[Deview 2021] 로그없이 영끌 키워드 추천

로그없이 영끌 키워드 추천

0. 알 수 있었던 점

- 라인 검색에서 제공하는 검색 키워드 추천하는 파이프라인 : 1. 실시간으로 주요 이슈 파악 → 2. 적절한 키워드 선정 → 3. 랭킹모델 (개인화 추천)
- 서비스 적용시 생기는 문제를 해결하는 방식 : 모델이 어떤 task를 수행할때 이를 수동적으로는 어떻게 진행하고 있는지 확인하고, 그 진행 방식을 모델이 어떻게 수행할지를 고려해보는 식으로 문제를 해결하였음
- 자동으로 키워드를 생성하는 방법과 자동으로 생성되는 키워드 이슈 해결 방법 : 자동으로 만들어진 키워드의 20%는 서비스에서 사용하기 어려운데 이를 해결하기 위해, 적절하지 않는 키워드는 필터링을 활용하여 제거함.
- ML이 아닌 다양한 알고리즘 방식을 활용하는 점

1. 키워드 추천 프로젝트 개요

일본 라인 앱에 재공중인 다양한 키워드 추천

새로운 문제들

- 모든 키워드가 수동으로 생산
- 뉴스 검색 키워드 위주로 소비
- 네이버에 비해서 작은 규모의 검색 로그

⇒ 따라서 키워드를 수동으로 생산하는 작업자나 검색 로그없이 키워드를 생산하고 싶음

새로운 검색 키워드 만들기

1. 24시간 수동관리 없이
2. 다양한 관심사에 대해
3. 검색을 유도할 만한 고품질의 키워드 생산

위 방식의 키워드 생산을 위한 모델 개요

- HyperCLOVA를 이용한 키워드 생산
    - 다양한 분야의 컨텐츠를 Hyper-scale Model의 입력으로 세팅하여, 고품질의 키워드를 자동생성.

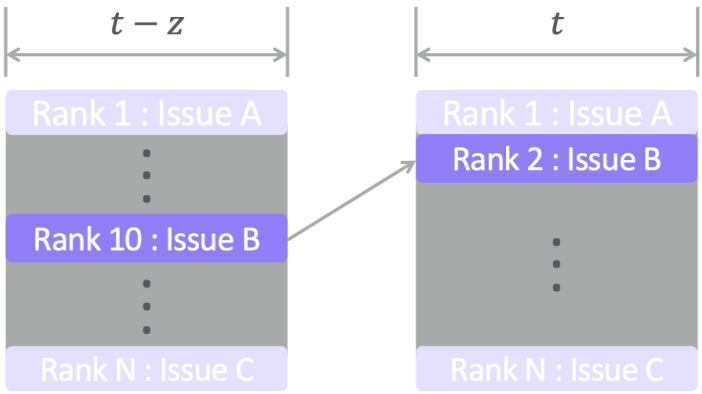
2. HyperCLOVA를 활용한 이슈 키워드 자동 생성

수동 편집키워드를 자동화 해보자

- “수동 편집키워드는 어떻게 생성되는가” 부터 파악 : 1. 실시간으로 주요 이슈 파악 → 2. 적절한 키워드 선정 → 3. 랭킹모델 (개인화 추천)

실시간 이슈 감지 모델 (1. 실시간으로 주요 이슈 파악에 해당됨)

- Latest Popular Model : 컨텐츠 소비자 관점에서, 사용자들이 최근에 많은 관심을 가지는 컨텐츠를 클릭로그를 사용하여 탐지
    - 단순히 클릭이 많은 컨텐츠를 사용하면 실시간 발생하는 이슈 탐지가 어려움. 따라서 클릭 절대량이 높으면서, 절대량의 랭크변화가 큰 컨텐츠를 감지.



- 현재 시점 t 기준 최근 n분 동안의 컨텐츠 v_j의 클릭 수 산출
- 산출된 클릭 수를 기준으로 각 컨텐츠의 순위를 $r_j(t)$로 표현
- $H(t)$: $r_j(t)$ 기준 t시점의 가장 인기 있는 상위 k개의 컨텐츠 set
- $H(t)$ 의 각 컨텐츠 v_j에 대한 최신 인기(latest popular) 점수 $lp_j(t)$를 계산

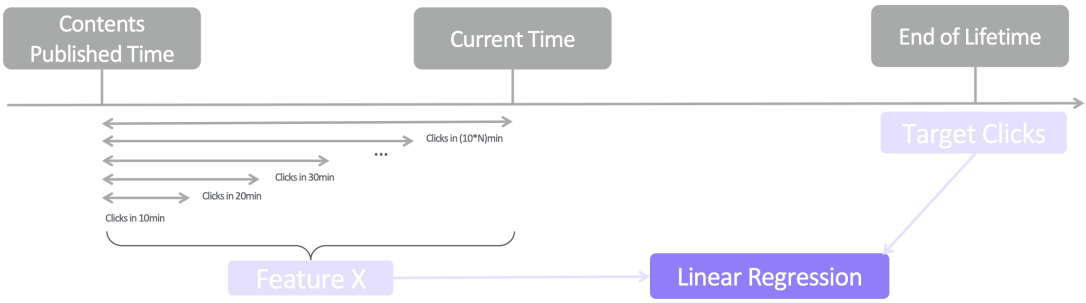
$$lp_j(t) = \alpha \cdot pop_j(t) + \beta \cdot raise_j(t),$$

$$pop_j(t) = 1 - \frac{rank\left(avg\left(r_j(t-n), r_j(t)\right), H(t-n,t)\right)}{|H(t-n,t)|},$$

$$raise_j(t) = 1 - \frac{rank\left(r_{j(t-n)} - r_{j(t)}, H(t-n,t)\right)}{|H(t-n,t)|},$$

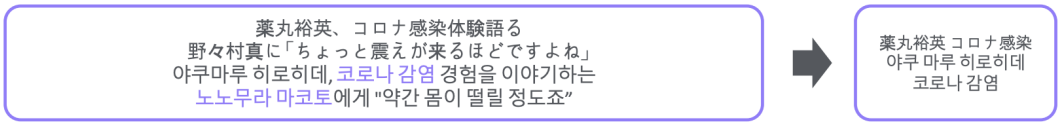
위 공식에서 $pop_j(t)$는 컨텐츠의 최근 순위가 얼마나 높은지 측정하는 스코어.
$raise_j(t)$는 컨텐츠의 최근 순위 변동을 측정하는 스코어.

- Cluster Model : 컨텐츠 생성자 관점에서, 다수의 컨텐츠 제공자들이 공통적으로 발행하는 이슈를 포함하는 컨텐츠를 탐지
  - 동일한 이슈를 다루는 컨텐츠를 클러스터링 기법으로 분류
  - 컨텐츠의 제목과 본문을 이용하여 TF-IDF vector 생성
  - vector 간의 유사도 기반으로 Hierarchical clustering 수행.
  - ⇒ 이를 통해 클러스터링 크기가 크면서 클릭이 많은 컨텐츠를 탐지
- Future Impact Model : 현재 컨테츠에 대한 클릭을 기반으로, 미래의 클릭 수를 예측해서 이슈를 탐지
  - 컨텐츠 생성 시점 이후 매 (10*N)분 동안의 클릭 수를 조합해서 feature로 사용.
  - 컨텐츠 lifetime 동안의 전체 click 수 예측

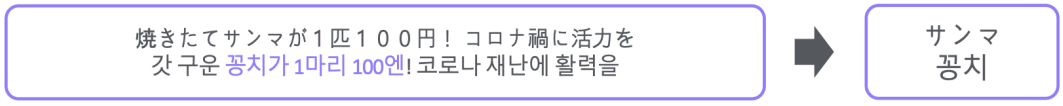


키워드 생성 모델(2. 적절한 키워드 선정에 해당됨)

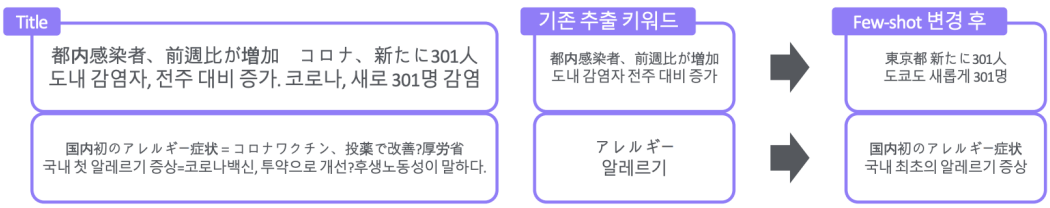
- HyperCLOVA 모델 이용
- 자동 생성 품질 고도화 이유 : 키워드 자동생성에서 발생하는 문제점 존재
  1. 정보 왜곡 (부정확한 키워드 생성)



  2. 정보량이 적은 키워드 생성



- Few-shot Example Tuning : 컨텐츠의 카테고리별 Few-shot Example을 다르게 사용
  - 가장 간단하고 빠르게 튜닝 가능
  - 카테고리별 제목의 형식이나 자주 사용되는 단어들이 다름 (e.g. 코로나 → 백신)
  - Few-shot Example을 코로나 관련 컨텐츠로 구성한 후 결과 변화



- 키워드 자동 생성의 한계

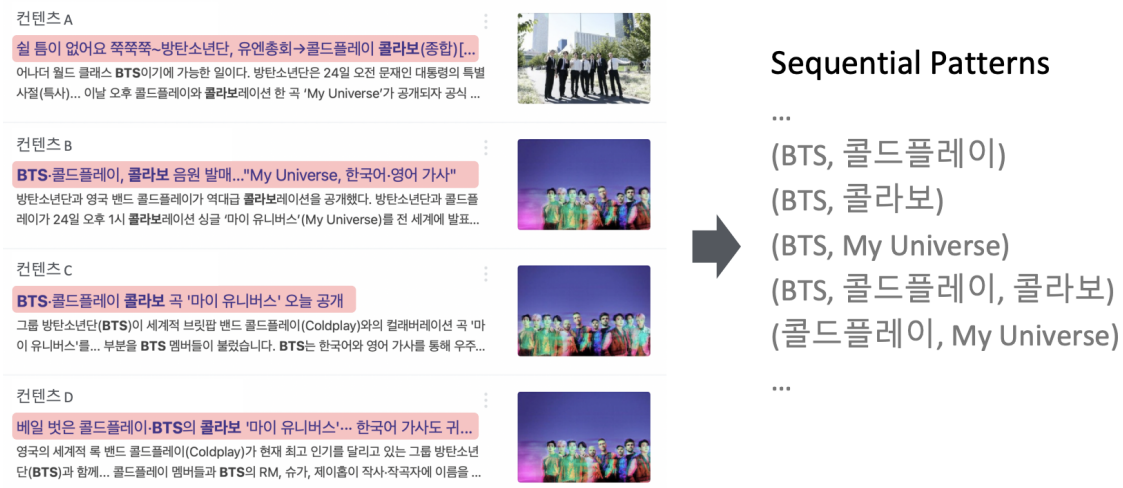
1. 키워드 생성 비용
2. 자동 생성 키워드 품질 이슈. 80%가 바로 서비스 할 수 있는 정도의 키워드를 생성하고, 20% 품질이 낮은 키워드를 생성. 이를 해결하고자 품질이 낮은 키워드를 잘 필터링하는 로직 필요.

3. 검색 다양화를 위한 키워드 확장과 개인화 추천 (3. 랭킹모델 (개인화 추천)에 해당)

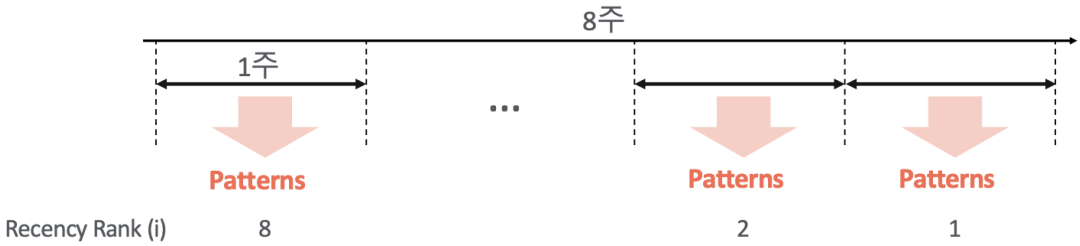
다양한 추천 키워드 생성

구상

- 컨텐츠의 제목을 활용하고자 함 ⇒ 규모가 큰 컨텐츠의 제목들에서 빈번하게 나오는 sequential 패턴을 찾고자 함.



- Sequential Pattern Matching
  - 너무 시의성 짙은 화젯거리는 지양함 → HyperCLOVA 모델이서 이를 수행
  - 어느 정도 steady한 패턴을 찾으려함
  - 패턴의 trendy한 정보를 고려하고자 함



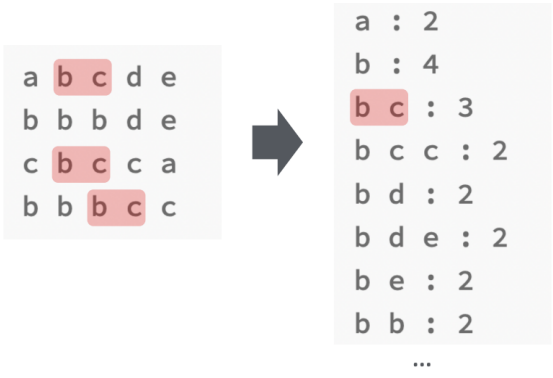
각 패턴에 대한 점수 계산

$$\sum_{i=1}^{8} \frac{support}{\log_2(i+1)}$$

여러 구간에서 뽑힐수록,
각 구간에서의 **support (출현 빈도)값이 클수록**,
최근에 뽑힐수록 값이 커짐

- 위 식에서
  - support ⇒ 출현 빈도값이 클수록 값이 커짐
  - $log_2(i+1)$ ⇒ 최근을 의미하는 i가 최근일수록 값이 작아지므로, 패턴의 trendy에 어느정도 점수를 부여할 수 있음
  - $\sum_{i=1}^{8}$⇒ 합산을 통해 여러구간에 뽑힐수록 값이 커짐. steady 반영

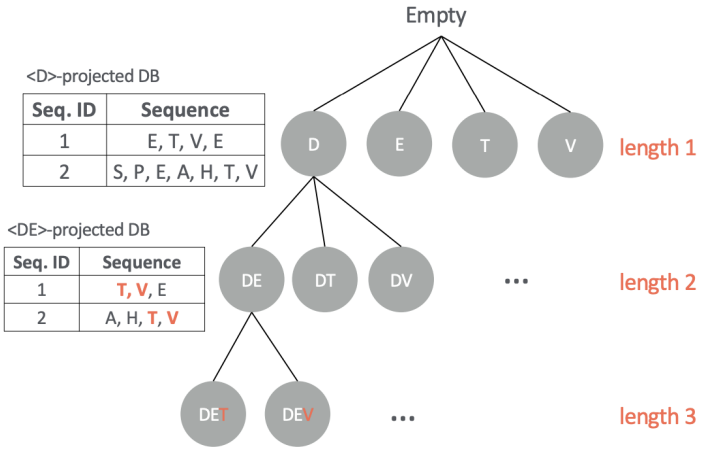
- Sequential Pattern Mining
  - 첫번째 시도 : PrefixSpan 알고리즘 사용 ⇒ 컨텐츠 제목으로부터 토큰화 수행



  - Pattern Growth 방식

- Pattern Growth 방식

Seq. ID	Sequence
1	C, D, E, T, V, E
2	D, S, P, E, A, H, T, V

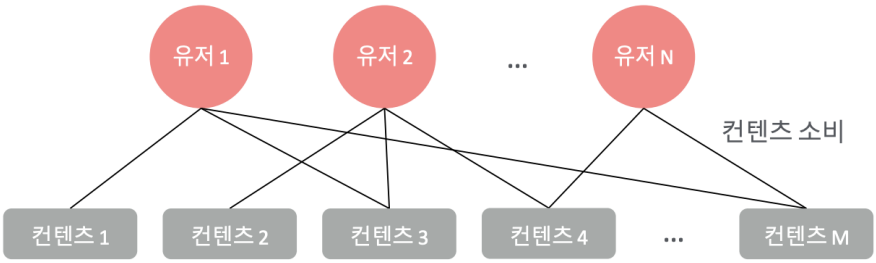


- 부적절한 단어제거
- 다소 어색한 패턴이 발생하는 경우에 대해 품질 개선 시도. 해결 방법 ⇒ window 고려!!

Frequent Sequential Pattern을 키워드 개인화 추천에 어떻게 사용할까??

우선 가지고 있는 데이터

1. Frequent Sequential Patterns
2. StarSpace Embedding Vectors → 각 종 entity를 임베딩하는 기법. entity 타입에 상관없이 (*) 같은 공간 (**space**)상에 매핑. ⇒ “**StarSpace”**. user embedding과 content embedding을 가지고 있음.

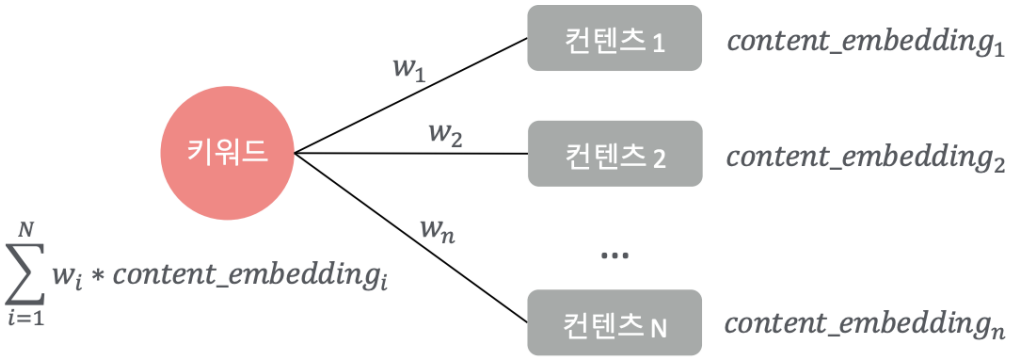


Loss Function $\sum_{\substack{(a,b)\in E^{+} \\ b^{-}\in E^{-}}} L^{batch}(sim(a,b), sim(a,b_1^{-}), \dots, sim(a,b_k^{-}))$

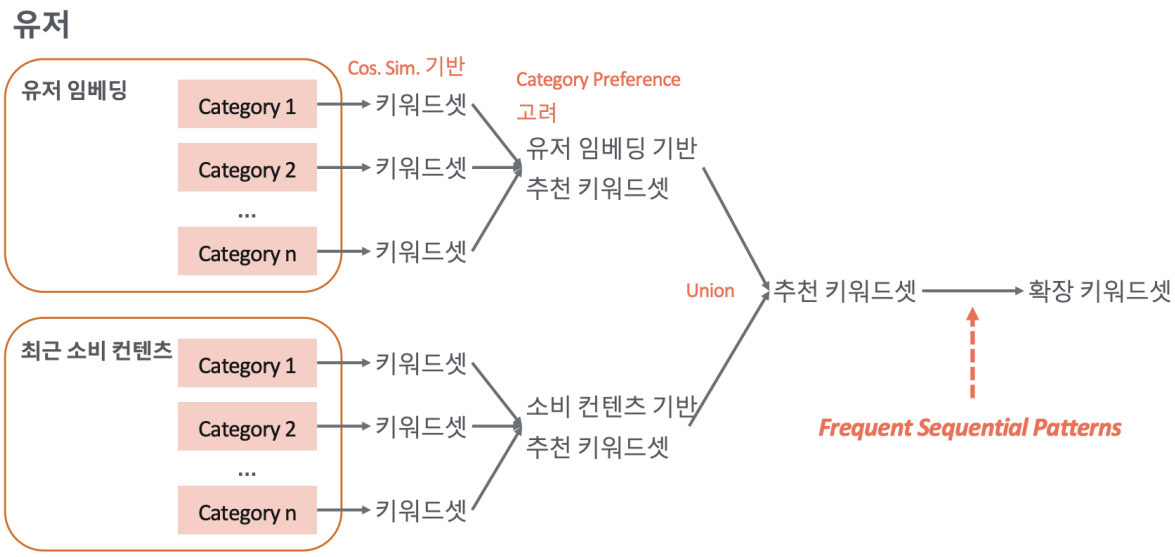
키워드 임베딩

- embedding vector가 user와 content에 대해서만 존재하므로, 키워드에 대한 embedding을 만들기 위해 연관 컨텐츠의 임베딩으로부터 계산. 이때, 각 키워드는 컨

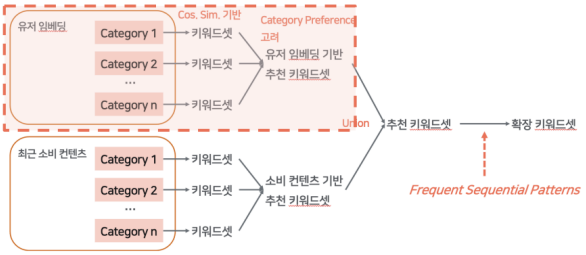
텐츠마다 연관도가 다르므로 각 컨텐츠와의 연관도 기반 weighted sum



키워드 추천 흐름도 (WIP)



- 카테고리별로 유저 임베딩과 콘텐츠 임베딩을 이용하여 키워드셋 추출
- 키워드 추천 예시

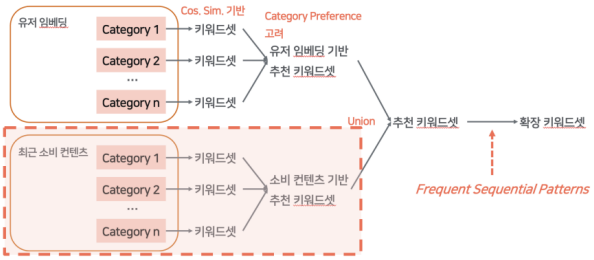


Category Preference

읽은 컨텐츠 수 / 날짜 수에 기반

엔터테인먼트: 0.557922
인터넷IT: 0.162259
문화: 0.147509
음식: 0.132310

엔터테인먼트	음식	인터넷 IT
BTS	로손	iPhone
JUNG KOOK	초밥	Apple Music
SUGA	카레	Google Pixel
BLACKPINK	미니스톱	Apple Watch
Wanna One	식빵	Microsoft Surface
Butter	코메다	ASUS
…	…	…



엔터테인먼트

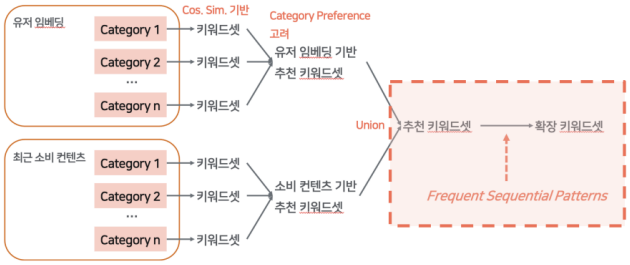
BTS: 0.764609
BLACKPINK: 0.745327
SUGA: 0.727481
JUNG KOOK: 0.686835
리사: 0.675646
Wanna One: 0.622867
Butter: 0.615348
Red Velvet: 0.577691
...

음식

스시: 0.625859
갓파스시: 0.623755
카레: 0.611984
로손: 0.601496
불고기: 0.546414
미니스톱: 0.528504
크로와상: 0.521363
...

인터넷 IT

Apple Watch: 0.679642
iPhone: 0.613610
Apple TV: 0.552783
Apple Music: 0.500514
Google Pixel: 0.474261
헤드폰: 0.436094
Samsung Galaxy: 0.425419



시드

Apple Watch
크로와상
Samsung Galaxy
햄버거
초밥
Red Velvet
코메다
Pokémon GO
TWICE
GOT 7
...

확장된 키워드셋

applewatch 밴드

...
redvelvet 새 드라마
redvelvet photo
redvelvet 미니앨범 queendom
redvelvet queendom
redvelvet 패션
redvelvet 조이 crush
redvelvet crush
redvelvet 조이
...

...
twice mv 재생 횟수
twice 일본인 멤버
twice 영상
twice 모모 공개
twice 근황 공개
twice 트와이스 메이크업
...

4. Ranking & System Pipeline

- 2020년도 교훈
  - 하단에 노출하는 키워드일수록 개인화가 잘 작동
  - 즉, 상위 랭크에는 모두가 관심있어 할만한 **화제 키워드**를..
    하위 랭크로 갈수록 개인화가 강화된 **개인화 키워드**를..
- rank 1 키워드 선정 파이프라인
  - 피드백 로그를 활용한 rank 1 키워드 선정.
    test group을 대상으로 키워드를 노출하고, 이에 대한 피드백 로그를 통해 최적 키워드 선정. 이를 10분마다 실행. 선정된 키워드는 control Group 사용자들의 rank1 키워드로 노출