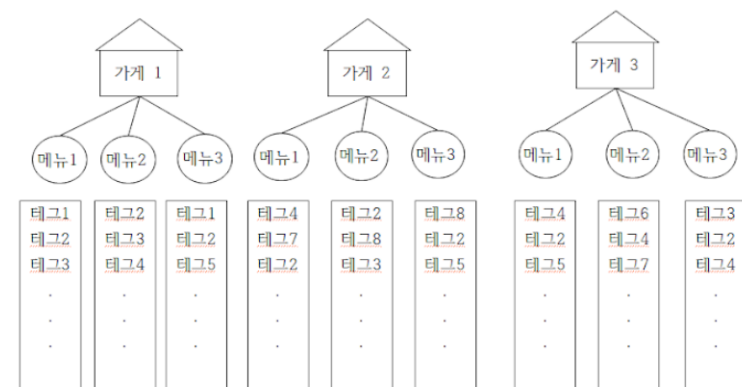
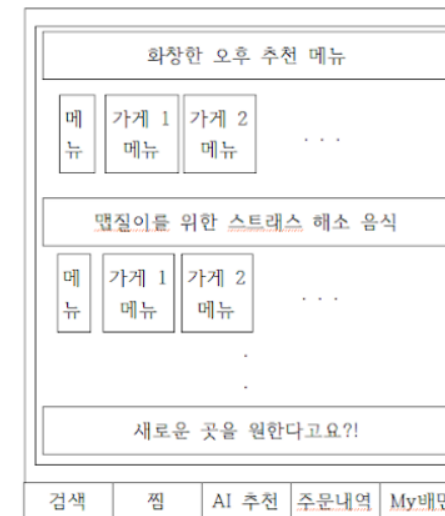


UI 예시



리뷰들을 통해서 태그 값 추출 (자연어 처리)

개인화 추천 (PR) = 유저프로파일링 (P) + 아이템 매칭 (M) + 아이템 랭킹 (R) + 아이템 필터링 (F)

1. 가게 태그 생성

- 가게의 태그: 해당 가게의 베스트 메뉴의 태그로 구성된다.
- 가게 메뉴의 태그: 특정 길이 이상의 리뷰를 사용, 음식의 특징을 잘 나타내는 단어를 태그로 구성한다.

<리뷰 분석에 사용할 알고리즘>

BPE(Byte Pair Encoding): 서브워드(sub-word)단위로 분리하는 알고리즘

<태그 생성 과정>

사용자가 메뉴를 주문

-> 메뉴의 태그가 사용자에게 축적

-> 태그가 중복될수록 가중치가 증가

-> 사용자의 선호 태그 랭킹화

2. 이웃추천, 가게추천, 패턴추천 방식

[이웃추천]

비슷한 태그를 갖는 사용자 군집화 (여기서 사용자끼리 태그의 랭킹이 다를 텐데 랭킹이 비슷한 사람들끼리 2차 군집화?)

비슷한 태그를 갖는 이웃의 사용 빈도가 높은 가게 추천(if 사용 빈도 높은 가게가 없다면 해당 태그를 갖는 음식점을 추천)

[가게 추천]

- 사용자의 태그와 같은/비슷한 태그를 가진 가게를 추천

<사용할 알고리즘>

CBF(contents based filtering)

[패턴 추천]

- 사용자의 구매 패턴 분석 -> 특정 패턴을 보이는 메뉴/가게 추천

- 날씨, 시간대, 계절에 따른 메뉴/가게 패턴 분석

- 특정 주기를 가지는 패턴으로 개인화 추천 진행

<사용할 알고리즘>

둘 중에 더 결과가 좋은 모델 사용:

다중퍼셉트론(MLP) 모형 VS 순환신경망(RNN) 모형

반복 구매제품의 재구매시기 예측을 위한 두 모형의 성능비교

3. '매니아'와 '모험가'의 성향 분류

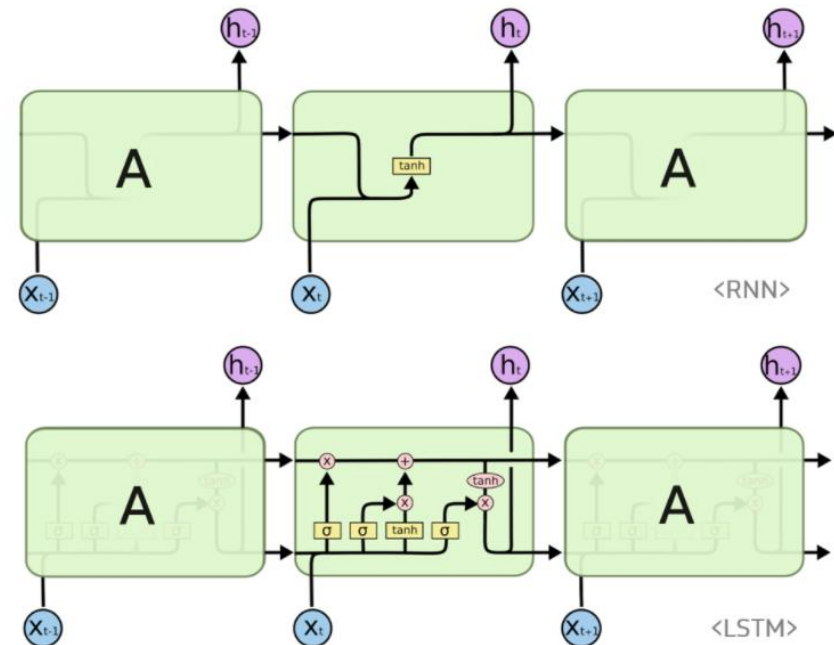
- 매니아: 똑같은/비슷한 음식을 계속/자주 먹는 사용자
- '모험가': 과거와 다른/새로운 음식을 먹는 사용자

<사용할 알고리즘>

LSTM(Long short term memory): RNN의 "vanishing gradient problem"을 극복하기 위해 고안된 알고리즘

RNN의 hidden state에 cell-state를 추가한 구조

- 사용자가 주로 단골집에서 자주 시키는지
 - 다양한 가게의 다양한 메뉴에 도전을 즐기는지
 - 과거의 데이터와 현재의 데이터에 가중치 차이를 두지 않는다.
-
- 사용자가 주문한 횟수, 즉 평균 주문 수를 계산 (전체)
 - 각 가게마다 가중치를 적용 (평균에서 얼마나 떨어져 있는지에 따라)
- ex) '매니아'
- 사용자가 A, B, C 가게에서 각각 10, 4, 1번 주문
- > 전체 평균 주문수 = 5번
 - > 평균에서 얼마나 어떻게 떨어져 있는지 따라 분류 (분산)
 - > A: +5 / B: -1 / C: -4 의 가중치를 둔다.



- ex) '모험가'

사용자가 A, B, C, D, E, F 가게에서 각각 1번씩 주문

-> 전체 평균 주문수 = 1번

-> 분산

-> A: 0 / B: 0 / C: 0 / D: 0 / E: 0 / F: 0

-> 가중치의 평균이 0에 가까움 = 모험가