 페이지에서 최대 5개까지 마이 앱을 등록 및 수정 할 수 있습니다. 또한, 마이 앱으로 설정

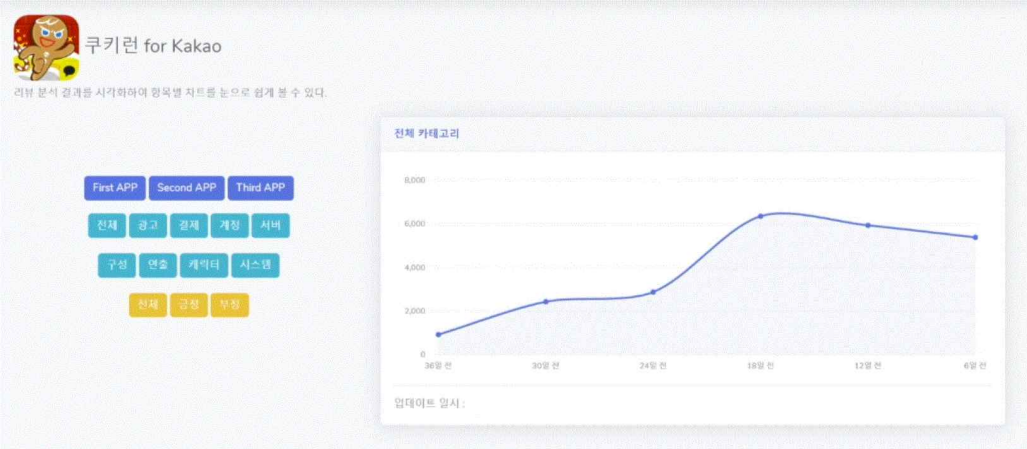


MY APP 설정(왼쪽), 알림 메시지(오른쪽)

해둔 어플에 대한 분석 정보가 업데이트되거나 특별한 이벤트가 발생하면 알림 기능으로 정보를 받을 수 있습니다.

라. 날짜 별 리뷰의 동향 파악 가능

좌측 메뉴의 Charts 메뉴를 통해 미리 설정해둔 마이 앱에 대한 자세한 날짜별 분석 정보를 제공 받을 수 있습니다. 분석하고자 하는 마이 앱 중 하나의 어플을 선택하고 카테고리와 감정 상태를 선택하면 해당 조건의 리뷰 개수가 날짜별로 분석된 시각화 그래프로 나타내어집니다. 이를 통해 프로그램 사용자는 업데이트 혹은 개발 후에 어플 사용자들이 어떤 반응을 보이고 있는지 동향 파악이 가능합니다.



Charts 페이지의 날짜별 리뷰의 시각화 그래프

3. 분석(작품 구현 과정 중의 문제점 분석 및 해결 방법)

3.1 과제수행에 사용된 이론 및 기술의 조사 및 분석 결과

3.1.1 본 과제를 수행함에 있어 활용된 수학, 기초과학

가. 유클리디언 거리

유클리디언 거리 공식이란 n차원 공간에서 두 점간의 거리를 알아내는 공식으로, 저희는 100차원 공간에 표현 된 각 단어들 간의 거리를 구해주기 위해 유클리디언 거리 공식을 사용했습니다. 유클리디안 거리 공식은 다음과 같습니다.

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

나. 정규분포의 확률밀도 함수

0부터 무한대의 거리로 표현되는 단어의 거리 행렬을 가지고 유사도를 측정에 어려움을 겪어 이를 0부터 1까지의 스케일링이 필요함을 인지했습니다. 이를 해결하기 위해 정규분포의 확률밀도 함수에서 계수를 제외한 부분을 이용하여 스케일링을 하여 가중치 행렬을 구축하였습니다. 저희가 사용한 공식은 다음과 같습니다.

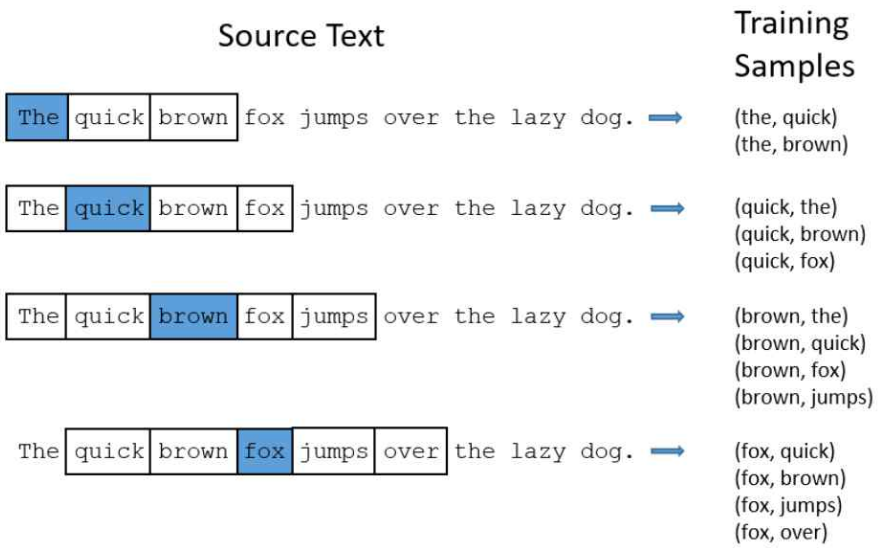
$$W_{ij} = exp\left(-\frac{d(x_i, x_j)^2}{2\sigma^2}\right)$$

δ : 거리행렬의 분산

3.1.2 본 과제를 수행함에 있어 활용된 전공 이론과 정보기술

가. Word2Vec[4)]

Word2Vec이란 단어의 관계를 두 계층의 신경망에서 추정해 벡터공간에 나타낸 모델입니다. 특징 공간 안에서 비슷한 관계에 있는 데이터의 특징을 파악해 제공합니다. Word2Vec은 Word Embedding을 위해 CBOW[5)],와 Skip-Gram 두 가지의 모델을 제시하는데 저희는 크기가 큰 Data Set에 조금 더 적합한 Skip-Gram 방식을 채택하였습니다. Skip-Gram이란 현재 주어진 단어 하나를 가지고 주위에 등장하는 나머지 몇 가지의 단어들의 등장 여부를 유추하는 방법입니다. 이 때 예측하는 단어들의 경우 현재 단어 주위에서 샘플링 하는데, '가까이 위치해있는 단어일수록 현재 단어와 관련이 더 많은 단어일 것이다'라는 생각을 적용하기 위해 멀리 떨어 져있는 단어일수록 낮은 확률로 택하는 방법을 사용합니다. Skip-Gram 방식을 간 단한 예로 설명하자면,



Skip-gram의 step 흐름도

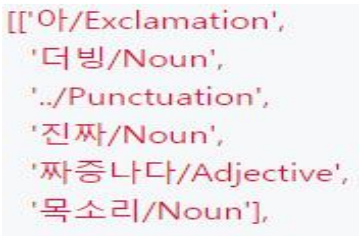
4) word2vec 방법론에 대한 논문 : https://arxiv.org/pdf/1301.3781.pdf
5) CBOW : 맥락으로부터 타깃을 추측하는 용도의 신경망

'The quick bronw fox jumps over the lazy dog'이라는 문장을 학습하고자 할 때, 한 번에 학습할 단어의 개수인 Window는 2로 설정되어 있다고 했을 때의 예입니다. 첫 번째 스텝에서 중신단어는 처음 등장하는 단어인 'The'입니다. Window의 크기가 2이기 때문에 중심단어 앞뒤로 두 개씩 봐야 하지만, 'The'를 기준으로 이전 단어가 존재하지 않으므로 어쩔 수 없이 뒤에 등장하는 두 개 단어(quick, brown)만 학습 됩니다. 이렇게 첫 번째 스텝이 끝나면 중심단어를 오른쪽으로 한 칸 옮겨 'quick'을 중심단어로 하고, 'The', 'brown', 'for'를 각각 주변단어 정답으로 두는 두 번째 스텝을 진행하게 되며 이런 식을 계속 모든 단어들을 슬라이딩해가며 학습하는 방식입니다. Skip-Gram의 경우 Window가 2일 때 중심단어의 업데이트 기회가 4번이나 확보된다는 점에서 CBOW보다 큰 Data Set에서 좋은 성능을 보여주고 있습니다.

나. 감정분석

NSMC(naver sentiment movie corpus)를 데이터로 가져와서 데이터를 학습하기에 알맞게 전처리를 실시해줍니다. 전처리 방법으로 KoNLPy 라이브러리 중에 Okt(Open Korean Text) 클래스를 이용하였습니다.

리뷰 특성상 맞춤법이나 띄어쓰기가 제대로 되어있지 않는 경우가 많이 존재하는데 KoNLPy는 띄어쓰기 알고리즘과 정규화를 이용해서 맞춤법이 틀린 문장도 어느 정도 고쳐주면서 형태소 분석과 품사를 태깅해주는 여러 클래스를 제공해줌으로 KoNLPy를 사용하였습니다. 따라서 먼저 KoNLPy를 사용하여 형태소 분석을 통해서 태깅된 품사를 이용하여 토큰화 해줍니다.

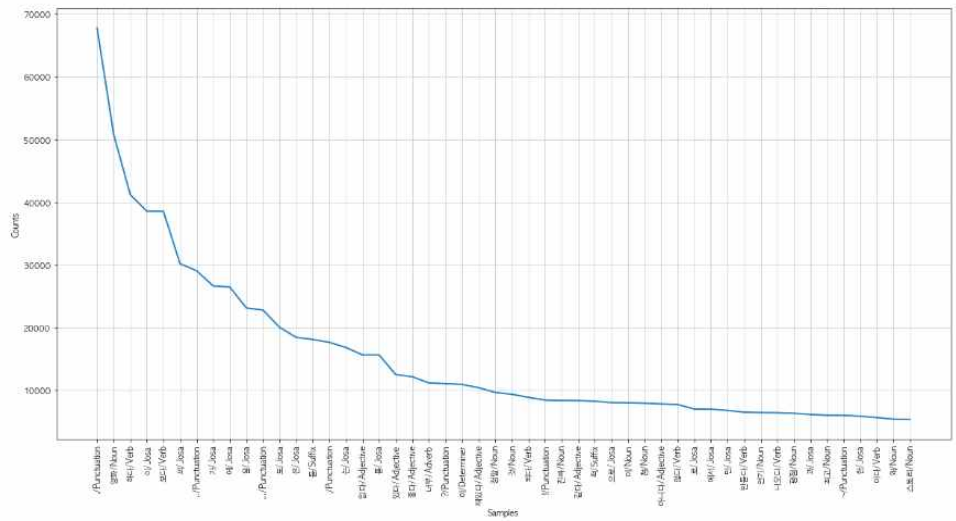


토큰별로 품사 태깅

그 다음 사용 빈도수가 많은 토큰들을 이용하여 딥러닝 모델이 쓰일 train , test 데이터를 만들어줘야 하는데 이 과정에서 문서를 편리하게 탐색하기 위하여 nltk 라이브러리에 Text클래스를 사용했습니다.

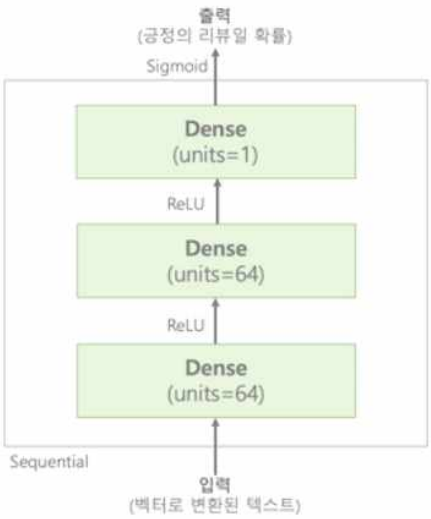
Text 클래스에서 vocab().most_common 메소드는 데이터에서 가장 자주 사용되는 단어를 가져와주는 메소드인데 이것을 활용하여 자주 사용되는 토큰(n)개를 사용해서 데이터를 백터화 시키고 이를 이용하여 딥러닝 모델 정의 및 설계 작업을 진행해줍니다.

저희는 NSMC로부터 가져온 데이터수를 고려하여 토큰 10000개를 사용해서 작업을 진행하였습니다.



NSMC 데이터 중 자주 사용된 상위 50개 데이터

딥러닝 모델 설계로는 두 개의 Dense 층은 64개의 유닛을 가지고 활성화 함수로는 relu[6]를 사용했으며, 마지막 층은 sigmoid[7] 활성화 함수를 사용해서 긍정의 리뷰일 확률을 출력합니다. 손실 함수로는 binary_crossentropy[8]를 사용했고 RMS Prop[9] 옵티마이저를 통해서 경사 하강법을 진행했습니다.



감정분석 딥러닝 모델

모델 설계과정에서 딥러닝 모델 설계를 더욱 간편하게 도와주는 keras 라이브러리를 사용하여 설계하였습니다.

---

6) relu : 활성화 함수중 하나로 f(x)=max(0,x)으로 정의된다.
7) sigmoid : 활성화 함수중 하나로 σ(x)=11+e−x 으로 정의된다.
8) binary_crossentropy : 2진 분류 모델에서 사영되는 손실함수
9) RMSProp : 경사 하강법 중 한 종류. 과거의 모든 기울기를 균일하게 더하지 않고 새로운 기울기의 정보만 반영하도록 해서 학습률이 크게 떨어져 0에 가까워지는 것을 방지하는 방법이다.

3.1.3 본 과제를 수행함에 있어 활용된 공학도구, 기술 및 장비

기획한 프로젝트의 특성상 모바일 게임 어플리케이션의 리뷰를 분석하는 웹 서비스 및 소프트웨어 프로그램이기 때문에 과제 수행 과정에서 특별히 활용된 공학도구, 기술 및 장비가 없습니다.

3.2 설계물에 대한 분석 및 보완

3.2.1 작동원리상에서 나타난 문제점 분석 및 보완

약 24만개의 리뷰로 단어들 간의 유사도를 계산하여 약 3,300개의 단어들에 대한 가중치 행렬까지 구현 후 3300개의 단어들로 24만개의 리뷰를 한 번에 내적하는 과정에서 과부화가 발생함을 확인하였습니다. 때문에 이를 해결하기 위해 3,300개의 단어들을 모두 사용하는 것이 아니라 저희가 사용할 카테고리를 분류하였고 24만개의 리뷰에서 1개의 게임 어플 리뷰를 분석하는 평균 만개정도로 조정하여 빠른 결과를 얻을 수 있었습니다.

3.2.2 구조도상에서 나타난 문제점 분석 및 보완[10)]

감정분석 딥러닝 과정에서 게임 리뷰 지도학습을 위한 Train Data Set이 필요하였는데, 본 프로젝트 기간 내에 만족할 수 있는 Set 구축은 불가능하다고 판단되어 GitHub에 있는 10만개의 Train Data Set, 5만개의 Test Data Set을 채택하였습니다.

3.2.3 주요 기능상에서 나타난 문제점 분석 및 보완

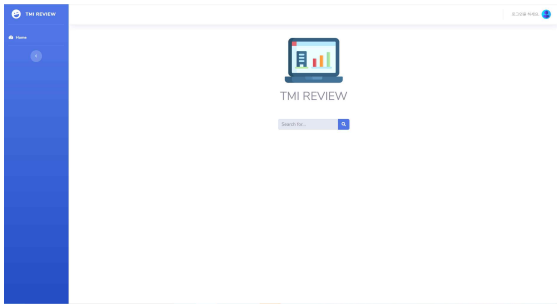
단어 유사도의 정확성에 대한 문제점이 가장 큰 중점이었습니다. 리뷰 유사도 분석 시 특정 카테고리 '아이템'으로 쏠리는 경향을 확인 하였습니다. 이를 해결하기 위해 '아이템' 카테고리를 삭제 한 뒤 시도하였고, 다른 특정 카테고리로 또 쏠리는 현상이 발생했습니다. 이에 리뷰 분석 스코어에 임계값을 두어 의미 없는 리뷰들의 카테고리화를 막아 주었습니다. 또한 5개의 카테고리에서 36개의 카테고리로 카테고리를 세분화 하여 더 세밀한 리뷰의 분산을 유도하였습니다. 또 하나의 리뷰가 하나의 카테고리만 보유하는 것이 아니라, 임계값을 넘은 카테고리 스코어는 최대 3개 까지 보유 할 수 있도록 하여 정확성을 높였습니다.

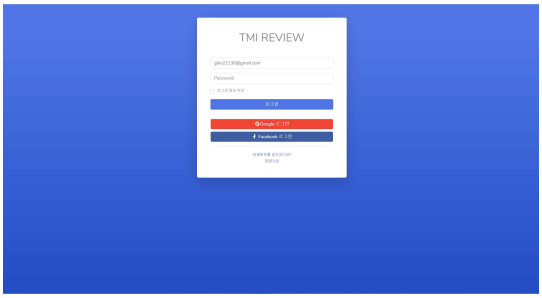
---

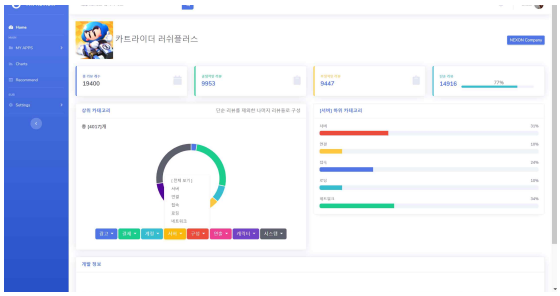
10) 참고 GitHub : https://github.com/e9t/nsmc/

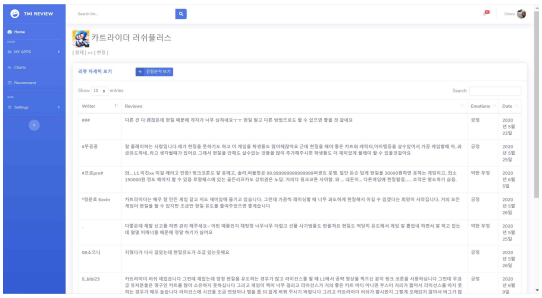
4. 제작

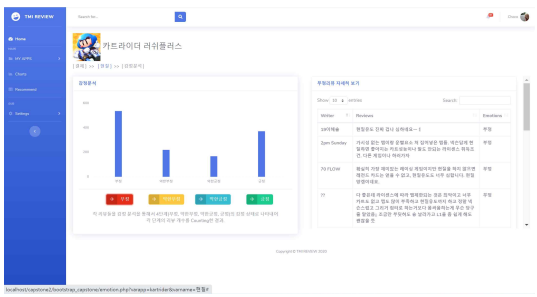
4.1 완성품 제작 결과 (사진)

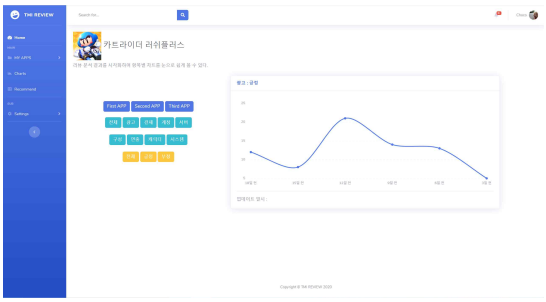












4.2 완성품 설명

로그인을 하게 되면 회원이 미리 설정한 첫 번째 관심 앱에 대한 분석이 출력이 되며 회원은 총 리뷰의 개수, 긍정적인, 부정적인, 기타 리뷰의 개수를 확인 할 수 있습니다.
화면 중앙에는 8개의 상위 카테고리의 비율을 Pie-Graph로 확인 할 수 있으며 광고를 제외한 각 카테고리의 5개 하위 카테고리에 대한 정보를 확인 할 수 있습니다.
사용자가 열람하기 원하는 하위 카테고리를 클릭하면 해당 카테고리로 분류가 된 리뷰들을 확인 할 수 있는 페이지로 이동하게 되고, 상단에 감정분석 보기 버튼을 통하여 감정 분석을 확인 할 수 있는 페이지로 이동하게 되어 4단계 (긍정, 약한 긍정, 약한 부정, 부정)으로 분류된 리뷰들의 비율을 확인 할 수도 있고 직접적으로 리뷰들을 확인 할 수 있습니다.
카테고리들의 변동 사항을 카테고리, 긍정-부정에 따른 날짜별 변동 사항을 확인 할 수 있습니다.

4.3 작품 제작 과정 정리

내용	담당자	시작 날짜	완료 날짜	진행도
크롤링				
수집해올 항목들 선정	팀원 전체	2020.04.03	2020.04.05	100
scroll down function , '더보기' click function 코드 구현	심재경	2020.04.03	2020.04.05	100
DataFrame 생성	심재경	2020.04.07	2020.04.08	100
app_url 추출 및 관련성순 클릭 코드 구현	심재경	2020.04.07	2020.04.08	100
app 개수 , review 개수 설정 코드 구현	심재경	2020.04.10	2020.04.13	100
생성된 DataFrame csv파일로 저장	심재경	2020.04.10	2020.04.13	100
발생하는 error들에 대하여 보수작업	심재경	2020.04.14	2020.06.01	100
Train Data로 쓸 인기게임 1위~200위 어플 리뷰 크롤링	신현욱	2020.05.19	2020.05.19	100
24만개 리뷰 하나의 csv 파일로 병합	신현욱	2020.05.19	2020.05.19	100
단어 유사도 분석				
수집 된 리뷰 불용어 제거	심재경	2020.04.14	2020.04.21	100
불용어 제거된 리뷰들로부터 토큰화	심재경	2020.04.14	2020.04.21	100
Word2Vec 모델 적용	김지수,신현욱	2020.04.21	2020.04.21	100
단어행렬 구축	김지수	2020.04.22	2020.04.29	100
가중치 행렬 구축	김지수	2020.04.29	2020.05.01	100
TDM 구축	김지수	2020.05.01	2020.05.10	100
Word2Vec으로부터 나온 단어 사전 효율적으로 배열 정리	김지수, 신현욱	2020.05.10	2020.05.11	100
가중치 행렬에서 원하는 카테고리 번호 추출	신현욱	2020.05.10	2020.05.11	100
TDM 내적과정 행렬 트랜스폼 구현	김지수	2020.05.10	2020.05.12	100
카테고리 5개에서 적절한 의미 찾아 세분화하기	김지수,신현욱	2020.05.12	2020.05.24	100
특정 카테고리로 쏠리는 현상 방지 적절 한 임계 값 찾기 – 정확도 향상	김지수,신현욱	2020.05.12	2020.05.24	100
카테고리 분산을 위한 세밀화	신현욱	2020.05.24	2020.06.01	100
각 리뷰 당 3개의 카테고리를 가지도록 프레임 설계	김지수	2020.06.05	2020.06.05	100
분석된 리뷰 데이터 프레임 설계 및 저장	김지수	2020.06.05	2020.06.05	100
감정 분석				
수집 한 리뷰 불용어 제거	심재경	2020.04.21	2020.04.22	100
불용어 제거 된 리뷰들 형태소 태깅 및 토근화	심재경	2020.04.21	2020.04.22	100
형태소 태깅 및 토큰화 된 데이터를 새로운 json 파일로 저장	심재경	2020.04.22	2020.04.23	100
각 토큰 별로 빈도수 체크 후 자주 사용되는 토큰 10000개 추출	심재경	2020.04.25	2020.04.28	100
추출한 토큰을 사용해서 리뷰 데이터 벡터화	심재경	2020.04.28	2020.04.28	100
딥러닝 모델 설계	심재경	2020.04.30	2020.05.05	100
준비한 데이터를 딥러닝 모델을 이용하여 학습	심재경	2020.05.05	2020.05.12	100
결과값으로 나올 score를 바탕으로 4단계 감정으로 나누기	심재경	2020.05.12	2020.05.20	100
각 App별 수집한 리뷰들을 이용하여 감정분석 결과 예측하기	심재경	2020.05.20	2020.06.02	100
웹 UI 구현				
전체 페이지 구성 설계 및 기획	팀원 전체	2020.05.30	2020.06.05	100
시작 화면 페이지 구현	신현욱	2020.06.08	2020.06.08	100
로그인 화면 페이지 구현	신현욱	2020.06.03	2020.06.03	100
회원 가입 페이지 구현	신현욱	2020.06.03	2020.06.03	100
메인 화면 페이지 설계 및 구현				
- 좌측 슬라이드 바 메뉴 기능	신현욱	2020.06.04.	2020.06.04	100
- 상단 검색 창 및 로그인 정보	신현욱	2020.06.04	2020.06.04	100
- 리뷰 분석 요약 정보 기능	신현욱	2020.06.04	2020.06.04	100
- 상위 카테고리 분석 Pie-Chart	신현욱	2020.06.04	2020.06.05	100
- 하위 카테고리 분석 Bar-Graph	신현욱	2020.06.04	2020.06.05	100
- 화면 내 FrameChange 기능	신현욱	2020.06.04	2020.06.05	100
마이 앱 편집 페이지 구현	신현욱	2020.06.07	2020.06.07	100
리뷰 자세히 보기 페이지 구현- 반응형 테이블	신현욱	2020.06.06	2020.06.06	100
감정 분석 보기 페이지 구현				
- 감정 분석 Bar-Graph	신현욱	2020.06.07	2020.06.07	100
- 리뷰 자세히 보기 FrameChange 기능 추가	신현욱	2020.06.07	2020.06.07	100
차트 시각화 페이지 구현 - Graph 활용	신현욱	2020.06.08	2020.06.08	100
계정 설정 페이지 구현	신현욱	2020.06.08	2020.06.08	100

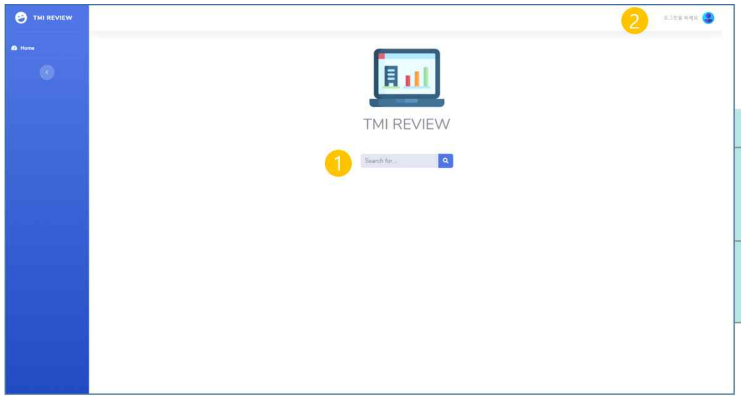
웹 DB 연동				
회원 정보 테이블과 로그인 페이지 연동	심재경, 김지수	2020.06.02	2020.06.02	100
리뷰 DB 연동	김지수	2020.06.02	2020.06.04	100
리뷰 DB – 긍, 부정 총 카운팅 개수 연동	김지수	2020.06.02	2020.06.04	100
리뷰 DB – js 파일에서 pie-chart에서 db 속 정보 출력	김지수	2020.06.02	2020.06.03	100
리뷰 DB – 상위 카테고리 버튼 클릭 시 하위 카테고리 출력	김지수	2020.06.02	2020.06.03	100
리뷰 DB – js 파일 bar- graph에서 db 속 정보 출력	김지수	2020.06.03	2020.06.03	100
리뷰 DB – 분석 완료 된 총 리뷰 카운팅	김지수	2020.06.03	2020.06.04	100
리뷰 DB – 카테고리 화 된 리뷰 출력	심재경	2020.06.04	2020.06.05	100
리뷰 DB – 감정 분석 보기 버튼	김지수	2020.06.07	2020.06.07	100
리뷰 DB – 감중 분석 보기 페이지에서 하위 카테고리로 페이지 연결 버튼	김지수	2020.06.07	2020.06.07	100
리뷰 DB – js 파일 bar- graph에서 db 속 감정 분석 정보 출력	김지수	2020.06.07	2020.06.07	100
리뷰 DB – 감정분석 버튼 클릭 시 해당 감정 리뷰 출력	김지수	2020.06.07	2020.06.07	100
리뷰 DB – js 파일 db 속 날짜별 변동 사항 정보 출력	김지수	2020.06.07	2020.06.07	100
회원 DB – 차트 페이지 속 회원의 관심 앱 버튼	김지수	2020.06.08	2020.06.08	100
리뷰 DB – 상위카테고리 버튼 연동	김지수	2020.06.08	2020.06.08	100
리뷰 DB – 긍정, 부정 카테고리 버튼 연동 (약한긍정, 약한부정 포함)	김지수	2020.06.08	2020.06.08	100
리뷰 DB – 회원 관심앱 설정	김지수	2020.06.09	2020.06.09	100

4.4 작품의 특징 및 종합설계 수행 결론

단어유사도 분석을 통해 플레이스토어에 있는 게임의 리뷰들을 카테고리별 스코어를 이용하여 카테고리화 시켜주어 방대한 양의 리뷰들을 분류하여 확인할 수 있게끔 도와줍니다. 이는 리뷰에서 원하는 정보를 빠르게 얻는데 용이하게 사용될 수 있다고 판단되며, 기존 리뷰 분석 플랫폼에서는 사용자의 성별, 사용시간, 설치한 앱의 수 등의 사용자 위주로만 분석이 중점이 되어 강조 되었고, 리뷰에 대한 분석은 자세한 분석이 아닌 2단계의 감정분석으로만 분류, 또한 분석이 이루어져도 영어가 아닌 한국어로 이루어지는 사이트는 소수였습니다. 때문에 이러한 리뷰의 카테고리 분류화는 사용자의 니즈에 더 다가갈 수 있게 도와주며 빠른 피드백이 가능하게끔 해줍니다.

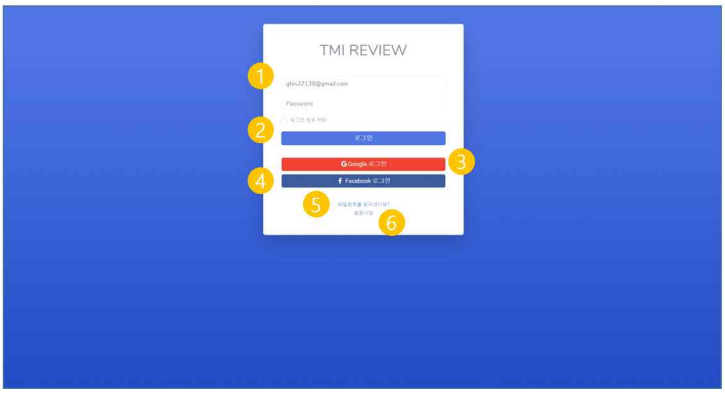
4.5 완성품의 사용 매뉴얼

- 시작 화면



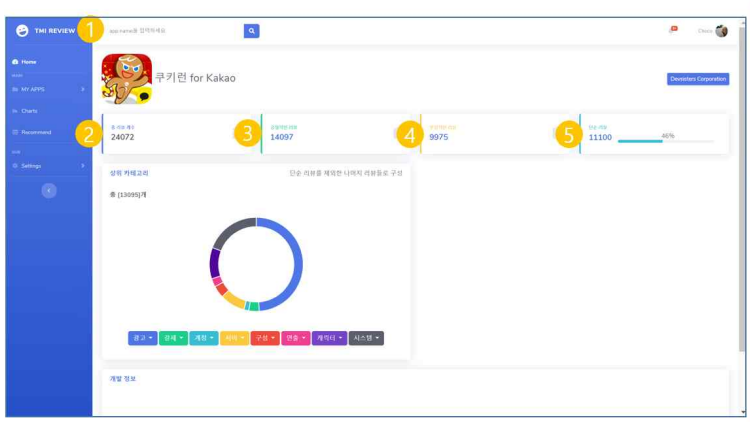
No	설 명
1	- 사용자가 열람하기 원하는 앱을 입력 받을 수 있습니다. - 앱을 입력 받으면 분석화면으로 이동 하게 됩니다.
2	- 로그인을 할 수 있는 화면으로 이동하게 됩니다.

- 로그인 화면

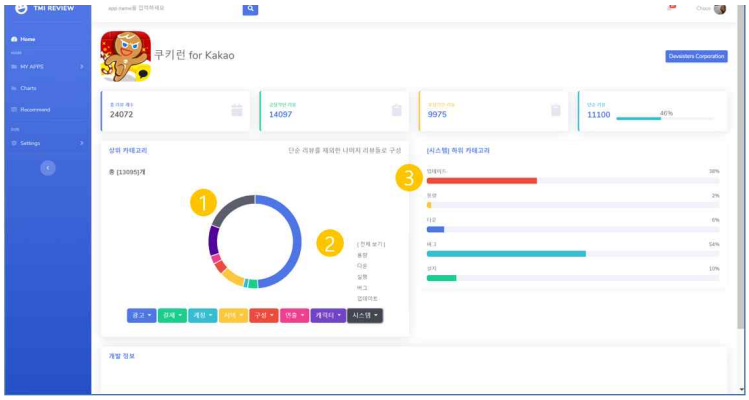


No	설 명
1	- 사용자의 아이디와 비밀번호를 입력 받습니다.
2	- 로그인을 하게 되면 사용자가 설정한 첫 번째 관심앱 분석 페이지로 이동하게 됩니다.
3	- Google 아이디로 로그인이 가능합니다.
4	- Facebook 아이디로 로그인이 가능합니다.
5	- 비밀번호를 잊었을 시 비밀번호 변경을 도와줍니다.
6	- 새로이 가입하려는 회원에게 회원가입을 도와줍니다.

- 메인 분석 화면

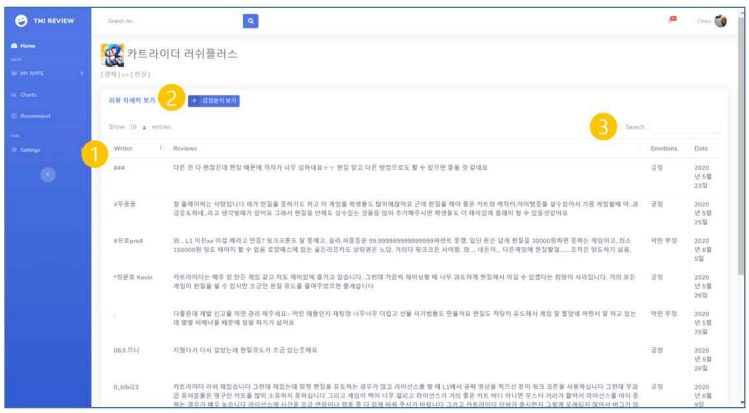


No	설 명
1	- 사용자가 열람하기 원하는 게임을 검색 받을 수 있습니다.
2	- 해당 앱의 총 분석된 총 리뷰의 개수를 출력 시켜 줍니다.
3	- 총 리뷰 중에서 긍정, 약한 긍정으로 분류된 리뷰들의 개수를 보여 줍니다.
4	- 총 리뷰 중에서 부정, 약한 부정으로 분류된 리뷰들의 개수를 보여 줍니다.
5	- 총 리뷰 중에서 의미 없는 리뷰들인 단순 리뷰들의 개수와 그 퍼센트를 보여줍니다.



No	설 명
1	- 8개의 상위 카테고리의 비율을 Pie Graph로 확인 할 수 있습니다.
2	- 상위 카테고리를 클릭하면 해당 카테고리의 하위 카테고리로 분류 된 리뷰들을 열람할 수 있는 페이지로 이동합니다.
3	- 상위 카테고리 속 하위 카테고리의 비율을 그래프로 확인 할 수 있습니다.

- 리뷰 상세보기

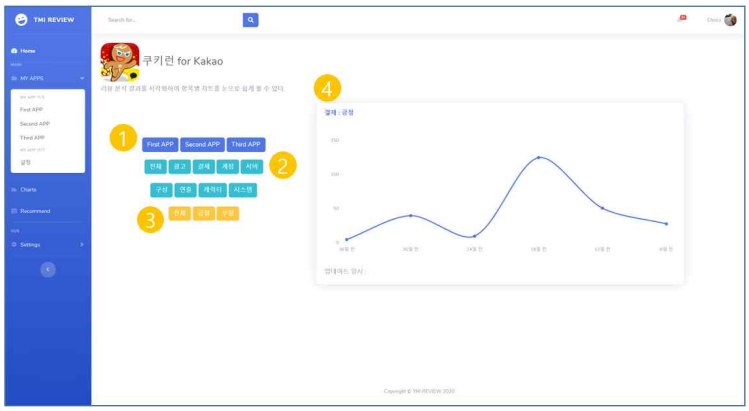


No	설 명
1	- 카테고리화 된 리뷰들을 열람 할 수 있습니다. 작성자, 리뷰내용, 감정, 날짜에 따라 분류되어 있으며 해당 컬럼명 클릭 시 정렬됩니다.
2	- 해당 버튼을 클릭하면 감정 분석 페이지로 이동하게 됩니다.
3	- 원하는 키워드나, 내용을 입력하면 해당 단어가 포함된 리뷰들만 볼 수 있습니다.

- 감정 분석

No	설 명
1	- 감정 분석의 결과를 그래프로 시각화 합니다. 해당 바에 커서를 올려 두면 해당 감정으로 분류 된 리뷰들의 개수를 확인 할 수 있습니다.
2	- 4단계의 감정 분석들의 버튼을 클릭하면 화면 우측에 리뷰들을 출력해줍니다.
3	- 사용자가 열람하기 원하는 감정의 리뷰들만 출력해 줍니다. 테이블의 기능은 전 카테고리 리뷰 화면과 동일합니다.

- 차트 페이지



No	설 명
1	- 사용자가 설정한 관심앱 3가지를 확인 할 수 있습니다.
2	- 전체 카테고리, 각 카테고리를 클릭 시 오른쪽에 변동사항을 날짜별 차트로 시각화 시켜줍니다.
3	- 전체, 긍정, 부정 버튼을 클릭시 상단에서 설정한 카테고리 설정에서 긍, 부정을 나누어 출력시켜 줍니다.

5. 시험 (시험 결과 기술)

5.1 최종 결과물에 대한 시험 결과

단어유사도에 대한 테스트 방법 및 결과

테스트 기능	테스트 제목	테스트 방법/절차	테스트 결과
단어 유사도	단어 유사도 스코어	가중치행렬과 TDM을 내적하여 리뷰별로 스코어가 산출이 되는가	리뷰별 스코어 산출 완료
단어 유사도	카테고리 분배	5개의 상위카테고리에서 36개의 하위카테고리로 증가 시켰을 때 리뷰 분배도 측정	카테고리의 고른 분배 확인
단어 유사도	임계값 설정	임계값을 0부터 0.05씩 증가 시키며 편향된 데이터(단순 리뷰)의 필터링 측정	0.2를 임계값으로 설정하여 단순 리뷰 최대 분류
단어 유사도	카테고리 변경	상위 카테고리와 더욱 연관성 있는 하위 카테고리들로 변경시켜 연관성 측정	네트워크, 모션 도입, 연출 제거
단어 유사도	유사도 출력 결과 확인	1. 상위 카테고리 속에 잘 포함되어있는지 2. 하위 카테고리에 속한 리뷰가 잘 출력이 되는지	유사도 상위 카테고리, 하위카테고리에 대한 리뷰가 정확히 잘 출력됨

감정분석에 대한 테스트 방법 및 결과

테스트 기능	테스트 제목	테스트 방법/절차	테스트 결과
텍스트 전처리	불용어 사전을 이용한 텍스트 제거	불용어 사전에 포함되어 있는 텍스트들이 잘 제거 되있는가	불용어 사전에 추가한 모든 텍스트가 제거됨
텍스트 전처리	형태소별 품사 태깅	각 형태소별로 품사가 잘 태깅되있는가	모든 형태소별로 적합한 품사가 잘 태깅됨
텍스트 전처리	형태소별 토큰화	각 형태소별로 토큰화를 통해 잘 나누어져있는지	모든 형태소별로 토큰화가 잘 됨
전처리된 텍스트 저장	전처리된 텍스트 json파일 저장	전처리된 텍스트가 빠짐없이 json 파일로 저장이 되는가	모든 전처리된 텍스트가 올바르게 저장됨
토큰 빈도수 확인	토큰별로 자주 사용된 상위 10000개 토큰 출력	자주 사용된 토큰 10000개를 출력해서 실제 데이터 파일과 일치하는지	데이터 파일에 알맞은 상위 10000개 토큰 출력됨
감정분석 딥러닝	딥러닝 모델 설계	설계된 딥러닝 모델의 test 데이터로 결과값이 90%이상 일치하는지 확인	95%일치
감정분석 딥러닝	4단계 감정분석	결과값으로 나오는 score를 대상으로 4개의 기준을 나눠 올바르게 감정분석이 되는지 확인	4단계의 단계별로 올바른 감정분석 출력
감정분석 결과 출력 페이지	감정분석 결과 데이터 연동 확인	감정분석 결과를 저장한 database가 웹페이지에서 잘 연동되는지 확인	웹페이지와 100% 연동됨
감정분석 결과 출력 페이지	감정분석 결과 데이터 출력 확인	1. 감정분석 결과가 저장된 database가 실제 웹페이지에서 정확하게 출력되는지 2. 4단계별로 분류된 리뷰의 개수가 그래프로 잘 표현이 되는지	감정분석 결과 및 그래프가 database와 동일하게 출력됨

6. 평가

6.1 작품의 완성도 및 기능 평가

본 과제의 완성을 위해 설정한 목표	완성도
리뷰 데이터의 대한 올바른 수집	100%
리뷰 데이터의 대한 실용적인 전처리 과정	95%
유사도 측정을 통한 카테고리별 분류 정확도	90%
4단계 감정분석 결과 정확도	90%
database관리 및 웹 UI연동	100%

해당 프로젝트에서 핵심적인 부분은 리뷰 데이터의 실용적인 전처리 과정을 통해 가공된 리뷰 데이터로 유사도 측정을 통한 카테고리별 분류 및 4단계 감정분석입니다. 게임 리뷰 전처리 과정에 적합한 불용어 사전 구축 및 게임 리뷰에 적합한 카테고리 설정으로 인해 유사도 측정 정확도를 증가시키고 데이터 토큰화와 적합한 딥러닝 모델구현으로 인해 감정분석의 정확도를 증가시켜서 완성도를 증가시키는 방향으로 구현하였습니다.

6.2 기대효과 및 영향

6.2.1 기대효과

1) 효율적인 소비자와 개발자간의 상호작용

현재 구글 플레이스토어에 인기게임 같은 경우 리뷰의 개수가 작게는 수 만개에서 많게는 수십만 개까지 존재하는데 이번 프로젝트를 통해서 수십만 개 리뷰의 카테고리별 분류를 통하여 개발자는 소비자의 요구에 대해 더욱 편리하게 알게 되고 소비자도 개인의 의견이 개발자에게 더욱 편리하게 전달되는 것을 인지하게 됨으로 더욱 개발자에게 도움이 될 만한 실용적인 리뷰들을 많이 작성하게 됨으로 인해 상호간의 긍정적인 상호작용이 발생한다.

2) 주기적인 업데이트

개발자들은 소비자로부터 해당 게임에 대한 문제점에 대한 리뷰에 대해 항상 신경을 쓰고 해당 문제점들에 대한 보완 및 보수가 된 업데이트 버전을 새로 출시해야한다. 이 때 이 리뷰 분석 프로그램을 활용하면 더욱 간편하고 신속한 리뷰 분석이 가능해지고 이로부터 단점에 관한 리뷰에 대한 빠른 피드백이 담긴 업데이트를 주기적으로 제공이 가능해진다.

3) 다양한 분야로의 확장

현대 사회에서는 게임 외에도 모든 분야의 App또는 상품에 리뷰가 존재하고 시간이 지날수록 리뷰에 대한 중요성이 증가하고 있다. 따라서 게임 App 외에 다양한 영역에서도 효과적인 카테고리만 설정한다면 긍정적인 효과를 불러올 수 있다고 생각합니다.

6.2.2 해결방안의 긍정적 및 부정적인 공학적 영향

TMI review 프로젝트를 통하여 리뷰의 빠른 피드백이 요구되는 시대에서 소비자들의 니즈를 충족시켜줌으로써 개발자들은 수익의 증가와 게임의 완성도를 증가시켜줄 수 있고 소비자들은 단순한 소비자가 아닌 해당 게임의 주체로써 적극적인 의견을 표출할 수 있게 되는 긍정적인 영향들이 있습니다.

6.3 작품제작 후기

이 사회에서 모바일 게임 리뷰의 영향이 생각보다 많은 사람들에게 작용하고 있다는 것을 발견했습니다. 모바일의 편리성을 이용하여 수십만 개의 리뷰들이 작성되고 대부분의 소비자들은 리뷰를 관심 깊게 확인한다는 것을 알게 되었습니다. 그리고 소비자들의 성향분석만으로는 대중적인 만족도를 증가시킬 수 없고 리뷰 자체의 빠른 피드백이 필요하다고 생각되어 개발자들이 빠른 피드백을 하기 위하여 먼저 리뷰에 대한 신속하고 편리한 접근을 가능하게 해주는 방법을 찾아보자는 생각에 이 프로젝트를 진행하게 되었고 효과적인 방법을 찾았다고 생각했습니다.

이번 프로젝트를 진행하면서 주로 이미지와 오디오로만 접하던 인공지능 분야에서 새로운 텍스트분야 인공지능에 대하여 접하게 되면서 어려움이 많았습니다. 아직 이미지와 오디오에 비해서는 조사할 수 있는 자료와 기술의 수도 부족하고 상대적으로 어려운 기술들을 접하게 되면서 프로젝트 수행에 대하여 어려움이 있었지만 새로운 다양한 경험과 기술을 접하게 되면서 흥미도 생겼고 기존의 알고 있던 지식과 새로 접하게 된 기술들을 융합해서 효과적인 프로젝트를 진행하게 되었습니다.

6.4 팀 개요 및 역할분담

팀원	역할
심재경	크롤링을 통한 데이터 수집 및 전처리 , 감정분석 모델 설계 및 구현
김지수	카테고리 선정 및 유사도 측정을 통한 분류작업 , DB 설계 및 연동
신현욱	카테고리 선정 및 유사도 측정을 통한 분류작업 , 웹/UI 설계 및 구현

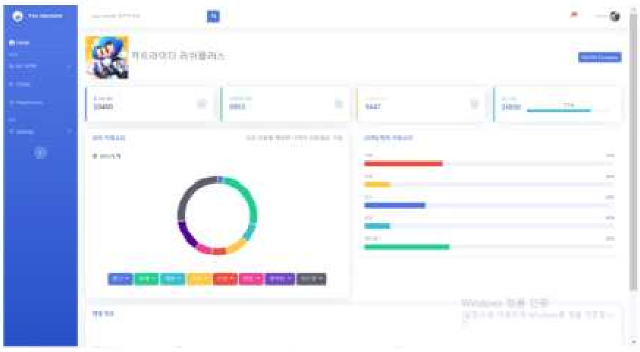
6.5 참고문헌

- 파이썬 자연어 처리의 이론과 실제 : 효율적인 자연어 처리를 위한 머신 러닝과 딥러닝 구현하기, 잘라지 트하나키; 이승준, 서울 에이콘 (2018)
- 영화 리뷰 감성분석을 위한 텍스트 마이닝 기반 감성 분류기 구축, 김유영, 송민, 한국지능정보시스템학회 (2016)
- Lee, K.Y "Development of An Automatic Classification System for Game Reviews Based on Word Embedding and Vector Similarity" (2019)
- https://sensortower.com/?gclid=CjwKCAjwltH3BRB6EiwAhj0IUGPJepcJ5VyoCpIFcyxegijCducVqt-V0JvDWtltZLtwew90dQuZvxoCegMQAvD_BwE&locale=ko (기존 앱분석시장 분석)
- https://www.wiseapp.co.kr/app/detail/7daadf25b56cfbd10c35f1ceede865ac/?tabType=usage&dateType=1&searchDate=2020-06-15 (기존 앱분석시장 분석)
- https://selenium-python.readthedocs.io/ (데이터 크롤링 문법 참고)
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (데이터 크롤링 문법 참고)
- https://ratsgo.github.io/natural%20language%20processing/2017/03/08/word2vec/ (리뷰 분석 참고)
- https://github.com/corazzon/petitionWrangling/ (리뷰 분석 참고 - 국민청원 분석)
- https://www.youtube.com/watch?v=16f5R5TPnMY&t=776s/ (리뷰 분석 참고)
- http://radimrehurek.com/gensim/models/keyedvectors.html#gensim.models.keyedvectors.Word2VecKeyedVectors/ (word2vec parameter 학습)
- https://blog.naver.com/bigdata-pro/221870004360/ (skip-gram,cbow 학습)
- http://blog.naver.com/PostView.nhn?blogId=dalsapcho&logNo=20132553174/ (확률밀도 함수 참고)
- https://github.com/jwnz/document-term-matrix/ (TDM 방법론 참고)
- https://blog.naver.com/myincizor/221643794025/ (TDM 참고)
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html/ (CounterVectorizer parameter 학습)
- https://blog.naver.com/danelee2601/221593266969/ (CounterVectorizer 참고)
- https://cyc1am3n.github.io/2018/11/10/classifying_korean_movie_review.html/ (감정분석 데이터 전처리 방법)
- http://dilab.kunsan.ac.kr/pub/hclt18c.pdf/ (감정분석 다양한 딥러닝 모델 참고)
- http://www.gisdeveloper.co.kr/?p=7663/ (경사하강법 파이썬 적용 참고)
- https://wikidocs.net/28146/ (활성화함수 파이썬 적용 참고)
- http://blog.naver.com/PostView.nhn?blogId=wideeyed&logNo=221017173808/ (활성화함수 파이썬 적용 참고)
- https://somjang.tistory.com/entry/Keras/ (여러가지 감정분석 모델 참고)
- https://getbootstrap.com/ (부트스트랩)
- https://startbootstrap.com/themes/ (부트스트랩 테마 이용)
- https://codepen.io/ (웹 오픈 소스 이용)
- http://www.happycgi.com/ (웹 오픈 소스 이용)
- https://www.graphberry.com/ (UI Kits 이용)

[첨부 1] 작품 주요 사진 첨부

Counter([illegible] 이벤트': 4411, '네트워크': 4044, '그래픽': 3883, '접속': 2977, '설치': 2955, '계정': 2749, '결제': 2744, '스테이지': 2594, '서버': 2458, '구매': 1824, '소리': 1802, '다운': 1766, '환불': 1752, '로그인': 1662, '연결': 1634, '로딩': 1630, '스킨': 1356, '스킬': 1250, '난이도': 974, '영웅': 925, '연동': 844, '퀘스트': 754, '충전': 648, '디자인': 570, '가입': 536, '배경': 531, '용량': 429, '아이디': 423, '구성': 375, '모션': 301})

실제 파이썬에서 24만개의 리뷰 분류 카운팅 결과



메인 분석 화면

감정 분석 화면

날짜별 분석 화면

리뷰 상세 보기 화면