


# Knowledge Graph에게 맛집과

## 사용자를 묻는다.

### : GNN으로 맛집 취향 저장하기!



# AirSPACE

## AI Recommender System from Place And Context Embedding (+User)

NAVER SME 풀케어 시스템

온라인 사업 오프라인 사업 교육 기술 기업용 솔루션 유용한 정보

프로젝트 꽃 5TH

### AI 기반 장소 추천 엔진

## AirSPACE

네이버데이터를 기반으로 최적의 장소를 추천  
새로운 발견과 만남의 기회를 제공합니다



네이버 데이터를 기반으로  
가볼 만한 최적의 장소를 추천



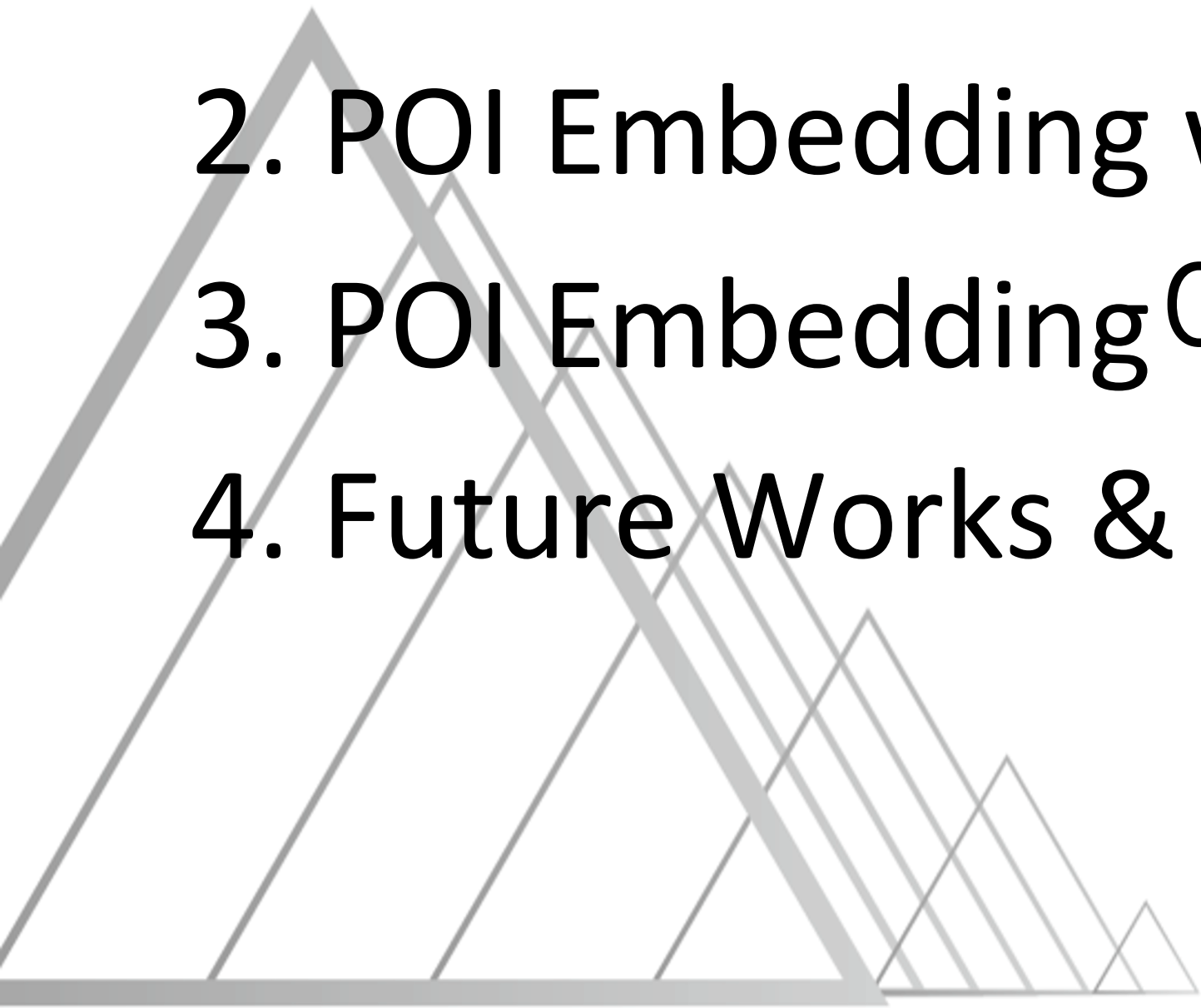
스마트어라운드와 지도 앱 등  
위치 기반 서비스에 기술 적용



잠재 고객과의 만남을 연결하고  
사용자에게 새로운 발견의 기회

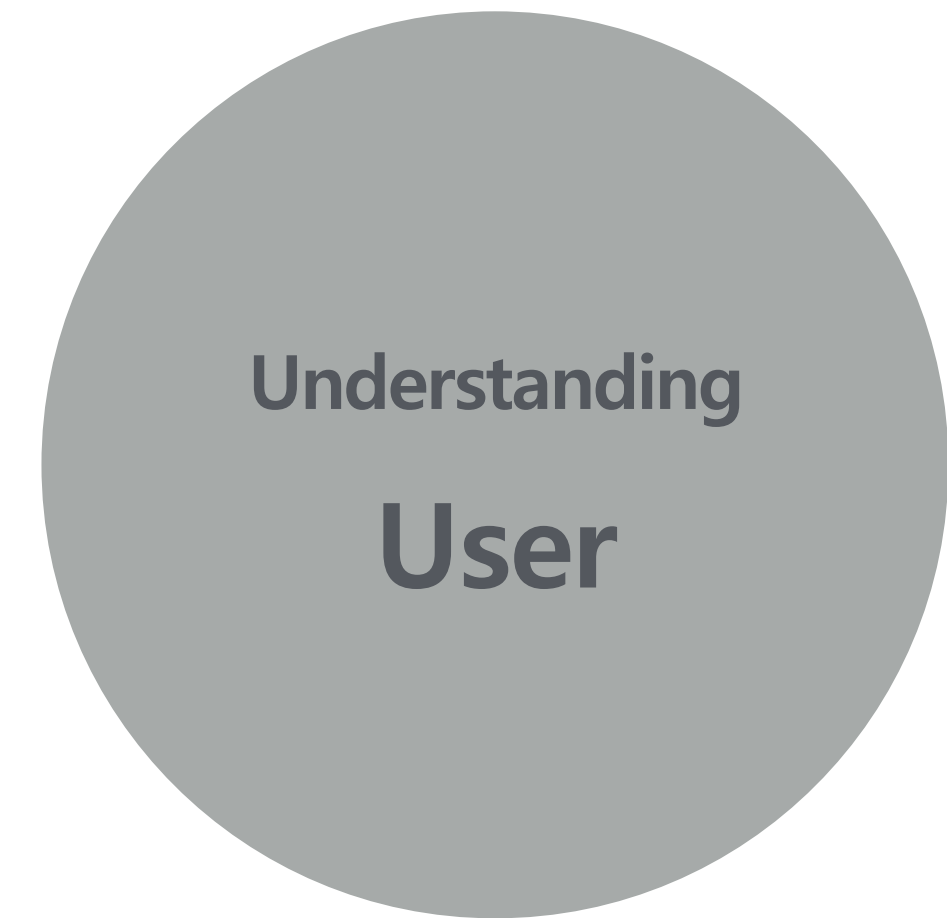
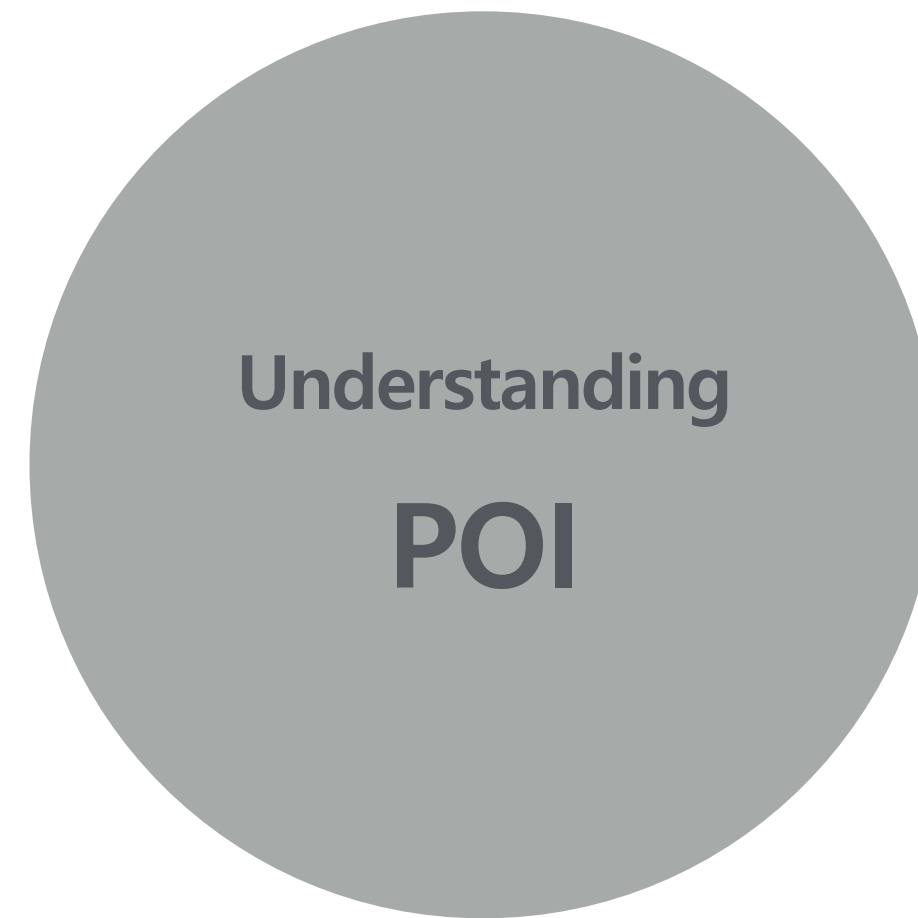
# CONTENTS

1. 장소를 추천한다는 것은?
2. POI Embedding with Graph
3. POI Embedding이 잘 되었다는 것은 어떻게 평가할 수 있을까?
4. Future Works & Conclusion

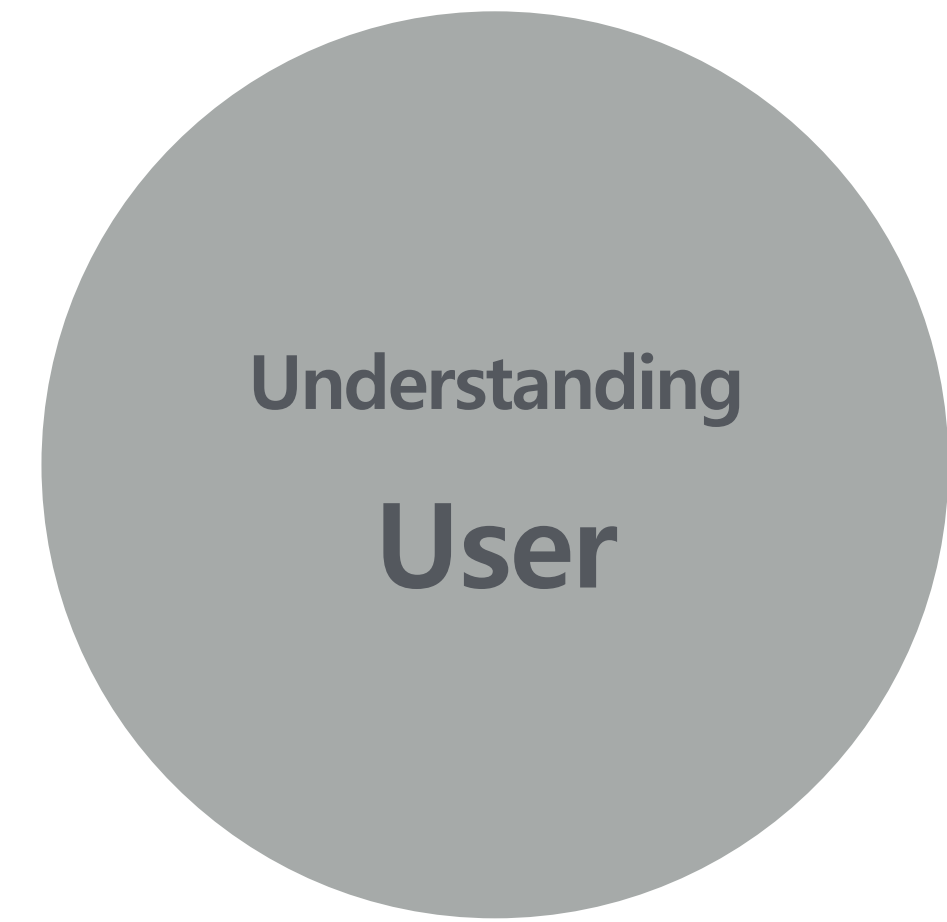
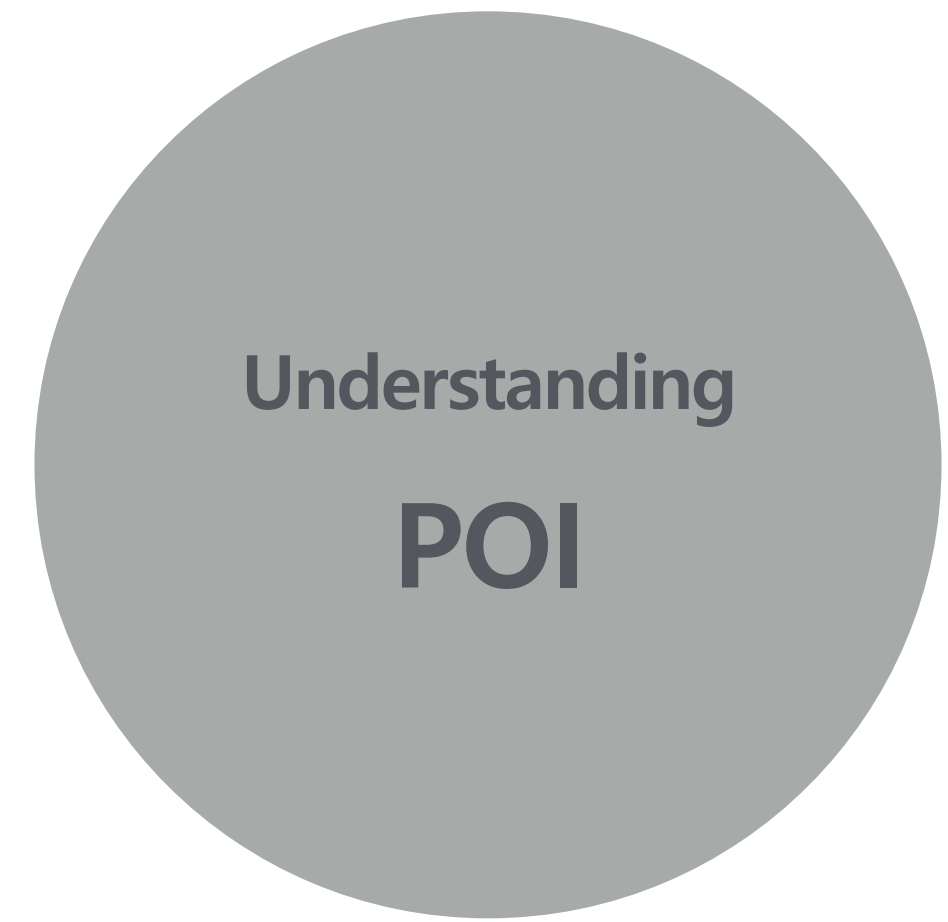


# 1. 장소를 추천한다는 것은?

# 1.1 장소를 추천한다는 것은?

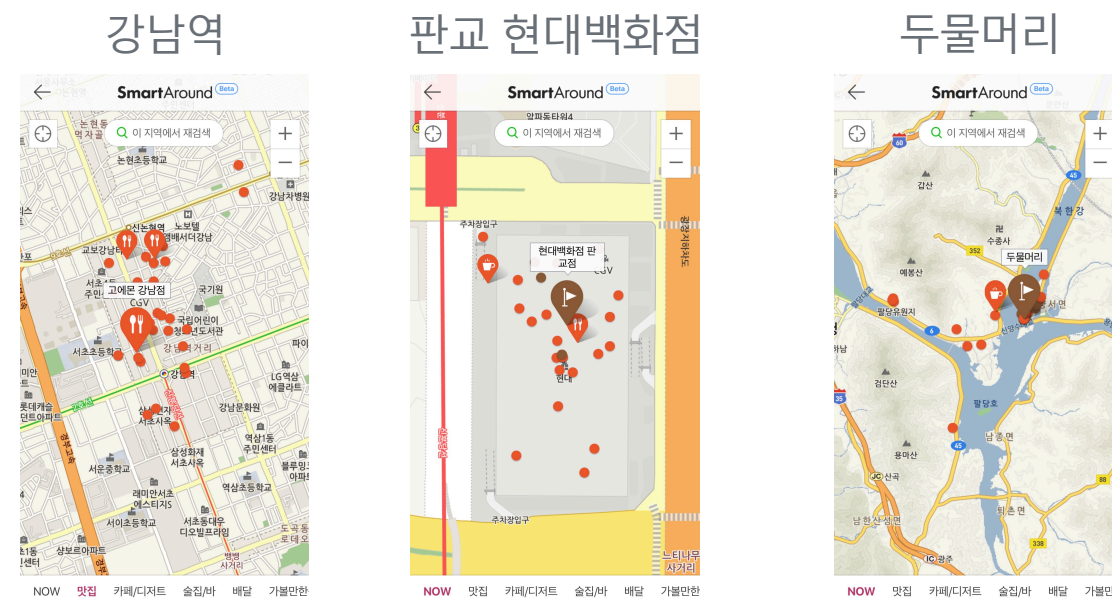


# 1.1 장소를 추천한다는 것은?

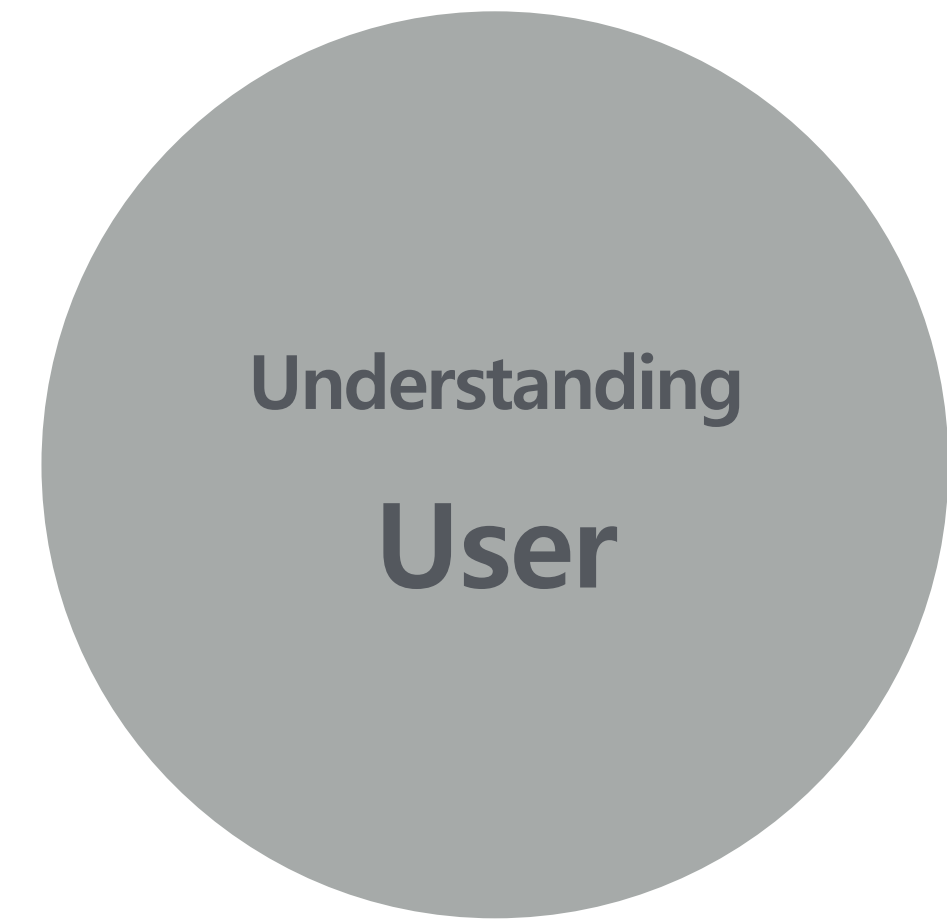
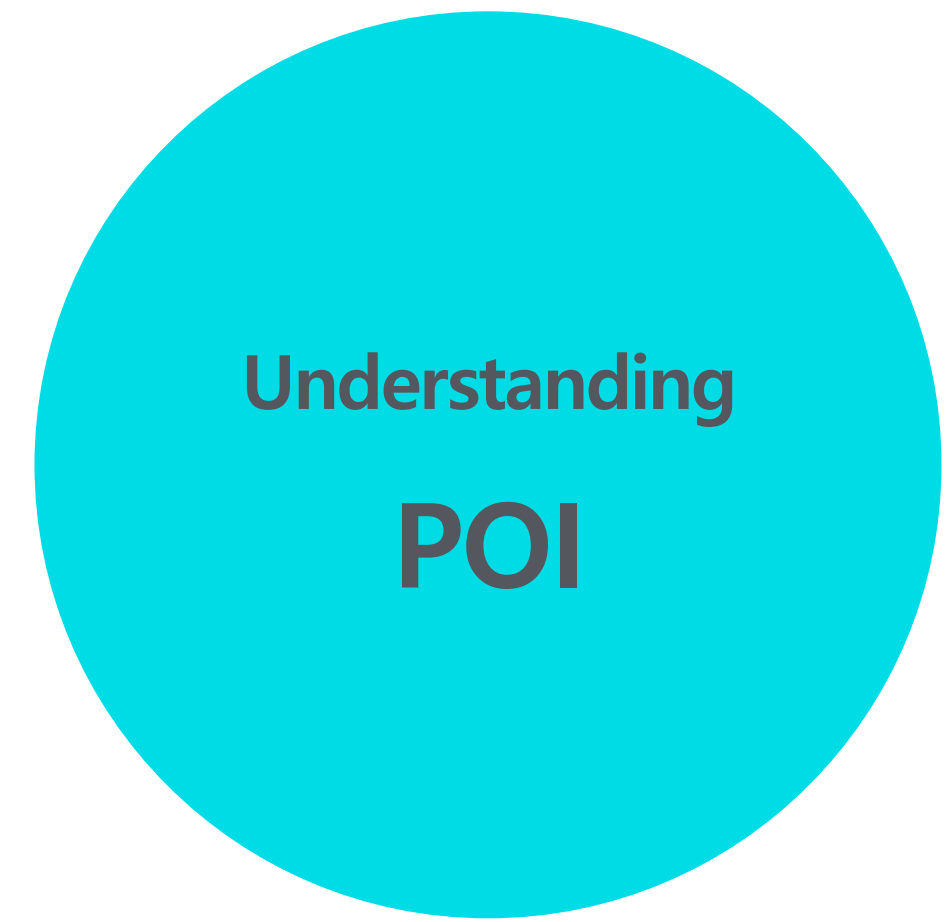


“이 주변 어디까지 가볼까?”

“이 지역은 어떤 특성을 가진 지역일까?”



# 1.1 장소를 추천한다는 것은?

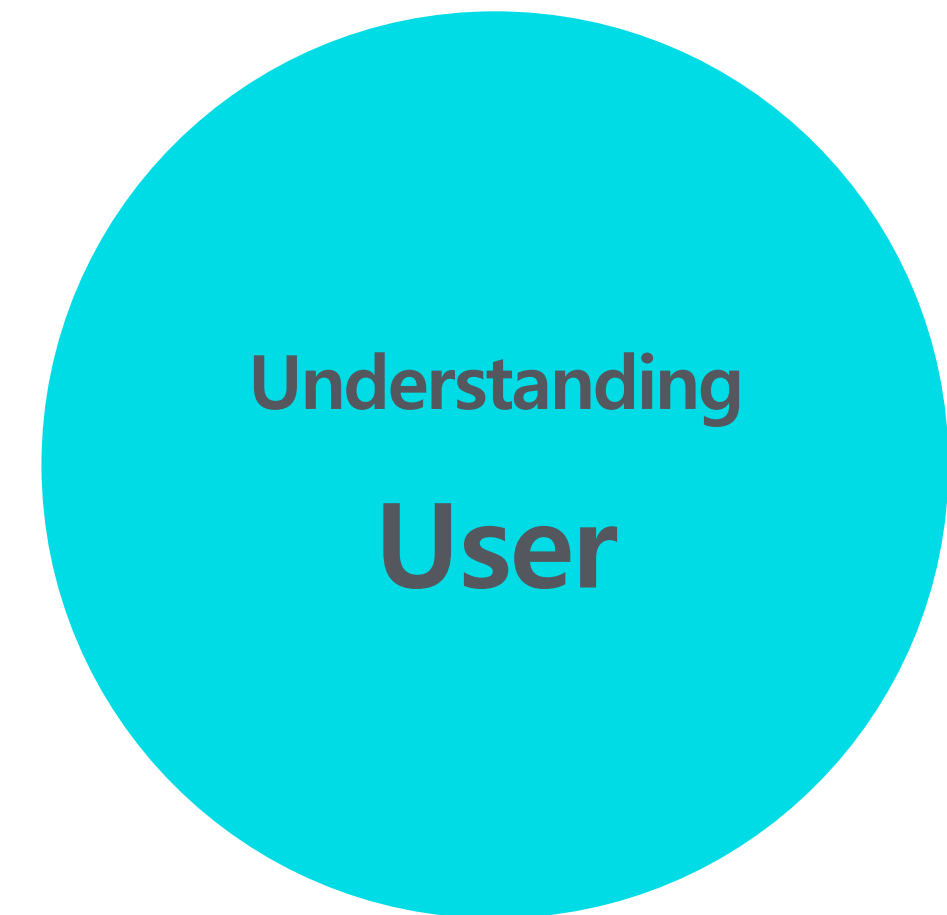
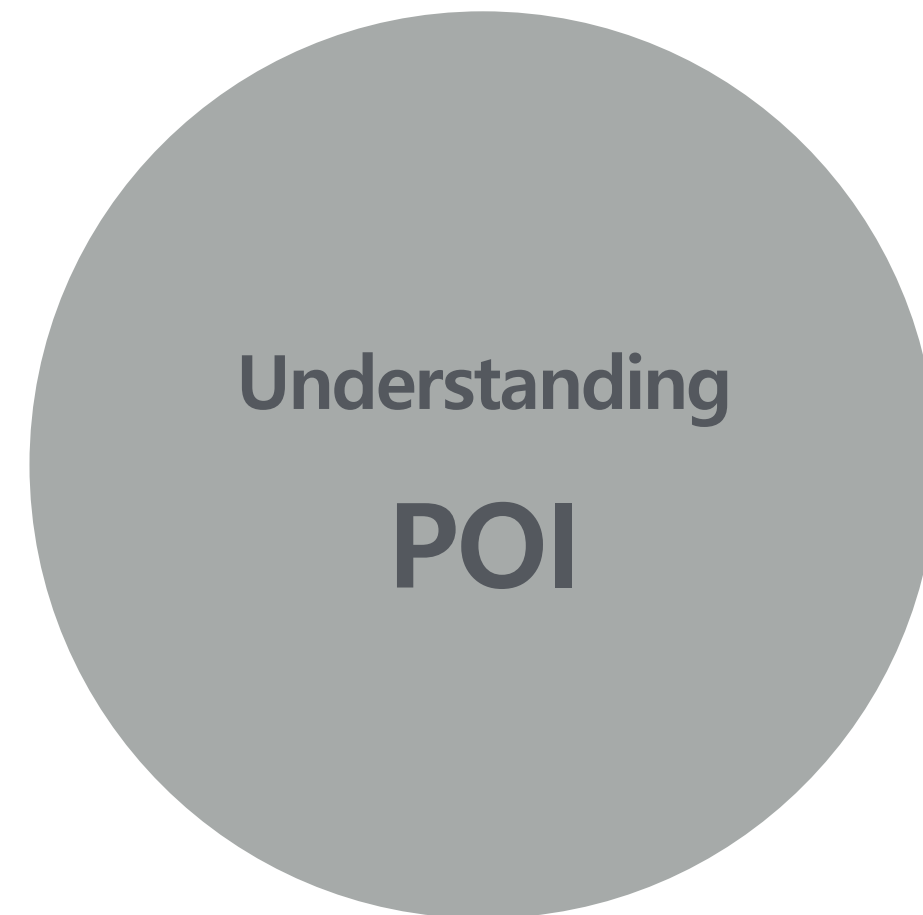


“이 장소는 어떤 사람들이 찾을까?”  
“이 장소는 언제 자주 찾을까?”

POI\* : Point of Interest의 약자,  
관심 장소를 말함. Ex) 맛집

The screenshots illustrate data analysis for a Point of Interest (POI). The first screenshot shows a list of restaurants with filters for cuisine type, popularity, and distance. The second screenshot shows a detailed view of a restaurant with a heatmap overlay indicating user density. The third screenshot shows a bar chart and a pie chart representing user demographics and time-based trends.

# 1.1 장소를 추천한다는 것은?



“나의 취향과 관심사는 무엇일까?”

“나와 비슷한 취향을 지닌 사람들은 누구일까?”

이름: 김여자  
나이: 23  
성별: 여자  
시간: 오후 12:30  
취향: 우아한 분위기

이름: 김남자  
나이: 26  
성별: 남자  
시간: 오후 6:30  
취향: 국물, 기름진





# 1.1 장소를 추천한다는 것은?

Understanding  
**Location**

Understanding  
**POI**

Understanding  
**User**

“이 주변 어디까지 가볼까?”

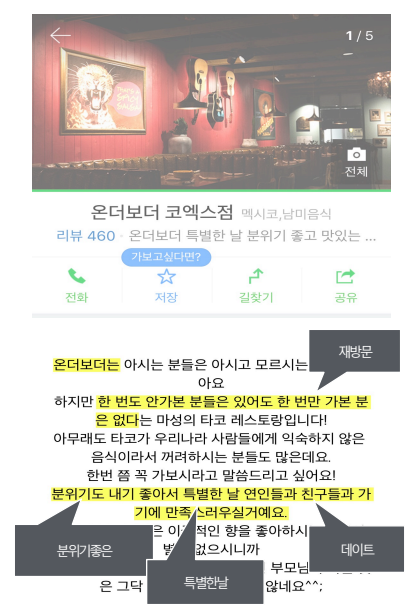
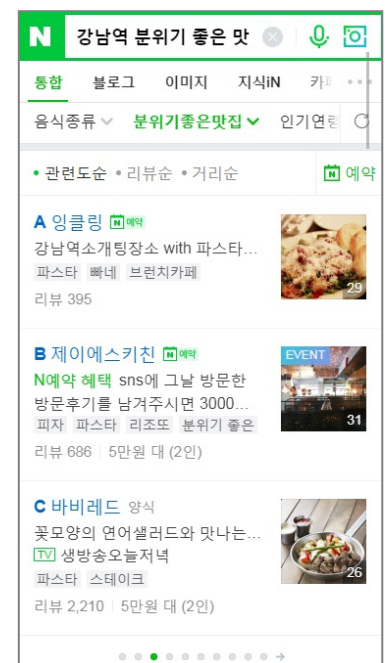
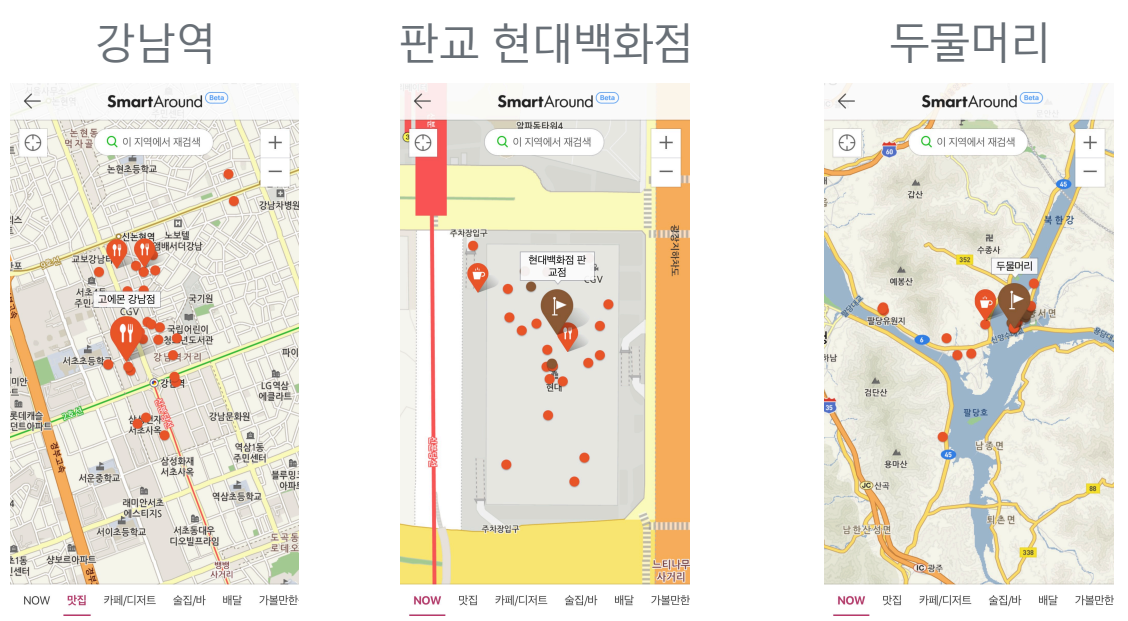
“이 장소는 어떤 사람들이 찾을까?”

“나의 취향과 관심사는 무엇일까?”

“이 지역은 어떤 특성을 가진 지역일까?”

“이 장소는 언제 자주 찾을까?”

“나와 비슷한 취향을 지닌 사람들은 누구일까?”



**이름:** 김여자  
**나이:** 23  
**성별:** 여자  
**시간:** 오후 12:30  
**취향:** 우아한 분위기

**이름:** 김남자  
**나이:** 26  
**성별:** 남자  
**시간:** 오후 6:30  
**취향:** 국물, 기름진



# 1.1 장소를 추천한다는 것은?

Understanding  
Location

Understanding  
POI

Understanding  
User

Location과 POI, User를 이해하여

User의 취향에 맞는 장소를 상황에 맞추어 제안해주는 것

“이 주변 어디까지 가볼까?”

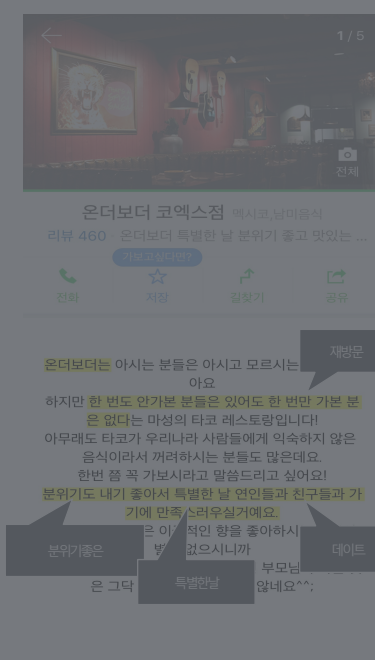
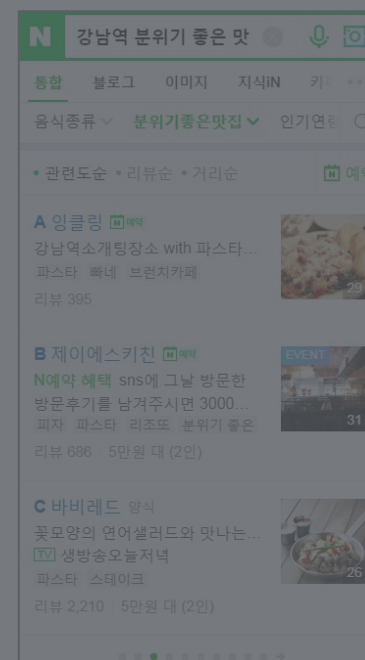
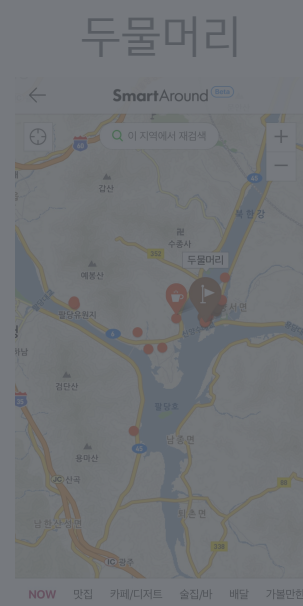
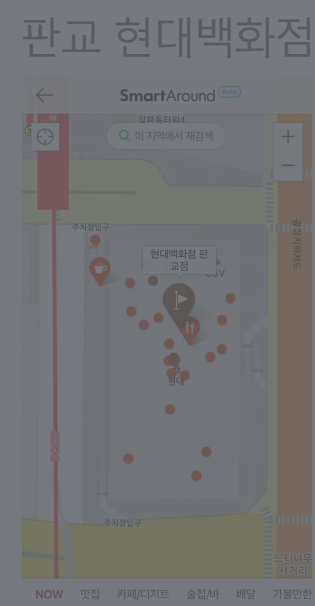
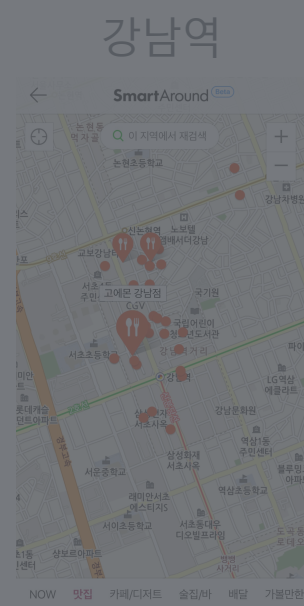
“이 장소는 어떤 사람들이 찾을까?”

“나의 취향과 관심사는 무엇일까?”

“이 지역은 어떤 특성을 가진 지역일까?”

“이 장소는 언제 자주 찾을까?”

“나와 비슷한 취향을 지닌 사람들은 누구일까?”



**이름:** 김여자  
**나이:** 23  
**성별:** 여자  
**시간:** 오후 12:30  
**취향:** 우아한 분위기

**이름:** 김남자  
**나이:** 26  
**성별:** 남자  
**시간:** 오후 6:30  
**취향:** 국물, 기름진



# 1.2 장소를 이해하는 다양한 기술

POI에서 분위기, 목적 등  
주요 키워드를 추출

현위치, 시간, 실시간 검색의도  
기반 POI 추천

유저의 취향을 분석해서  
개인화 POI 추천

DEVIEW 2016

## 딥러닝을 이용한 지역 컨텍스트 검색

Local Context Search Utilizing Deep Learning

김진호  
NAVER Search



**2016 Deview**

## Location Based Recommendation

김진호  
AirSPACE



**2018 AI 콜로키움**

DEVIEW 2020

## 당신 취향의 맛집을 추천드립니다 : 장소 개인화 추천 시스템의 비밀

*Collaborative Filtering Meets the Item Embedding*

전영환 NAVER Search

**2020 Deview**

LOCAL SEARCH PROJECT

## POI Understanding AI. ConA

최지훈 리더  
NAVER SEARCH / LOCAL & PRODUCT

**2017 AI 콜로키움**

NAVER AI Colloquium

## SmartAround

From embedding to POI recommendation

김재균 / SmartAround



**2019 AI 콜로키움**

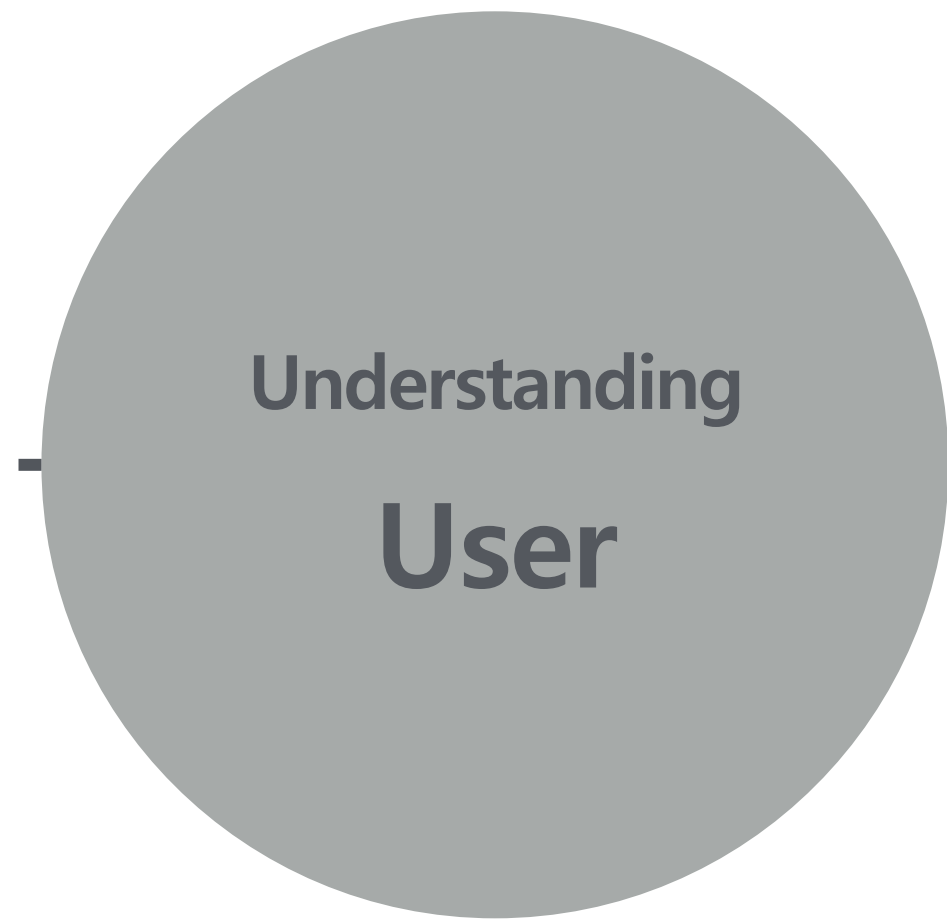
## Understanding POI

- From Context to Embedding

김창희 NAVER / AirSPACE  
전영환 NAVER / AirSPACE

**2021 Search 콜로키움**

# 1.3 장소에 대한 이해는 좋은 추천으로 이어진다.



POI\* : Point of Interest의 약자,  
관심 장소를 말함. Ex) 맛집

# 1.3 장소에 대한 이해는 좋은 추천으로 이어진다.



이름: 김여자

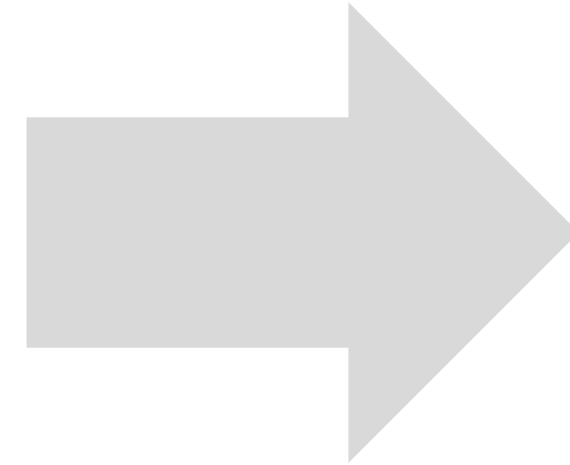
나이: 23

성별: 여자

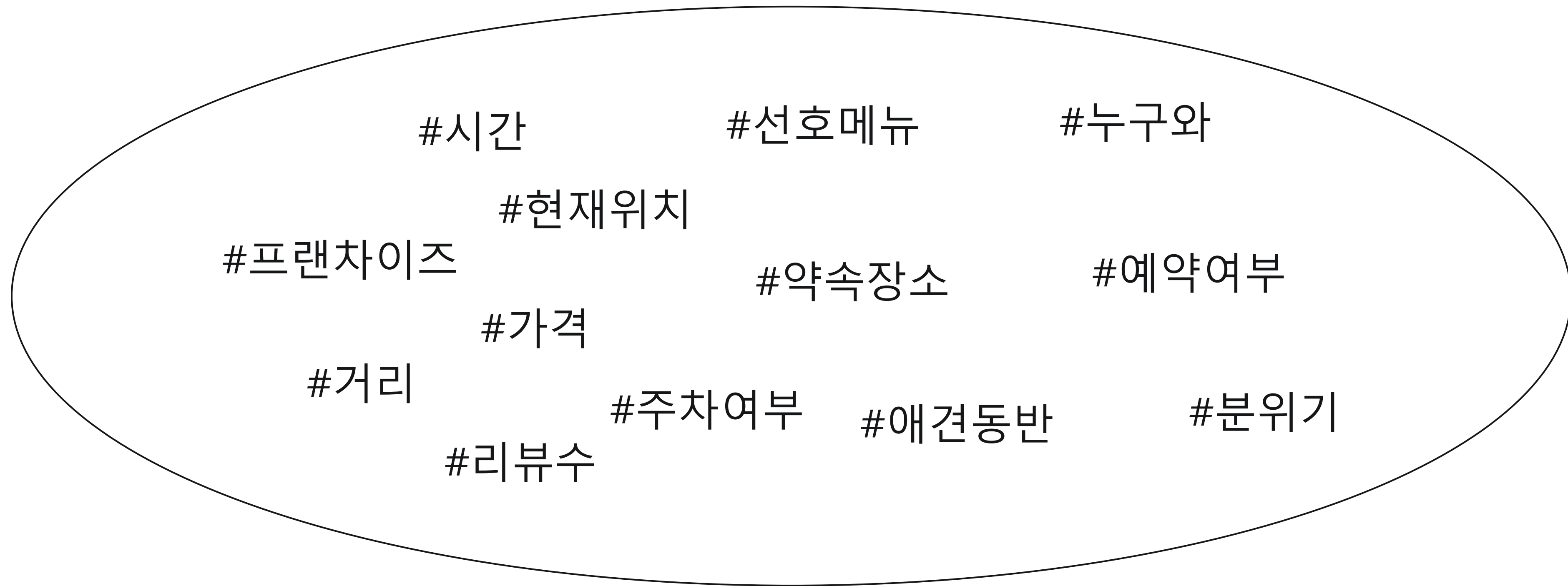
시간: 오후 6:30

취향: 우아한 분위기, 양식

위치: 가로수길

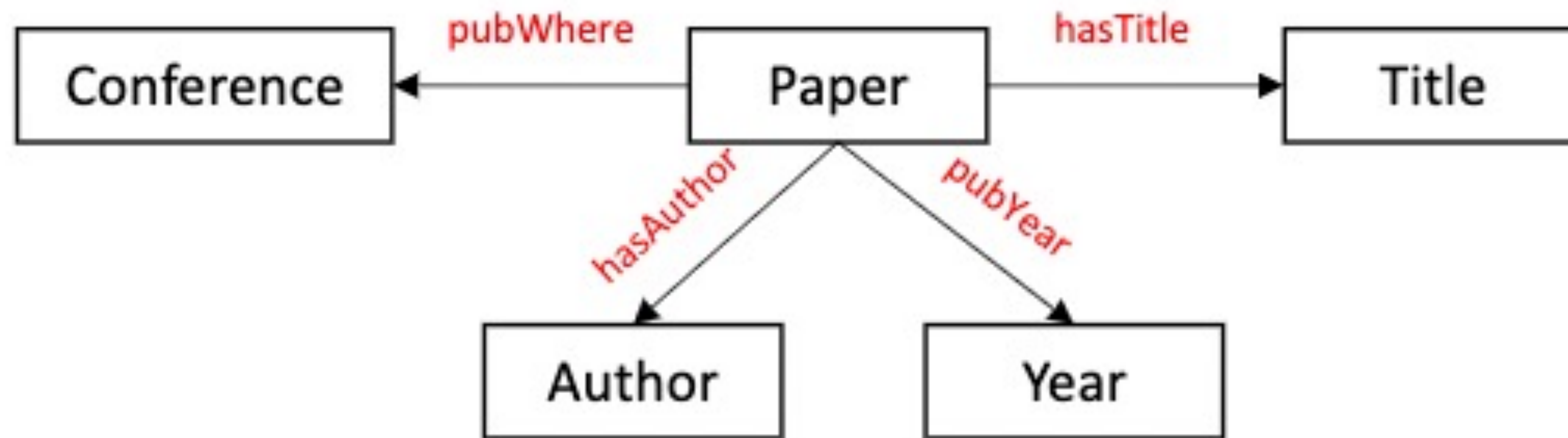


# 1.3 장소에 대한 이해는 좋은 추천으로 이어진다.



**Context를 종합적으로 고려하여 최적의 POI를 선택하게 된다!**

# 1.4 POI Knowledge Graph



Node: Entity  
 Node Label: Entity Type  
 Edge with Relationship: Relation

특정 Domain에 대한 Knowledge를 Graph 형식으로 저장하는 것

# 1.4 POI Knowledge Graph

## Why Knowledge Graph?

Understanding  
Local/Global  
Structure Deeply

그래프는 노드 간의 관계, 구조,  
연결성에서 오는 정보를 보존함

Flexible and  
Intuitive  
Graph Design

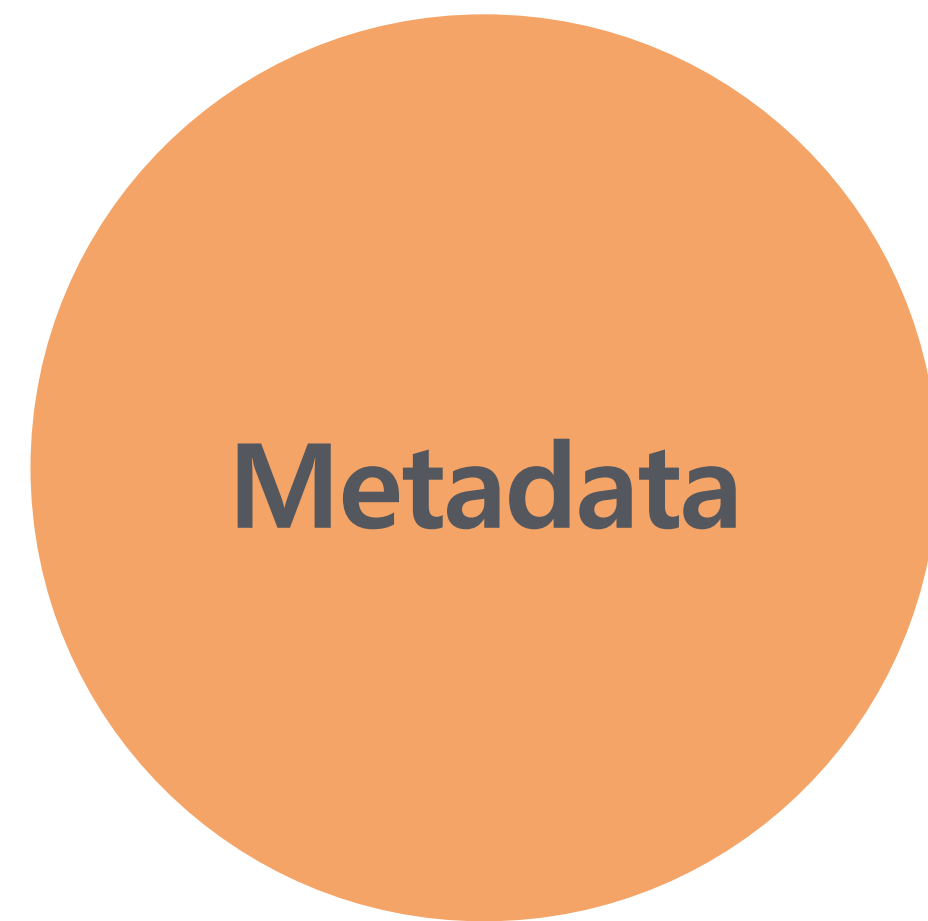
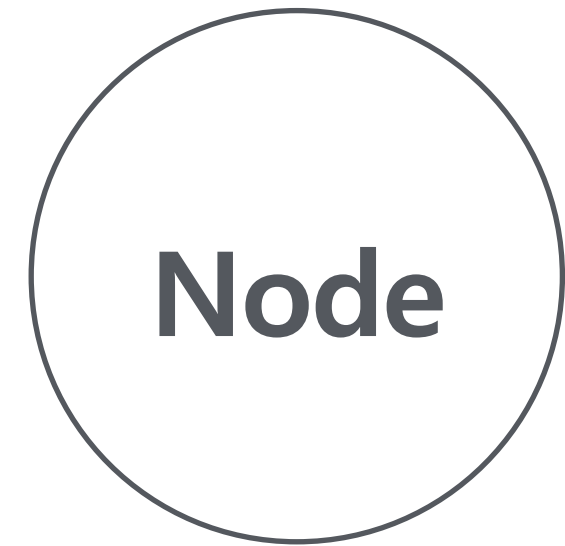
다양한 Type의 노드들을 통해,  
Graph를 다양하게 정의할 수 있음

Huge Collections  
of Entities  
from Multiple  
Sources

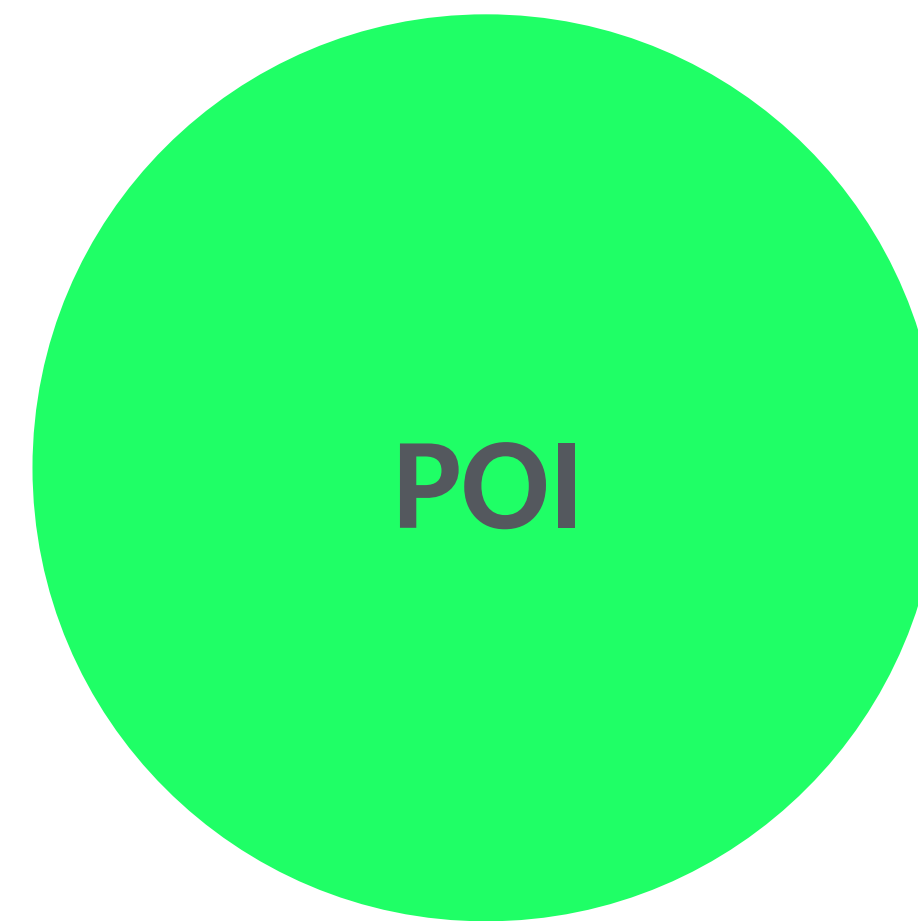
여러 Source의 데이터를 활용하여,  
Graph를 풍부하게 만들 수 있음



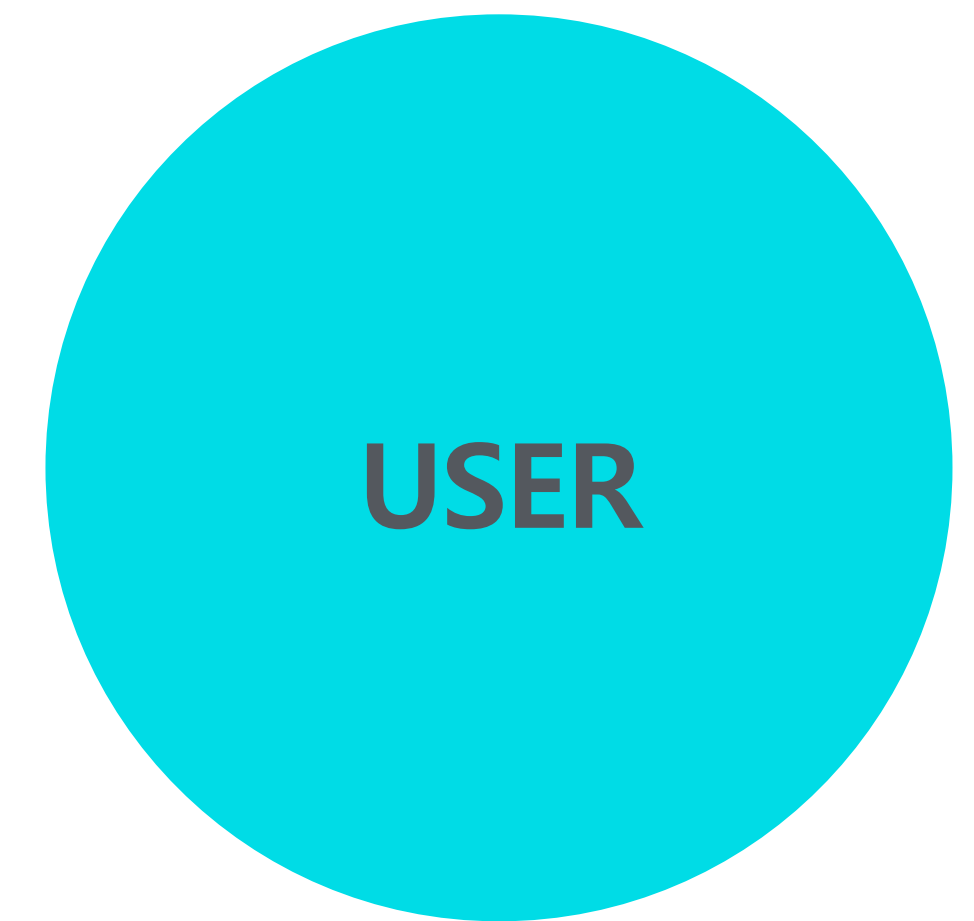
# 1.4 POI Knowledge Graph



POI에 대한 다양한 정보들  
Ex) 분위기, 메뉴, 가격대, 업종

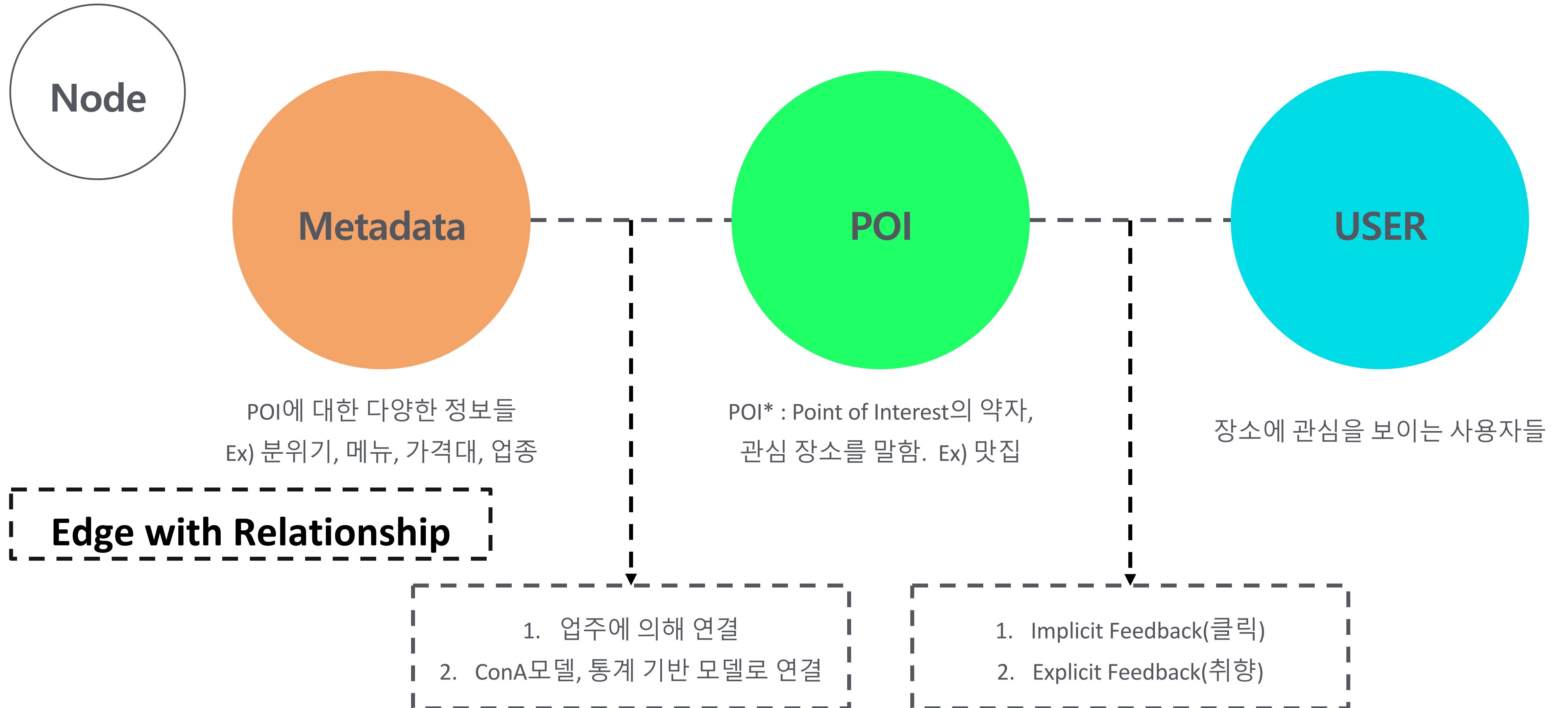


POI\* : Point of Interest의 약자,  
관심 장소를 말함. Ex) 맛집

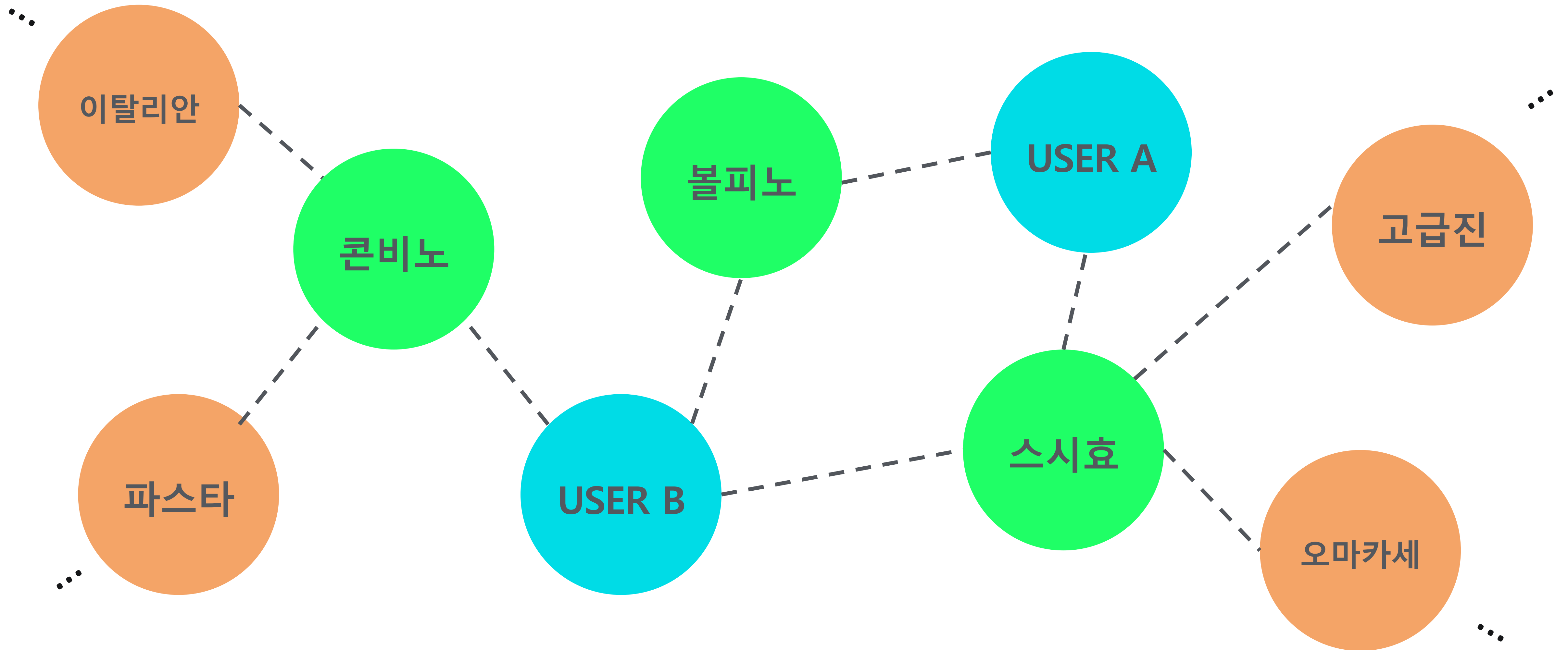


장소에 관심을 보이는 사용자들

# 1.4 POI Knowledge Graph



# 1.4 POI Knowledge Graph



## 2. POI Embedding with Graph

## 2.1 POI Embedding의 과정

**Problem Design**

1. 풀고자 하는 문제 상황 및 Context를 정의

**Graph Design**

2. 상황 및 Context에 맞는 Knowledge Graph Design

**Model Design**

3. Knowledge Graph에 적합한 Modeling

**POI Embedding**

## 2.2 POI2VEC

**Problem Design**

1. 장소를 찾을 때, 근처의 대체재를 어떻게 추천해야할까?

**Graph Design**

2. POI Click Log를 바탕으로 POI Homogeneous Graph를 만들자.

**Model Design**

3. 하나의 검색 질의 내의 POI들끼리 가깝게 Embedding되게 하자.

**POI Embedding**

## 2.2 POI2VEC\_Problem Design

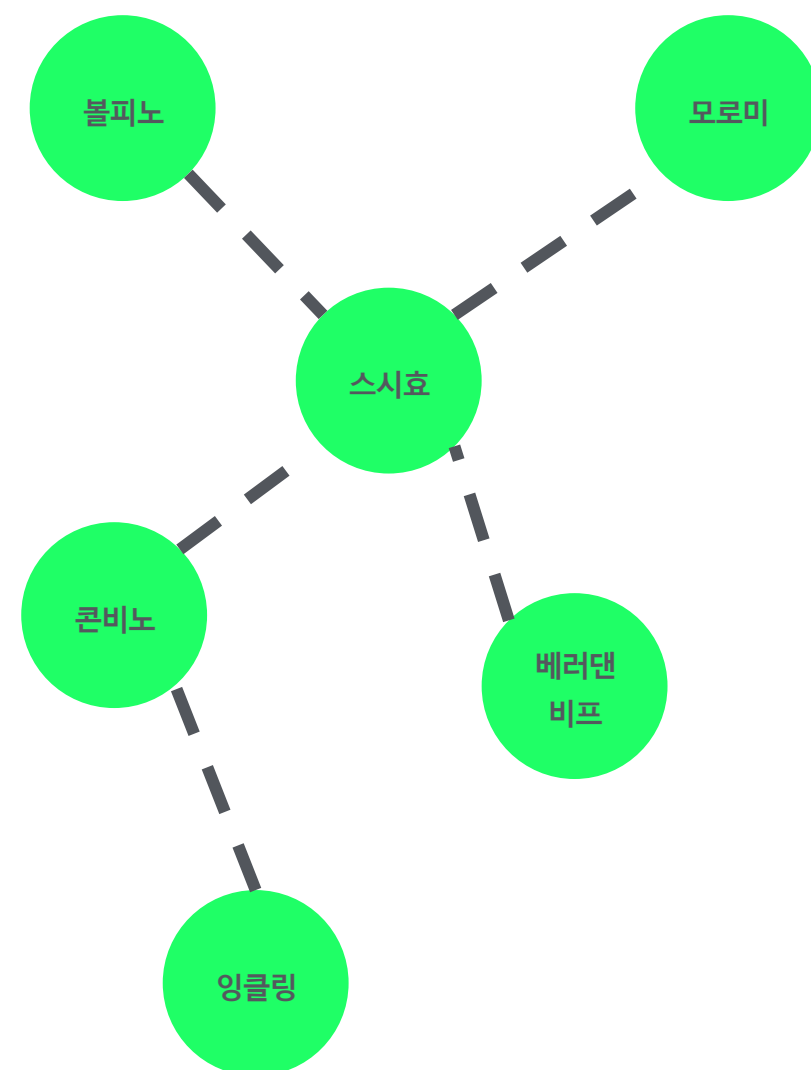
1. 장소를 찾을 때, 근처의 대체재를 어떻게 추천해야할까?



# 2.2 POI2VEC\_Graph Design

2. POI Click Log를 바탕으로 POI Homogeneous Graph를 만들자.

## POI Graph



## Graph Design

1. **Node Design** : POI간의 유사성만 강하게 고려하기 위하여 POI Homogeneous Graph로 Design
2. **Edge Design** : USER가 연속적으로 관심을 보이는 POI들은 Relation으로 연결함

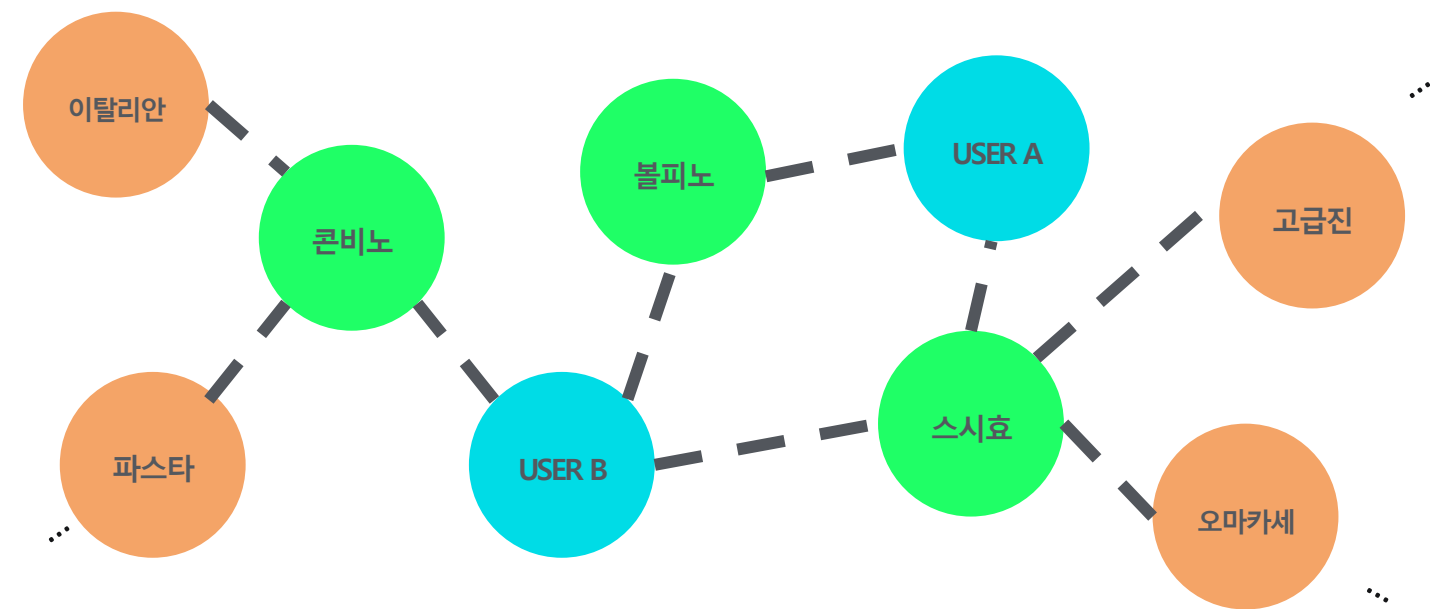
• Homogeneous Graph = Node Type이 한 가지인 그래프



# 2.2 POI2VEC\_Graph Design

2. POI Click Log를 바탕으로 POI Homogeneous Graph를 만들자.

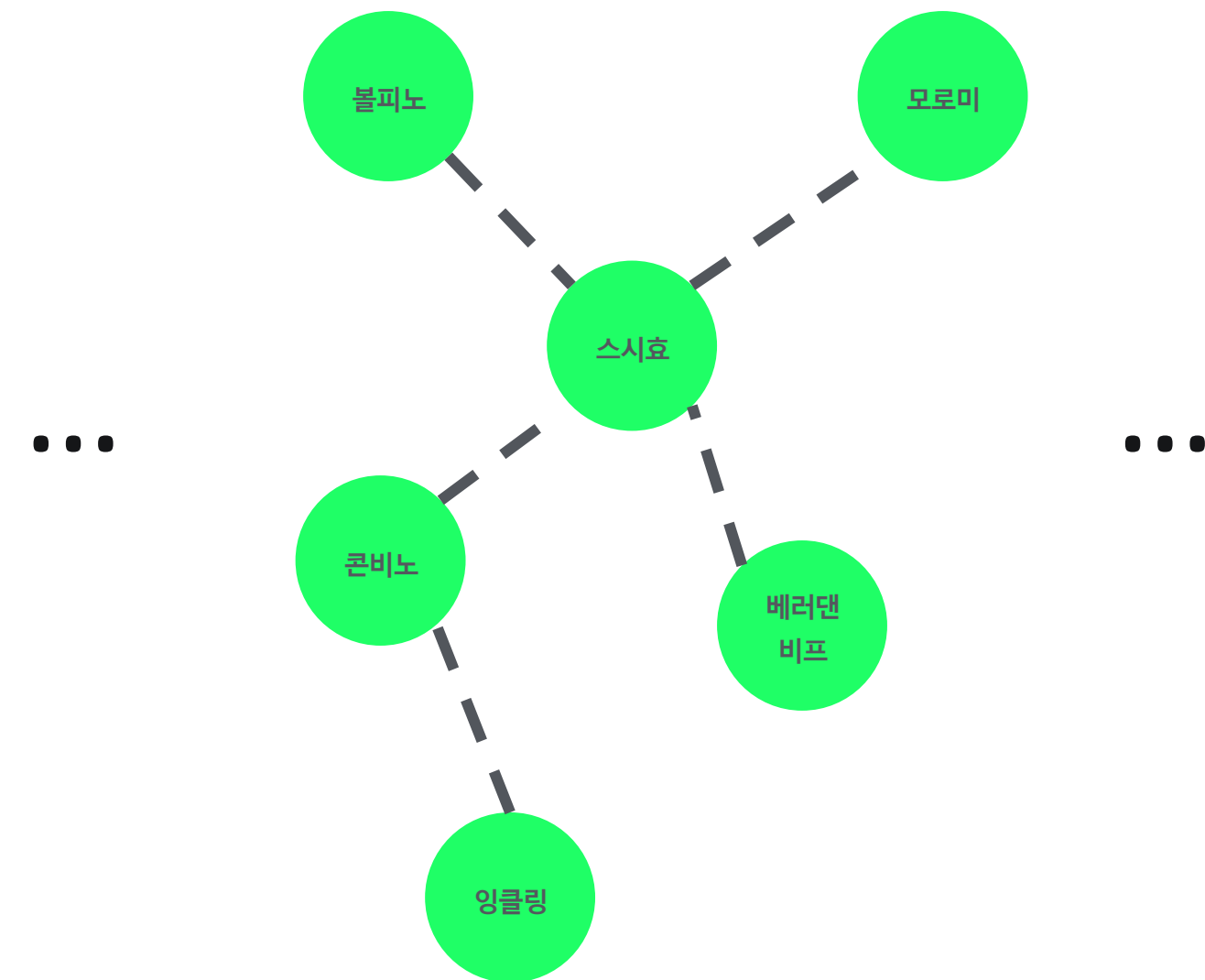
**POI Knowledge Graph**



Projection POI Node

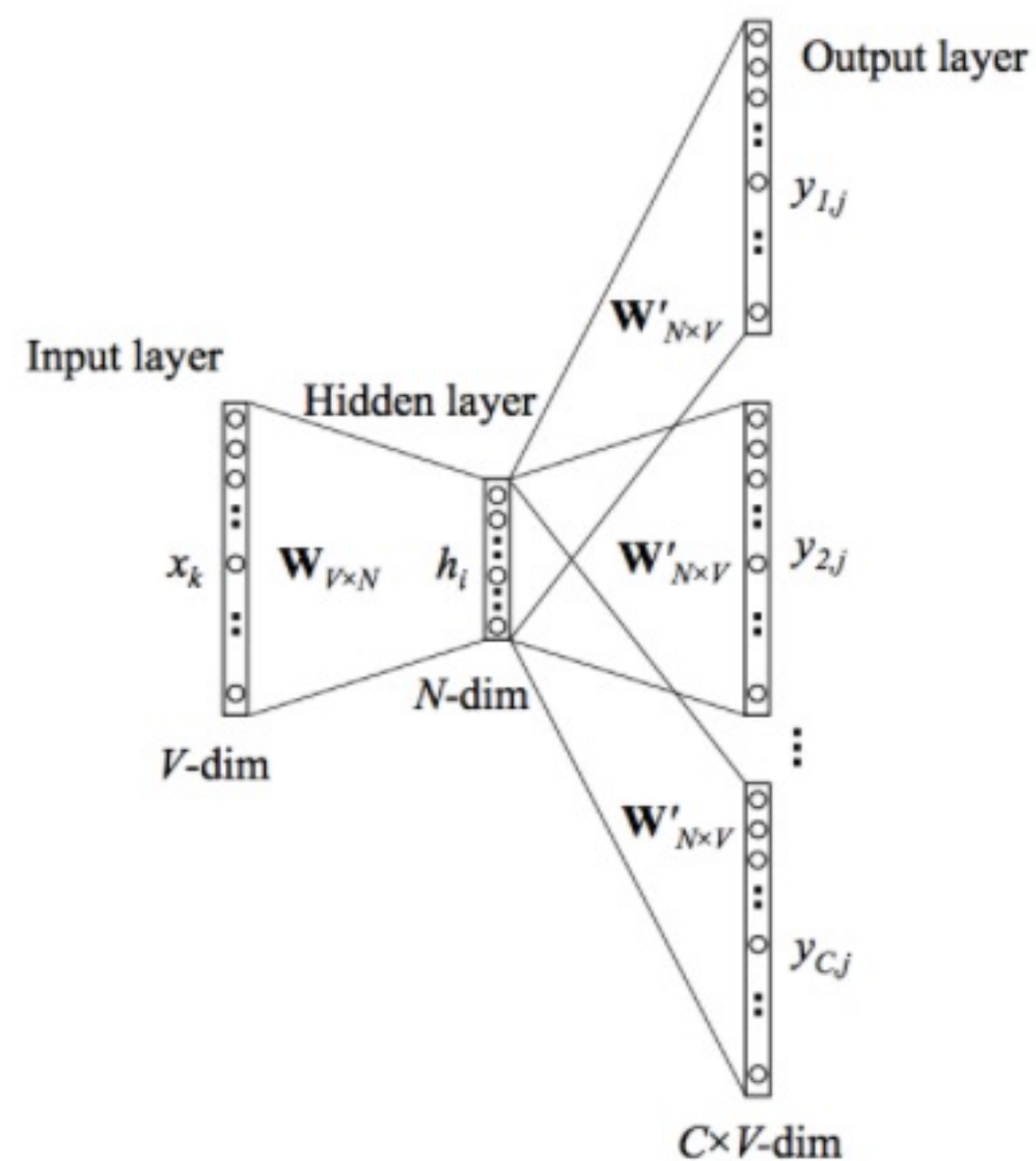
Edge Relation by User Action

**POI Graph**



# 2.2 POI2VEC\_Model Design

3. 하나의 검색 질의 내의 POI들끼리 가깝게 Embedding되게 하자.



$$\theta = \operatorname{argmax}_{\theta} \sum_{(w,c) \in V_{pos}} \frac{1}{1 + \exp(-u_w^T v_c)} + \sum_{(w,c) \in V_{neg}} \frac{1}{1 + \exp(u_w^T v_c)}$$

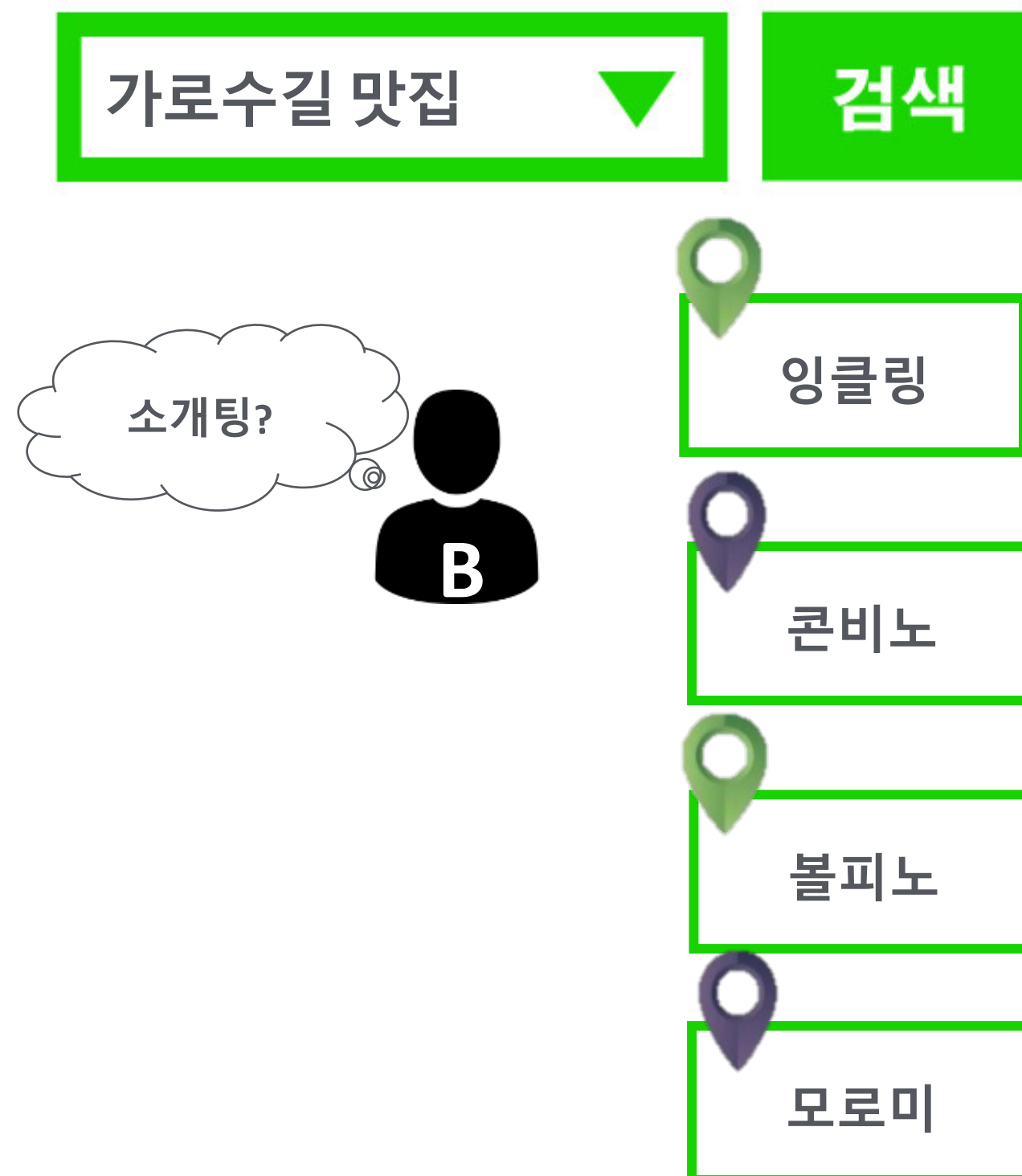
*Sentence = Session*

*Center word(c) = Session 내의 모든 POI*

*Around word(w) = Center word를 제외한 Session 내의 모든 POI*

# 2.2 POI2VEC\_Model Design

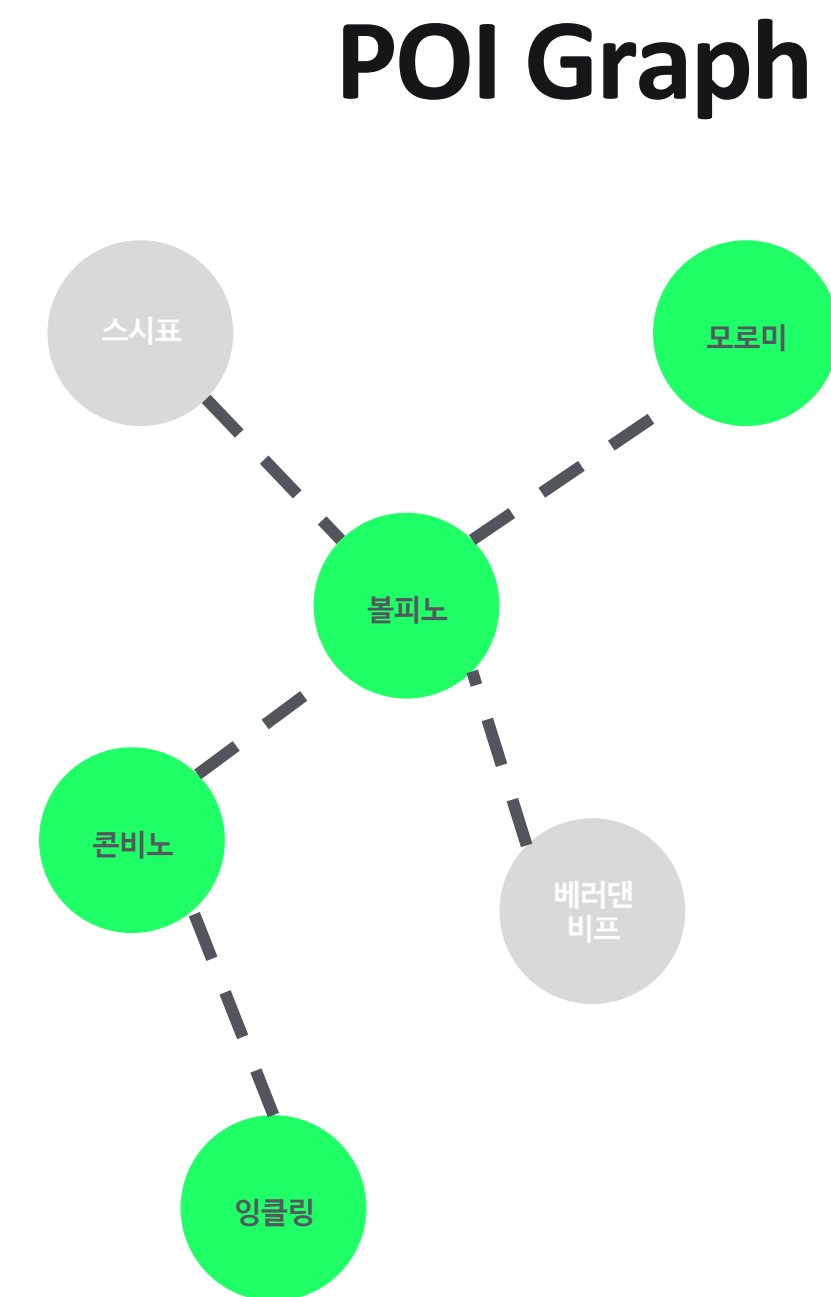
3. 하나의 검색 질의 내의 POI들끼리 가깝게 Embedding되게 하자.



검색 질의 내 동일 의도를 가정

—————→

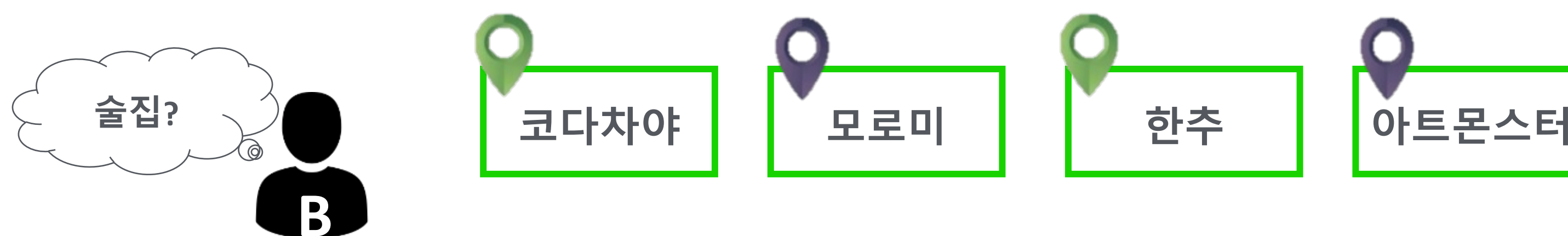
Edge Relation by Session Click



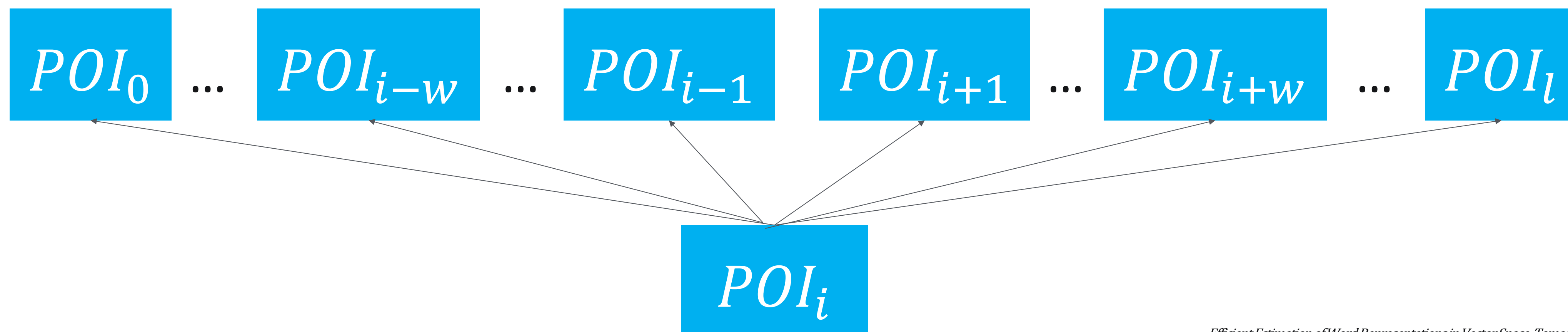
*Session = 유저가 click 행동을 보이는 시간 단위*

## 2.2 POI2VEC\_Model Design

3. 하나의 검색 질의 내의 POI들끼리 가깝게 Embedding되게 하자.



Click sequence



## 2.3 POIMeta2vec

**Problem Design**

1. POI의 Metadata가 유사한 업체를 추천할 수 있을까?

**Graph Design**

2. POI와 Metadata간의 관계를 이용한 Heterogeneous Graph를 만들자.

**Model Design**

3. 공통된 Metadata를 가지는 POI끼리 가깝게 Embedding되게 하자.

**POI Embedding**

# 2.3 POIMeta2vec\_Problem Design

1. POI의 Metadata가 유사한 업체를 추천할 수 있을까?

## 사용자의 취향



쉐이크쉑 강남점 햄버거

#수제버거 #핫플레이스 #아메리칸스타일

## 추천 결과



자코비버거 햄버거

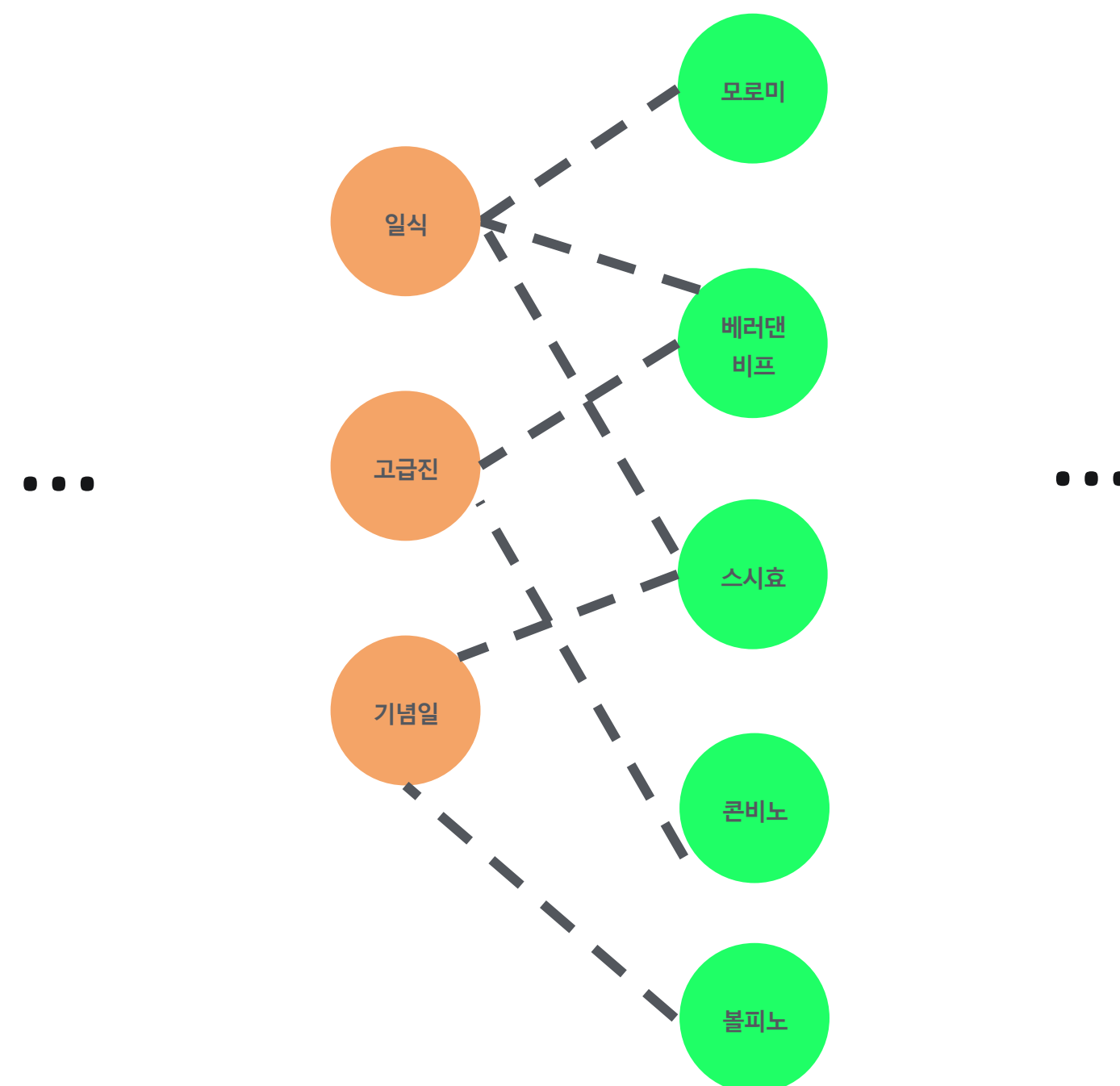


버거마인 햄버거

# 2.3 POIMeta2vec\_Graph Design

2. POI와 Metadata간의 관계를 이용한 Heterogeneous Graph를 만들자.

## POI-Metadata Graph



## Graph Design

### 1. Node Design

- POI의 Metadata를 고려하기 위해, **Heterogeneous Graph**로 Design
- POI를 설명하는 Metadata를 잘 보존하기 위해, Local Structure를 보존할 수 있는 **Bipartite Graph**로 Design

### 2. Edge Design

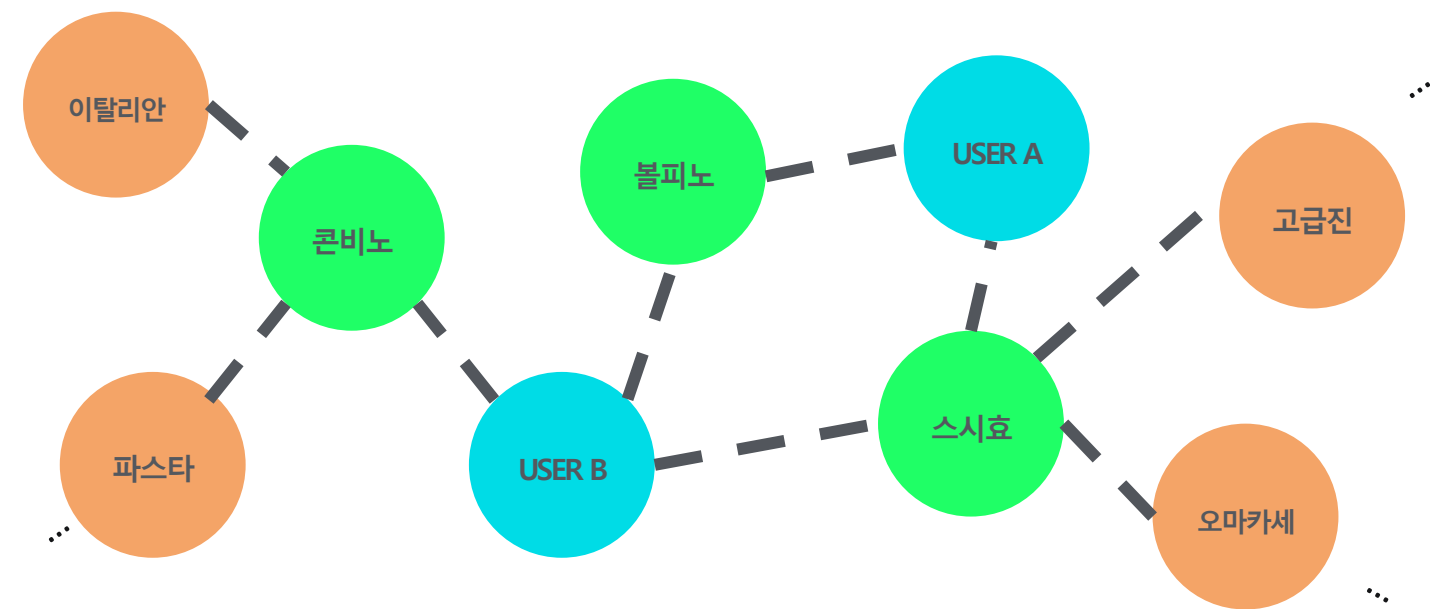
- 통계기반, NLP 기반 모델로 부터 추출된 Metadata를 Binary Edge로 Design

- **Heterogeneous Graph** = Node Type이 여러가지인 그래프
- **Bipartite Graph** = 모든 노드들이 두 그룹으로 나뉘지고 서로 다른 타입의 노드들이 edge으로 연결되어져 있는 그래프

# 2.3 POIMeta2vec\_Graph Design

2. POI와 Metadata간의 관계를 이용한 Heterogeneous Graph를 만들자.

**POI Knowledge Graph**

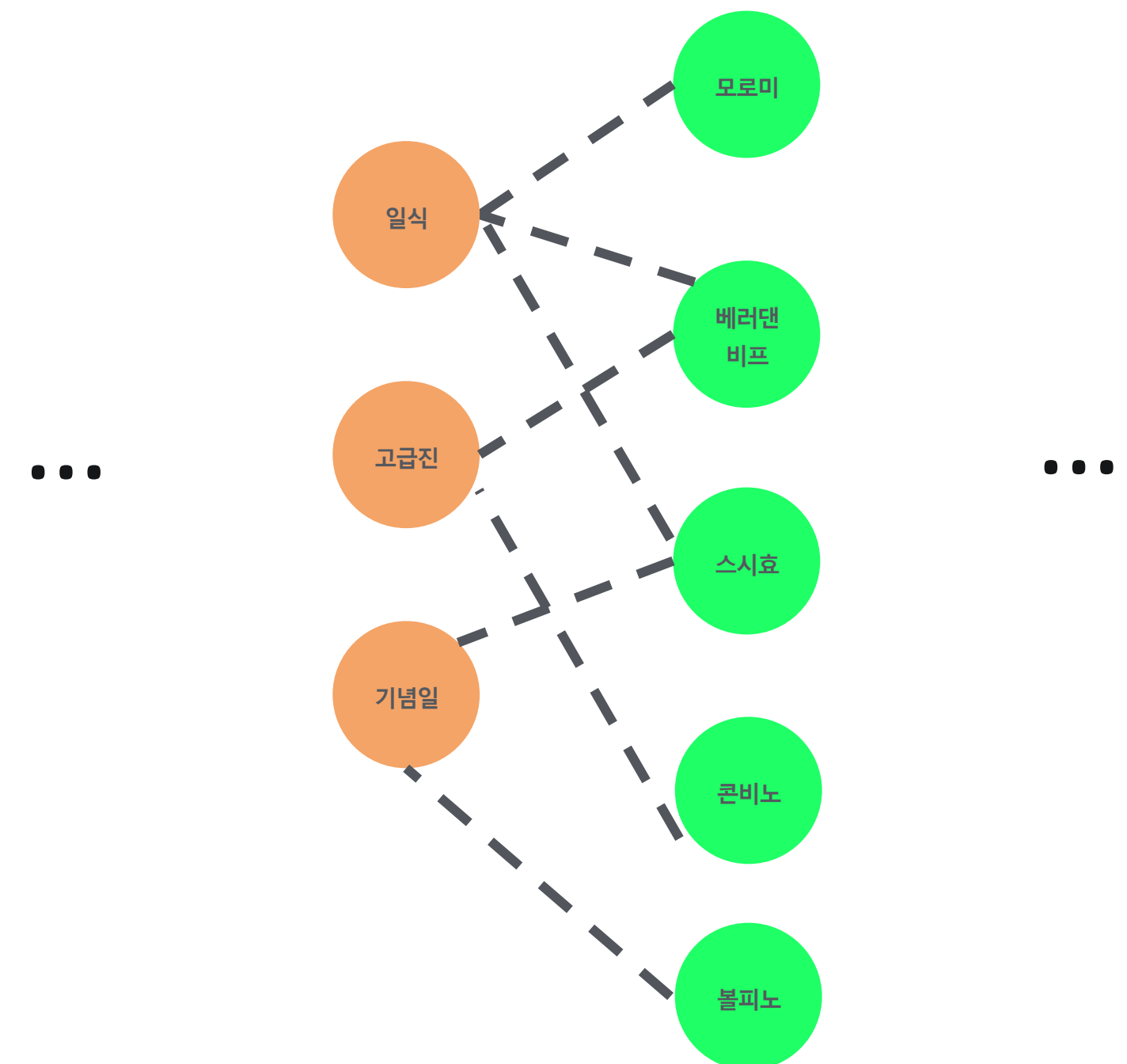


Projection POI, Metadata Node



Edge Relation by 업체 정보, ConA Model, 통계 모델

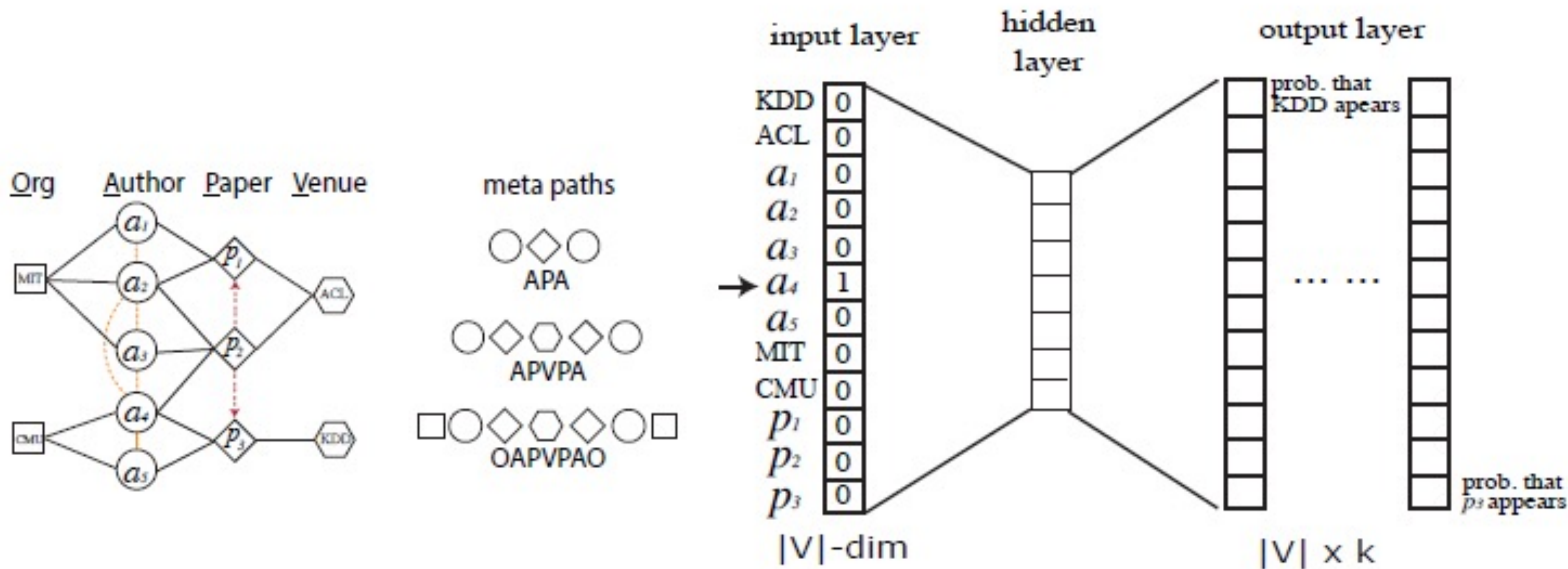
**POI-Metadata Graph**





# 2.3 POIMeta2vec\_Model Design

3. 공통된 Metadata를 가지는 POI끼리 가깝게 Embedding되게 하자.



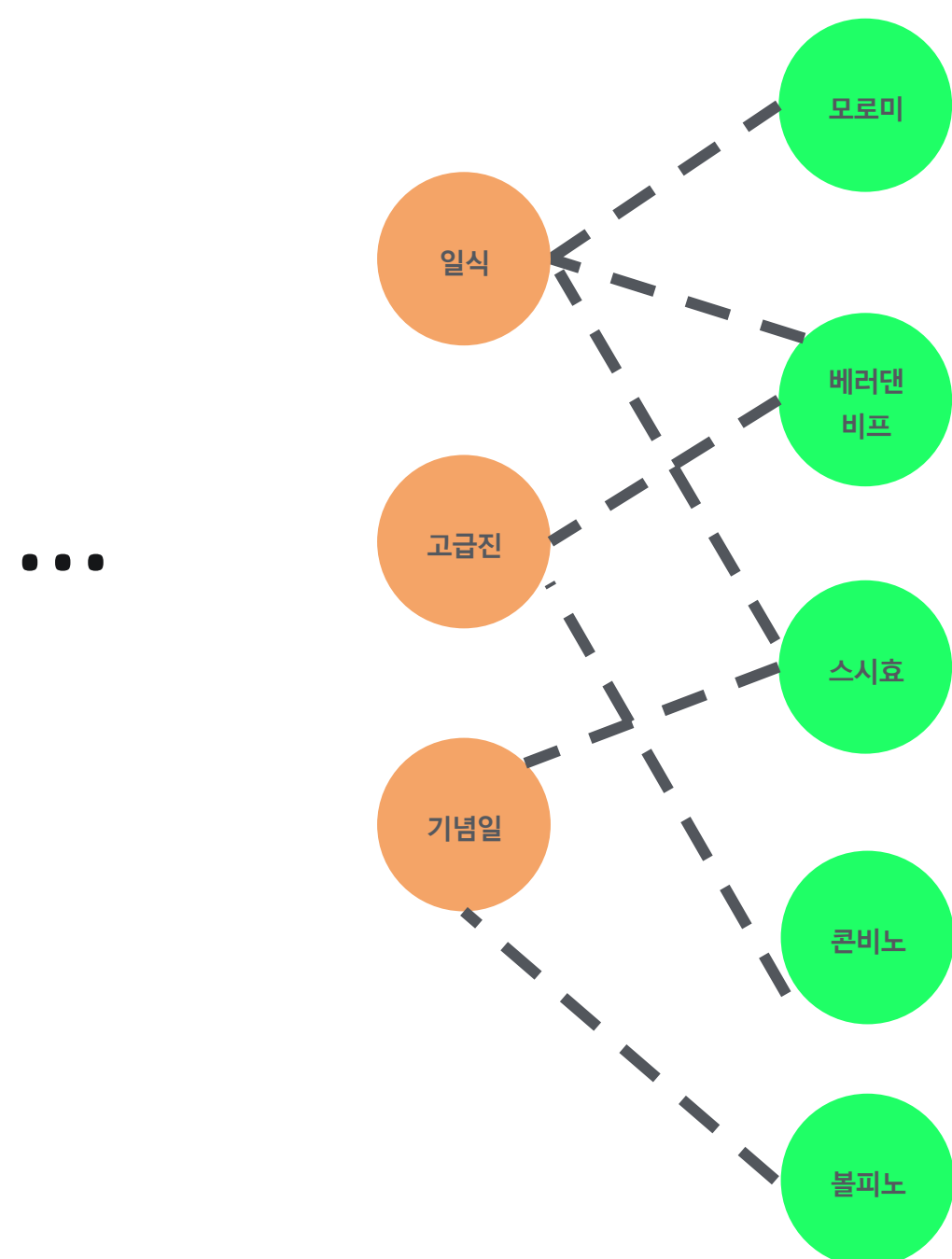
1. KnowledgeGraph Generation    2. Metapath Generation

3. Item Embedding(skipgram)

## 2.3 POIMeta2vec\_Model Design

3. 공통된 Metadata를 가지는 POI끼리 가깝게 Embedding되게 하자.

### POI-Metadata Graph



Metapath Generation



...

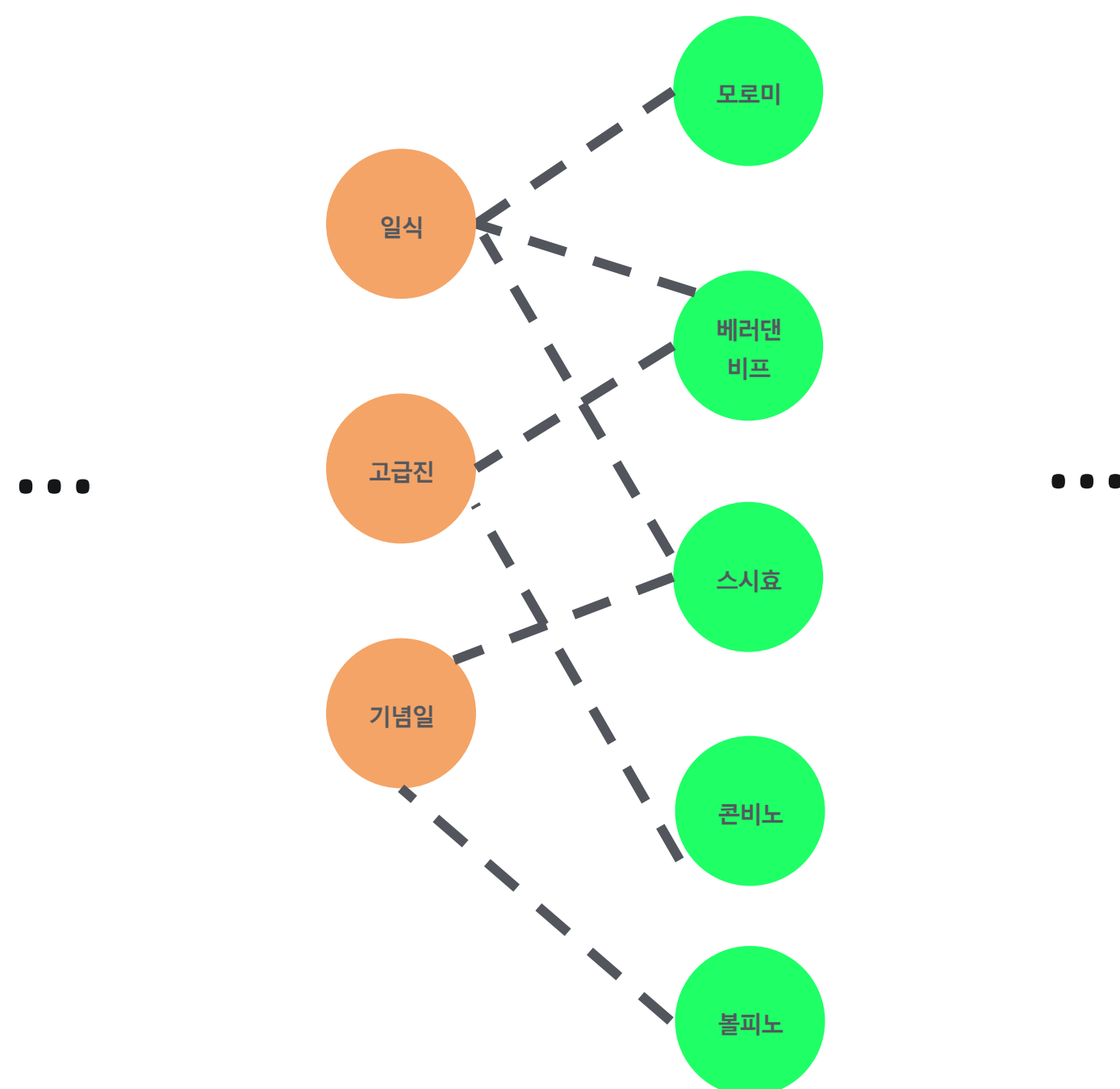
By Random Walk

**Metapath**

## 2.3 POIMeta2vec\_Model Design

3. 공통된 Metadata를 가지는 POI끼리 가깝게 Embedding되게 하자.

### POI-Metadata Graph



$$\text{Random walk}(V_i \rightarrow T_j) = \frac{P(V_i \rightarrow T_j)}{\sum_j P(V_i \rightarrow T_j)}$$

$$P(V_i \rightarrow T_j) = (0, 1) \begin{cases} 1, & \text{if } (V_i, T_j) \text{ is occurred sequentially even once,} \\ 0, & \text{else} \end{cases}$$

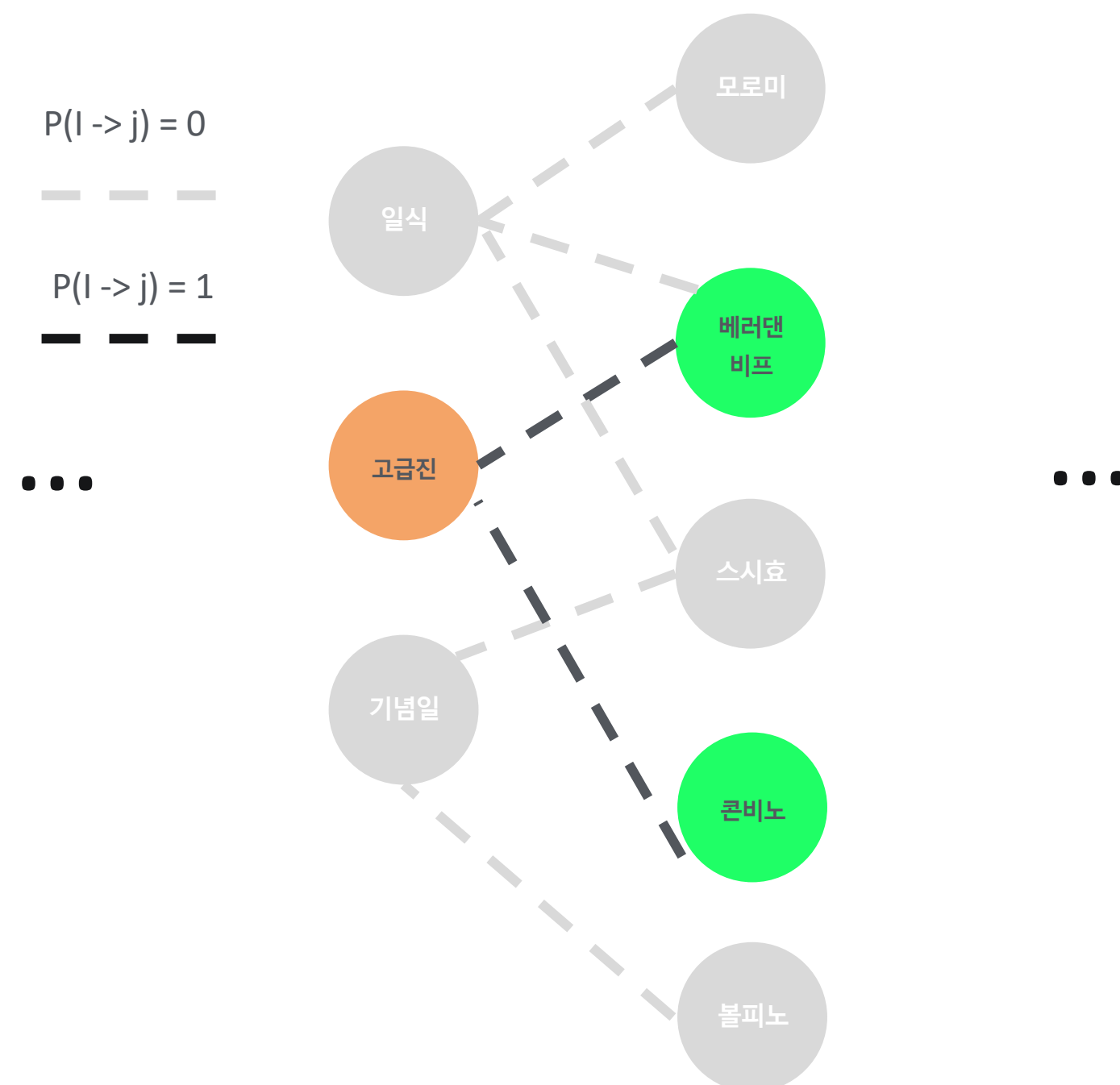
If  $(POI_i, POI_k)$  이 연속적으로 등장했고,  
둘 사이의 공통 metadata  $j$ 이 있는 경우,

Then  $P(V_i, T_j), P(V_j, T_k) = 1$

# 2.3 POIMeta2vec\_Model Design

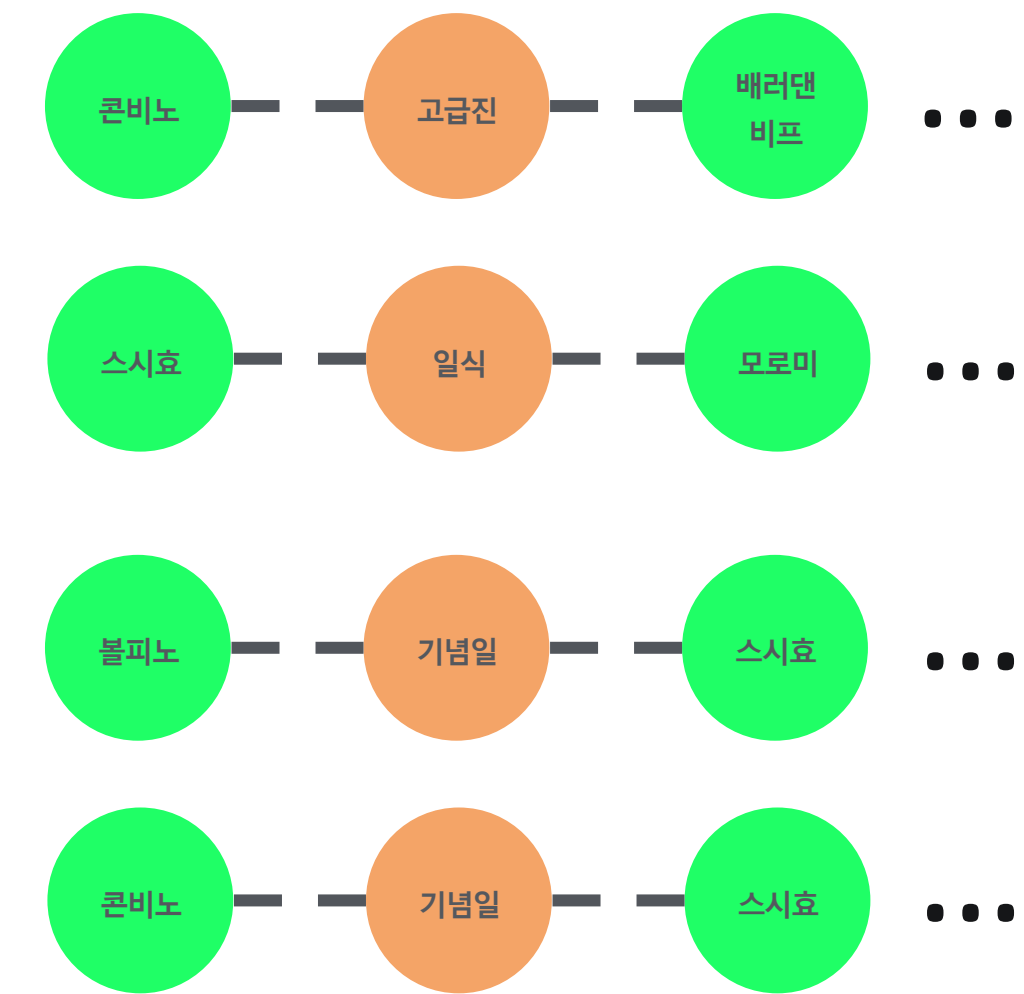
3. 공통된 Metadata를 가지는 POI끼리 가깝게 Embedding되게 하자.

POI-Metadata Graph



Metapath

Metapath Generation  
 →  
 Related by  
 취향 Data



취향 (Explicit Feedback) = 저장, 좋아요를 포함한 사용자 취향 데이터

# 2.3 POIMeta2vec\_Model Design

3. 공통된 Metadata를 가지는 POI끼리 가깝게 Embedding되게 하자.

## Metapath



$$\theta = \operatorname{argmax}_{\theta} \sum_{(w,c) \in V_{pos}} \frac{1}{1 + \exp(-u_w^T v_c)} + \sum_{(w,c) \in V_{neg}} \frac{1}{1 + \exp(u_w^T v_c)}$$

*Sentence = Metapath*

*Center word(c) = Metapath 내의 모든 Node*

*Around word(w) = Center word 앞 뒤의 window size 범위내의 Node*

## 2.4 POISAGE

**Problem Design**

1. 나와 유사한 사용자의 취향을 반영하여 추천 할 수 없을까?

**Graph Design**

2. USER의 선호 POI를 고려한 Heterogeneous Graph를 만들자.

**Model Design**

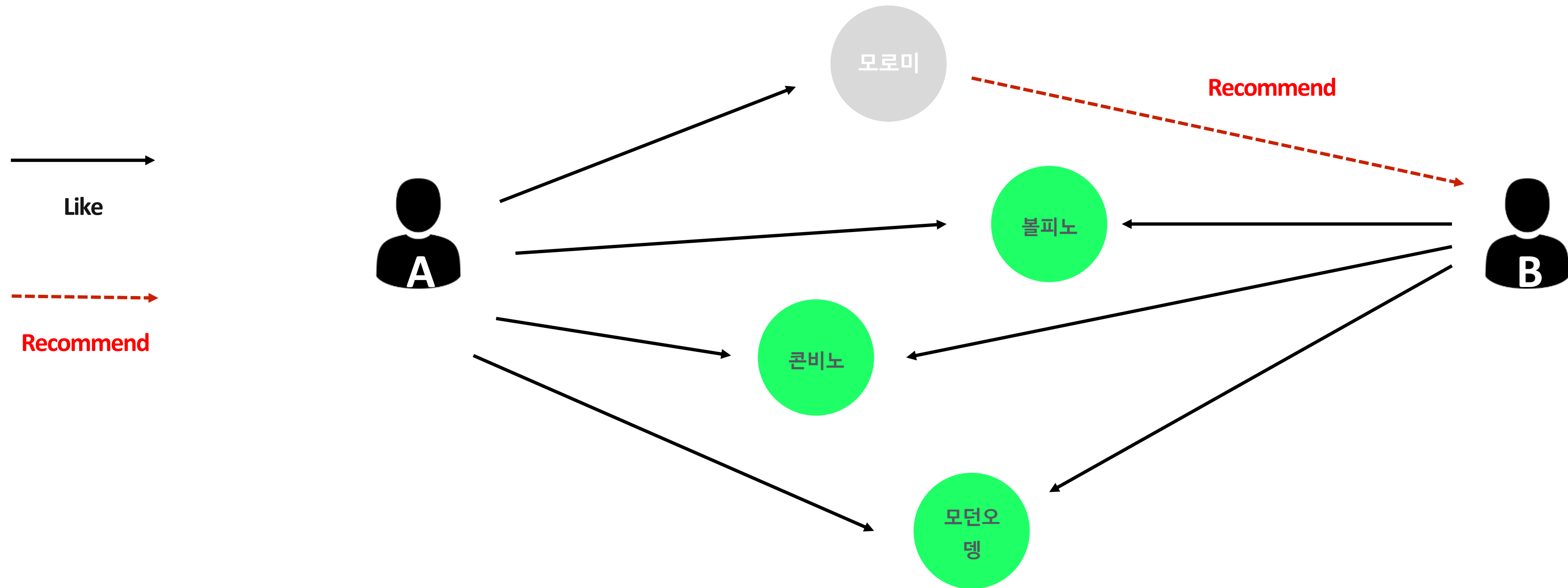
3. Collaborative Filtering이 반영되도록 POI Embedding을 하자.

**POI Embedding**

# 2.4 POISAGE\_Problem Design

1. 나와 유사한 사용자의 취향을 반영하여 추천 할 수 없을까?

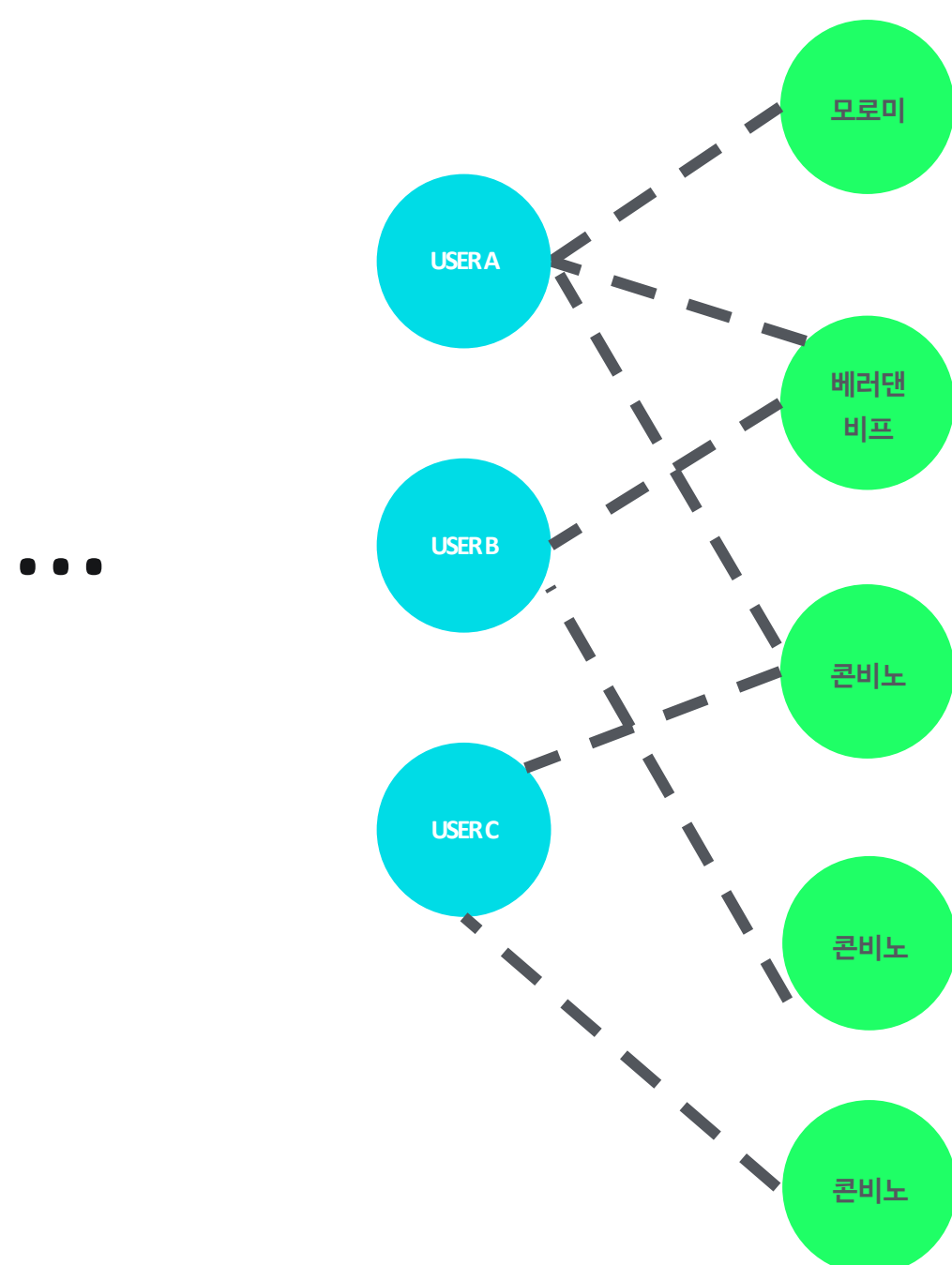
## Collaborative Filtering



# 2.4 POISAGE\_Graph Design

2. USER의 선호 POI를 고려한 Heterogeneous Graph를 만들자.

## POI-USER Graph



## Graph Design

### 1. Node Design

- USER가 선호하는 POI를 고려하기 위해, **Heterogeneous Graph**로 Design
- Collaborative Filtering 성격을 잘 보존하기 위해, Local Structure를 보 존할 수

... .. 있는 **Bipartite Graph**로 Design

### 2. Edge Design

- USER가 취향으로 담은 POI를 Edge로 Design

*취향 = 저장, 좋아요를 포함한 사용자 취향 데이터*

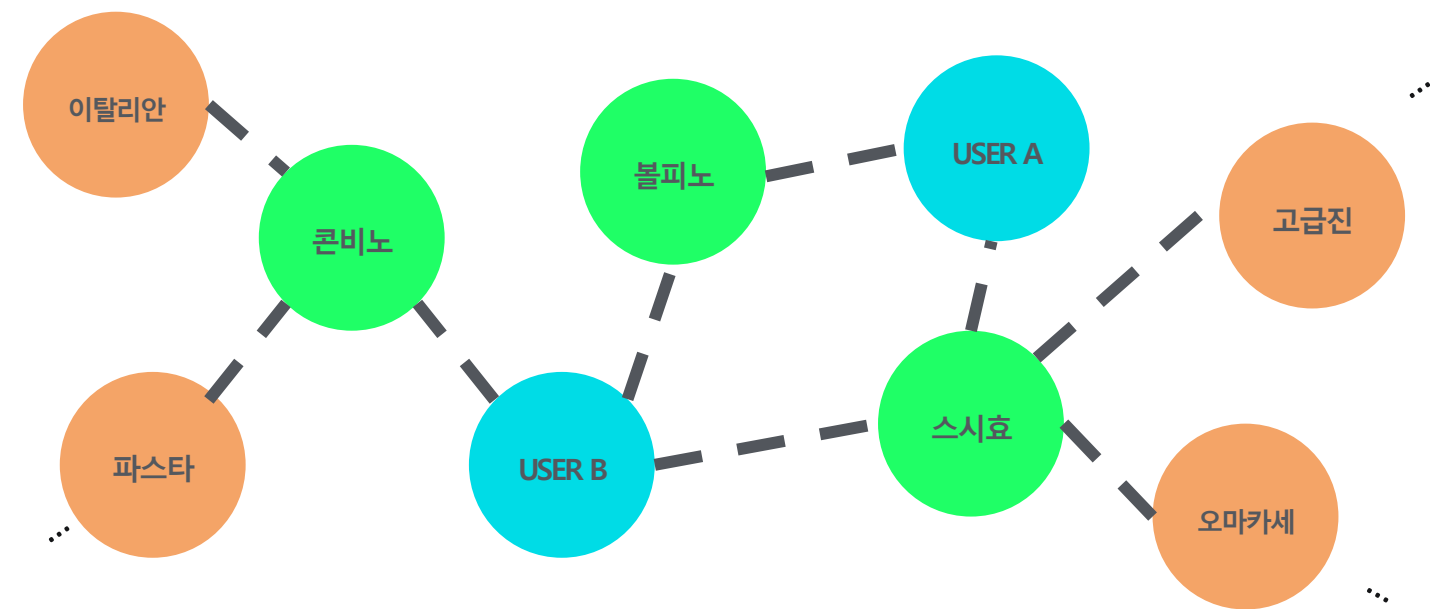
- **Heterogeneous Graph** = Node Type이 여러가지인 그래프
- **Bipartite Graph** = 모든 노드들이 두 그룹으로 나뉘지고 서로 다른 타입의 노드들이 edge으로 연결되어져 있는 그래프



# 2.4 POISAGE\_Graph Design

2. USER의 선호 POI를 고려한 Heterogeneous Graph를 만들자.

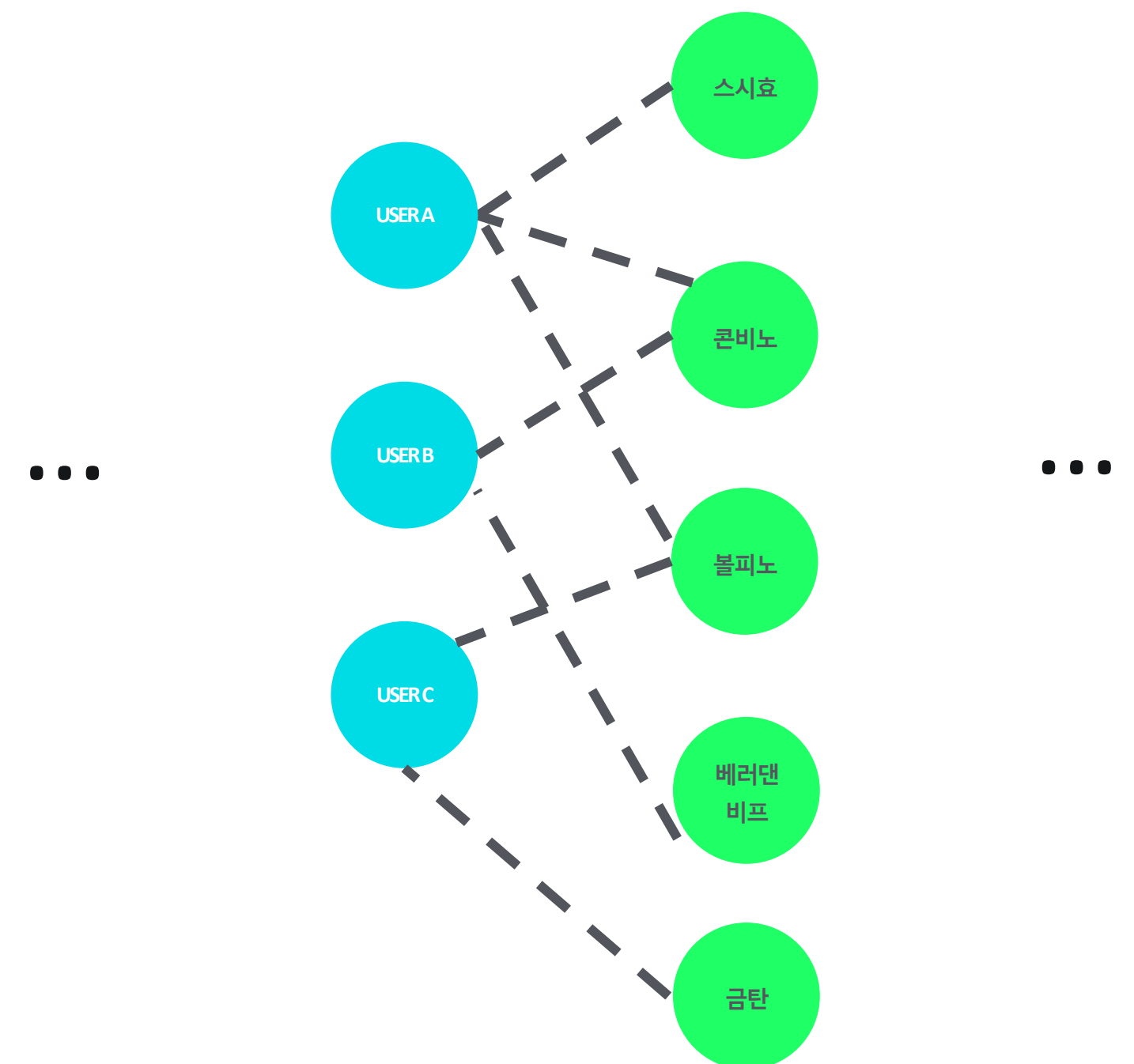
**POI Knowledge Graph**



Projection POI, USER Node

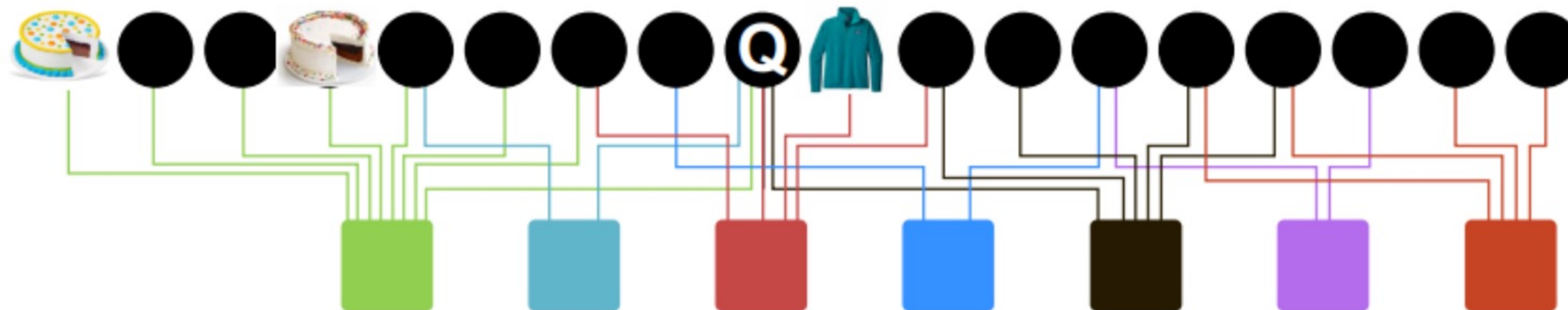
Related by  
취향 Data

**POI-USER Graph**



## 2.4 POISAGE\_Model Design

3. Collaborative Filtering이 반영되도록 POI Embedding을 하자.



### PinSAGE Key Point

- Heterogeneous Graph, Bipartite Graph 기반의 Graph Model
- 다양한 Node Feature의 활용
- Domain Knowledge에 기반한 Sampling 방식

## 2.4 POISAGE\_Model Design

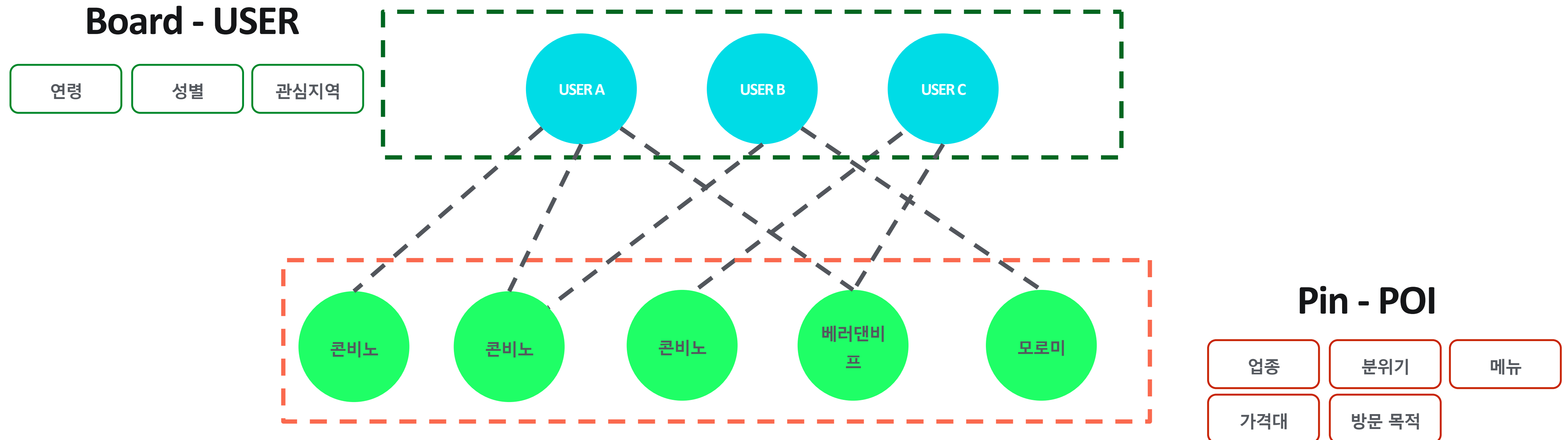
3. Collaborative Filtering이 반영되도록 POI Embedding을 하자.

1. Pin과 Board는 어떻게 정의 해야할까?
2. Local Domain에서 Negative Sampling은 어떻게 해야할까?
3. USER의 POI에 대한 선호도 차이를 고려할 수 없을까?
4. 어떻게 POI의 특성을 조금 더 잘 보존할 수 있을까?

# 2.4 POISAGE\_Model Design

3. Collaborative Filtering이 반영되도록 POI Embedding을 하자.

## 1. Pin과 Board는 어떻게 정의 해야할까?



# 2.4 POISAGE\_Model Design

3. Collaborative Filtering이 반영되도록 POI Embedding을 하자.

## 2. Local Domain에서 Negative Sampling은 어떻게 해야할까?



**한외담 한남본점** 소고기구이

★ 4.26/5 · 방문자리뷰 279 · 블로그리뷰 436

한남동에서 맛있는 한우를 먹을 수 있는 고기집

#모던한 #치즈구이 #핫플레이스 ...



업종이 같음

Metadata는 **다름**



**한우마을** 소고기구이

★ 4.49/5 · 방문자리뷰 321 · 블로그리뷰 330

알 사람만 안다는 마장동 한우 맛집

#정육식당 #착한가격 #꽃등심 ...

# 2.4 POISAGE\_Model Design

3. Collaborative Filtering이 반영되도록 POI Embedding을 하자.

## 3. USER의 POI에 대한 선호도 차이를 고려할 수 없을까?

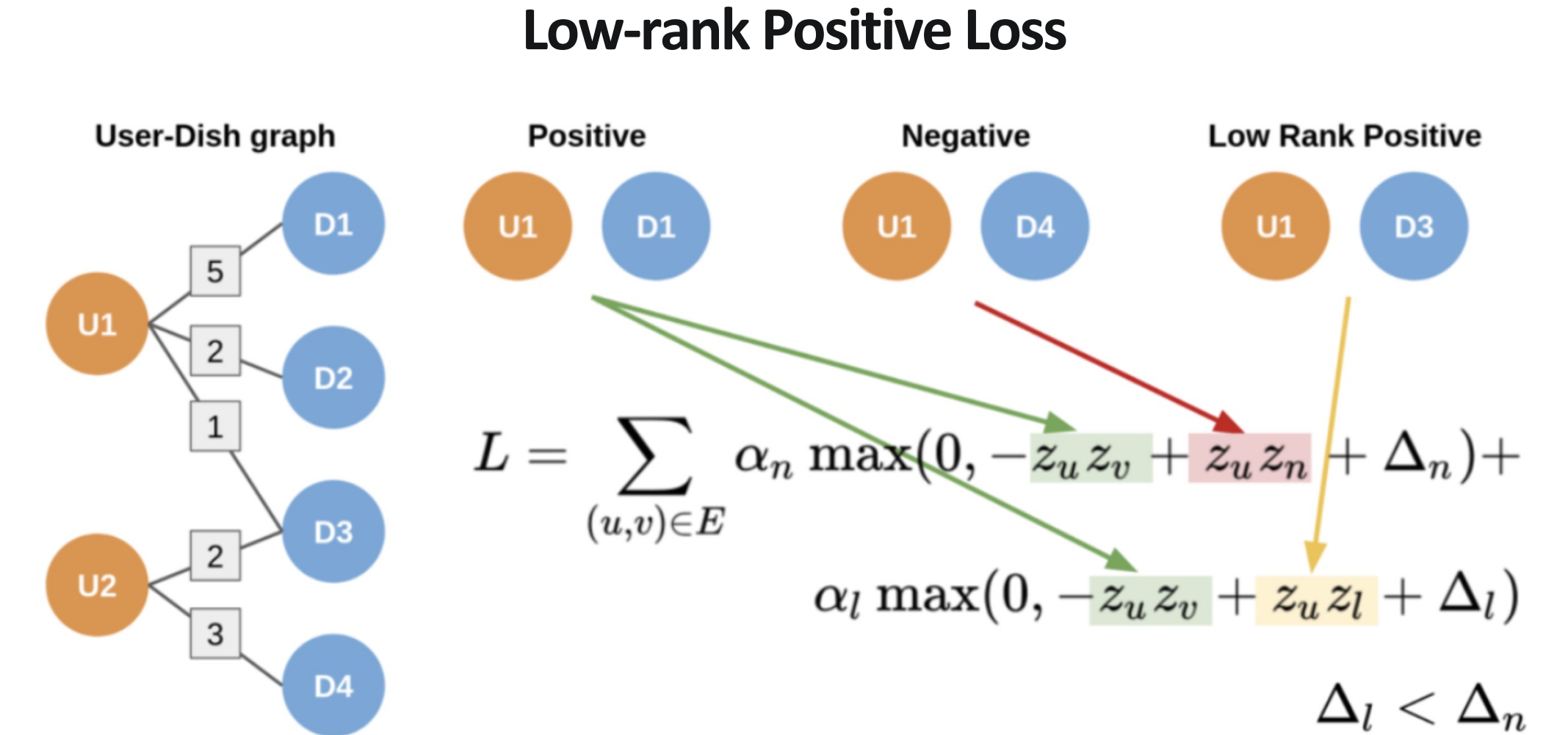
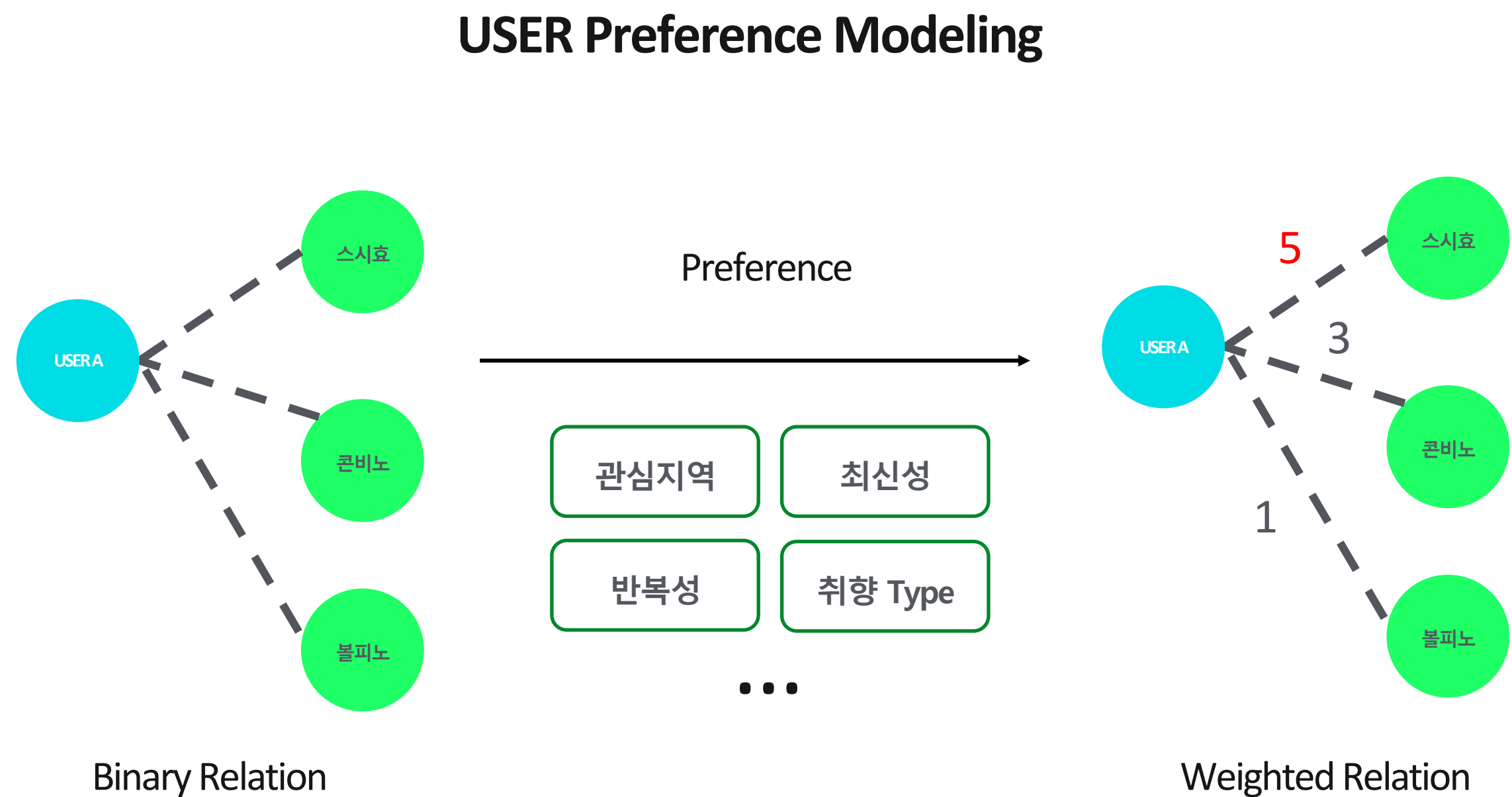


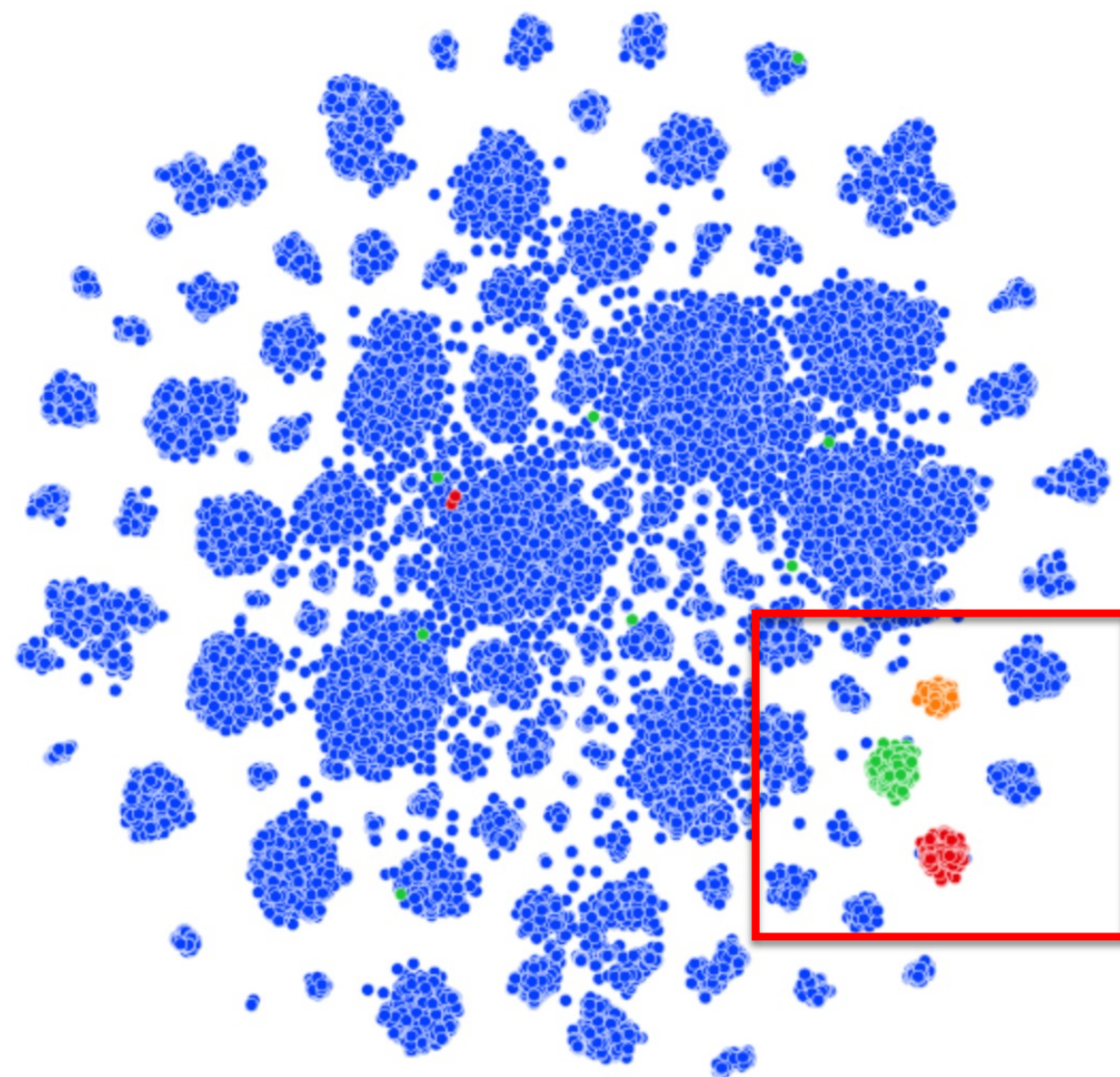
Figure 4: Our Uber Eats recommendation system leverages max-margin loss augmented with low rank positives.

# 2.4 POISAGE\_Model Design

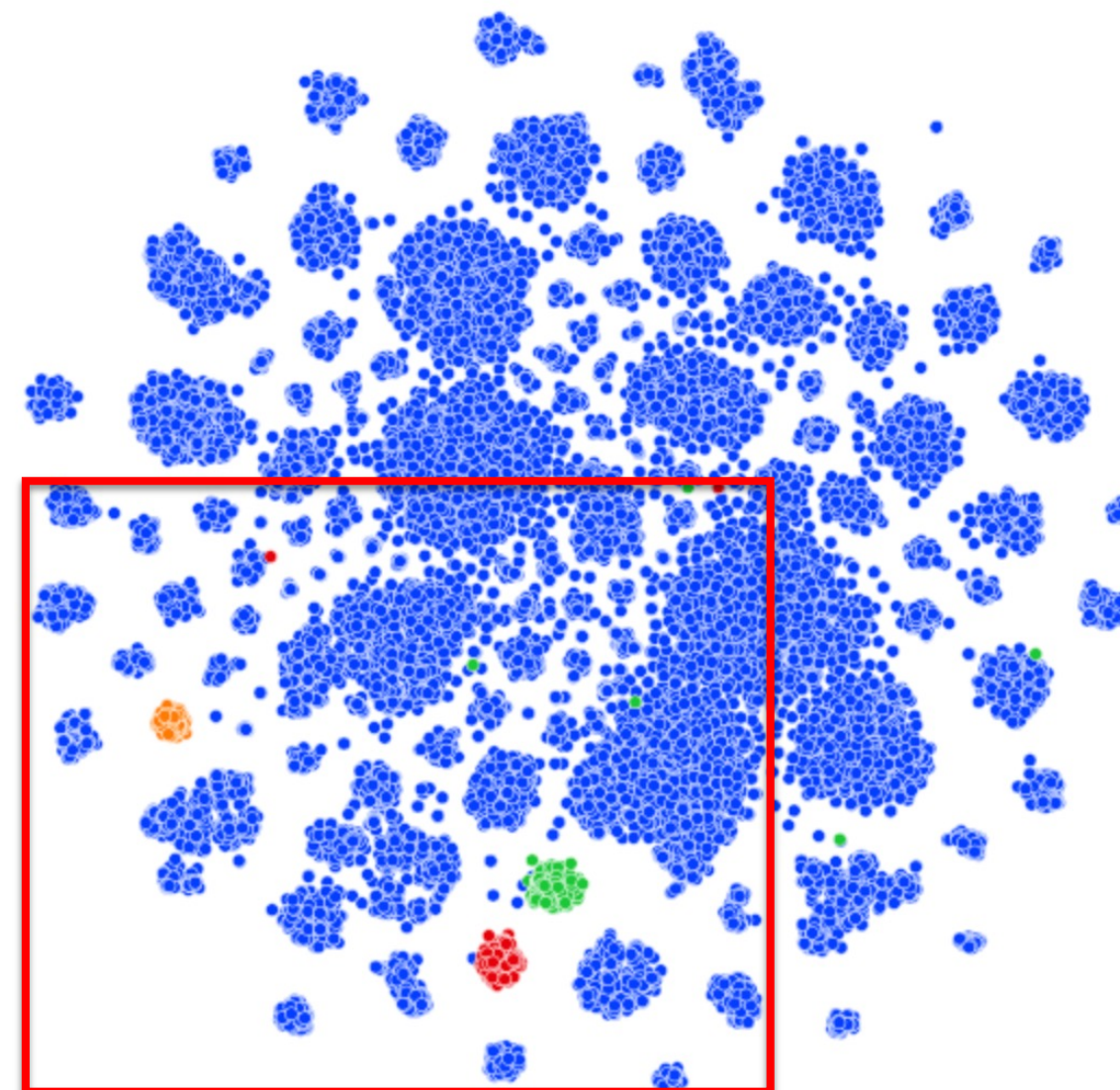
3. Collaborative Filtering이 반영되도록 POI Embedding을 하자.

## 4. 어떻게 POI의 특성을 조금 더 잘 보존할 수 있을까? (Layer 수 측면)

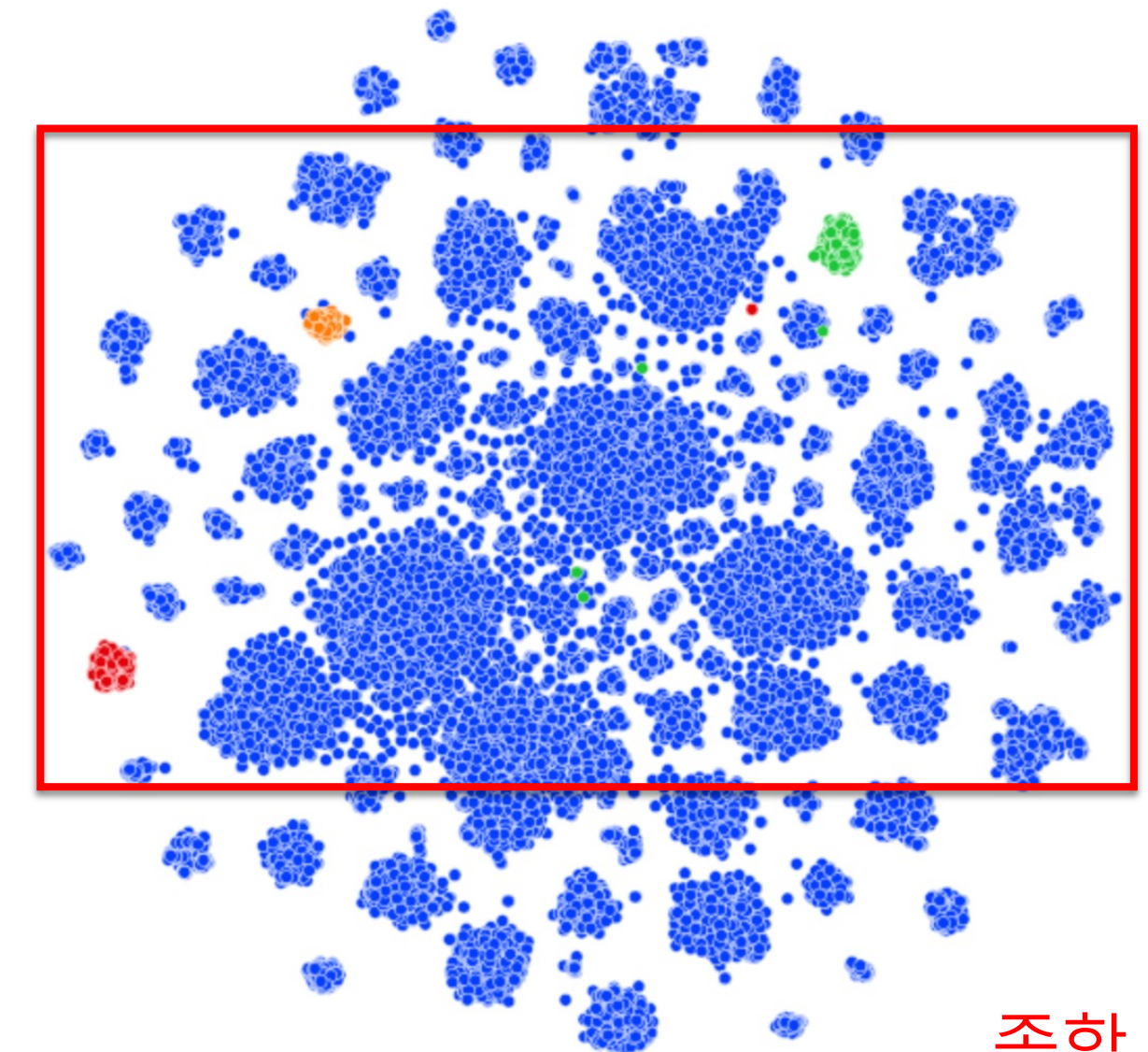
D=2, POISAGE



D=5, POISAGE



D=10, POISAGE



종합 분식  
떡볶이  
김밥

Over Smoothing Effect



## 2.4 POISAGE\_Model Design

3. Collaborative Filtering이 반영되도록 POI Embedding을 하자.

### 1. Pin과 Board는 어떻게 정의 해야할까?

- Pin = POI(업종, 가격대, 분위기, 메뉴, 목적), Board = USER(연령, 성별, 관심지역)

### 2. Local Domain에서 Negative Sampling은 어떻게 해야할까?

- 업종은 같지만 Metadata의 Jaccard Distance가 먼 것은 Negative로 가정

### 3. USER의 POI에 대한 선호도 차이를 고려할 수 없을까?

- Low Rank Positive Loss를 활용

### 4. 어떻게 POI의 특성을 조금 더 잘 보존할 수 있을까?

- Over Smoothing되지 않는 범위 내에서 layer 수( $k = 2$ )를 설정



# 2.5 Summary of POI Embedding

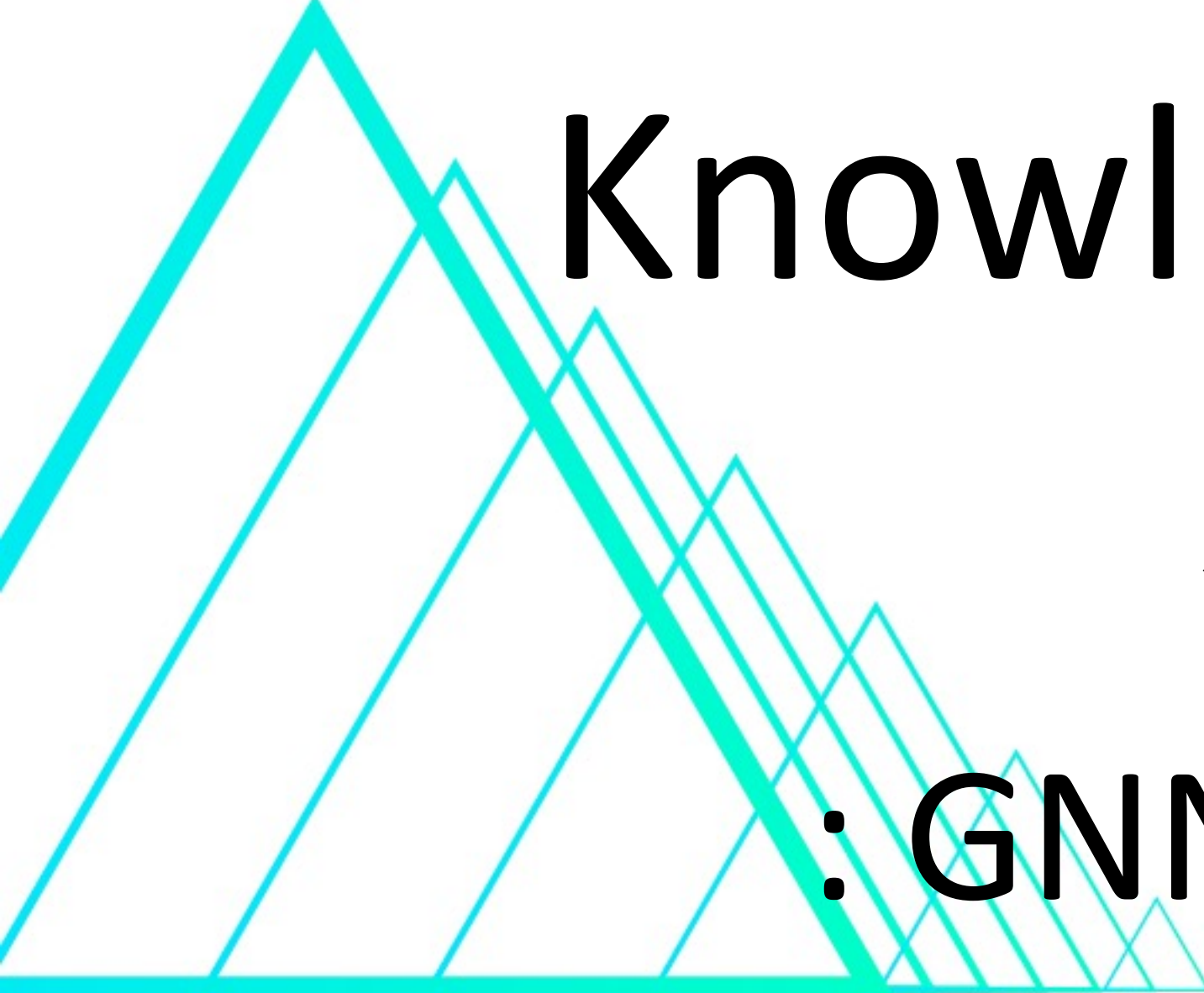
	POI2VEC	POIMeta2vec	POISAGE
Problem Design	동일 탐색 의도를 가진 POI	POI Metadata가 유사한 POI	유사한 사용자가 좋아할만한 POI
Graph Design	POI Homogeneous Graph	POI-Metadata Heterogeneous Graph	POI-USER Heterogeneous Graph
Model Design	Word2vec 기반 모델	Metapath2vec 기반 모델	PinSAGE 기반 모델

# 3. POI Embedding이 잘 되었다는 것은 어떻게 평가할 수 있을까?




Knowledge Graph에게 맛집과

사용자를 묻는다.



: GNN으로 맛집 취향 저장하기!



# 3.1 Overview



**Context  
Visualization**

각 모델들의 목적을  
잘 녹여냈을까?



**Precision**

어떤 상황에서  
얼마나 정확하게 추천할 수 있을까?



**Value of  
Personalization**

얼마나 개인화의 가치를 잘 담아냈을까?

## 3.2 Visualization



**Context  
Visualization**

각 모델들의 목적을  
잘 녹여냈을까?



**Precision**

어떤 상황에서  
얼마나 정확하게 추천할 수 있을까?



**Value of  
Personalization**

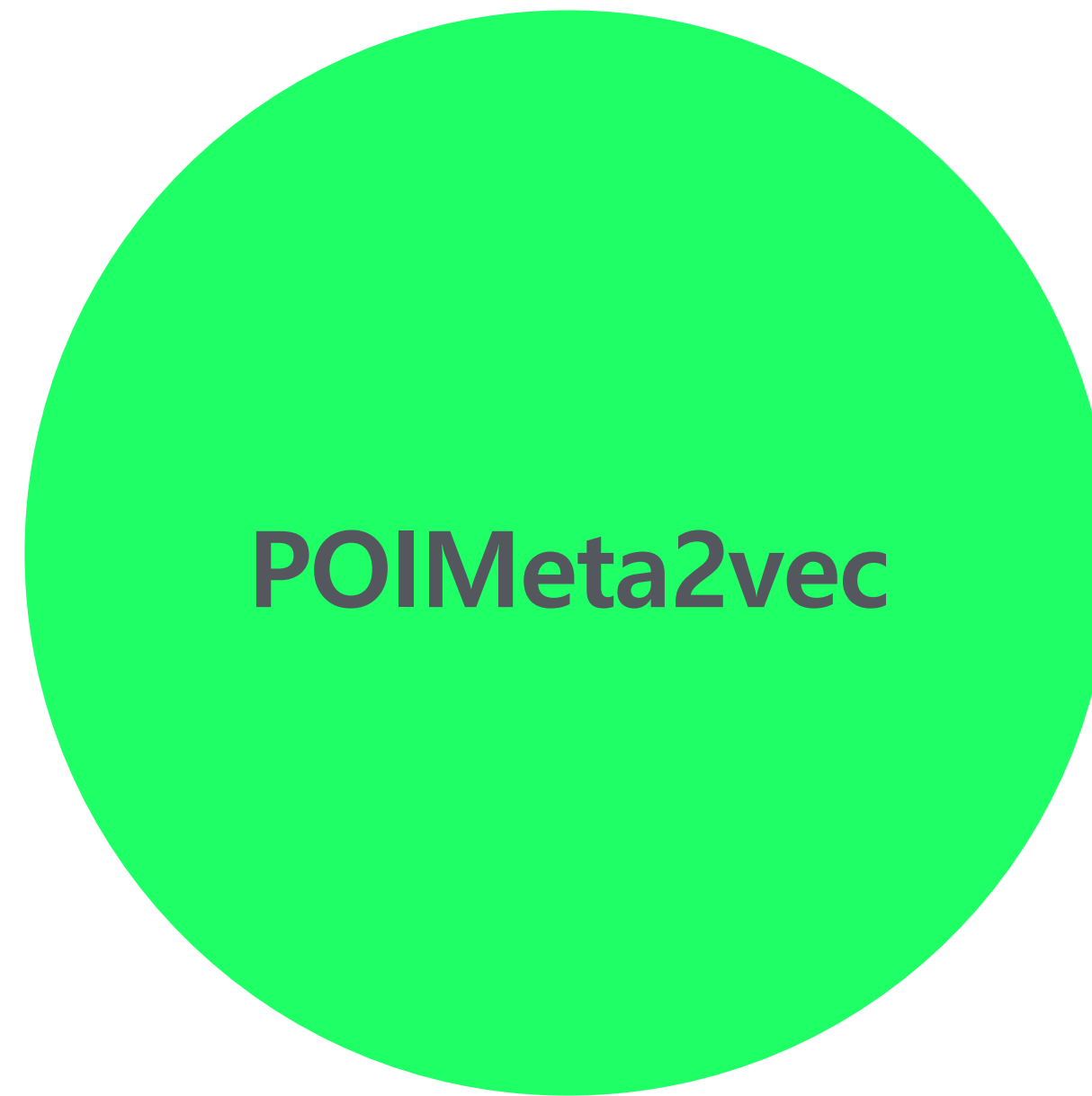
얼마나 개인화의 가치를 잘 담아냈을까?

## 3.2 Visualization: Preview



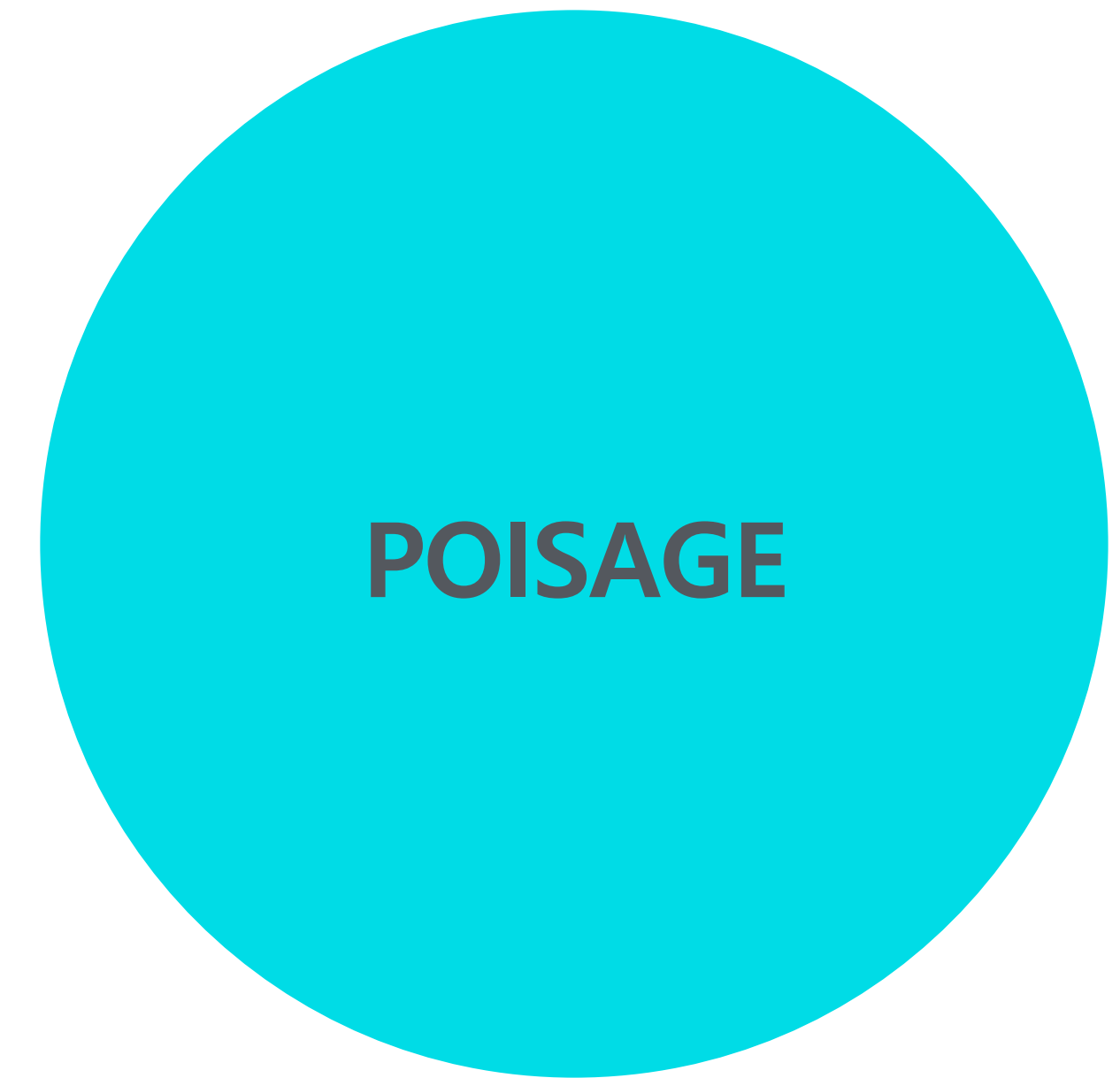
**POI2VEC**

근처, 동일 탐색 의도를 가진  
POI끼리 유사하도록



**POIMeta2vec**

같은 Metadata를 가지는  
POI끼리 유사하도록

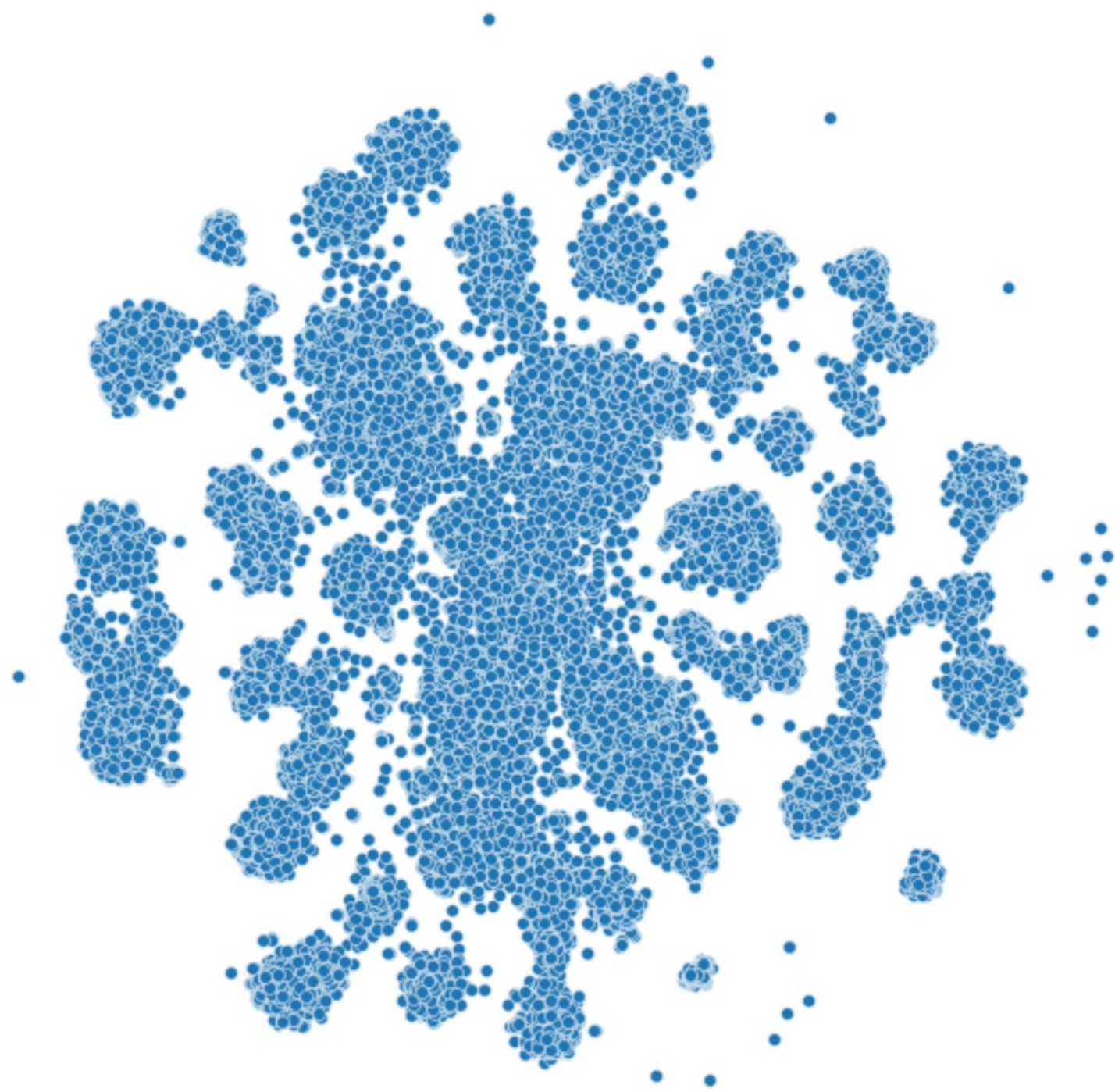


**POISAGE**

유사한 성향의 유저들이 방문한  
POI끼리 유사하도록

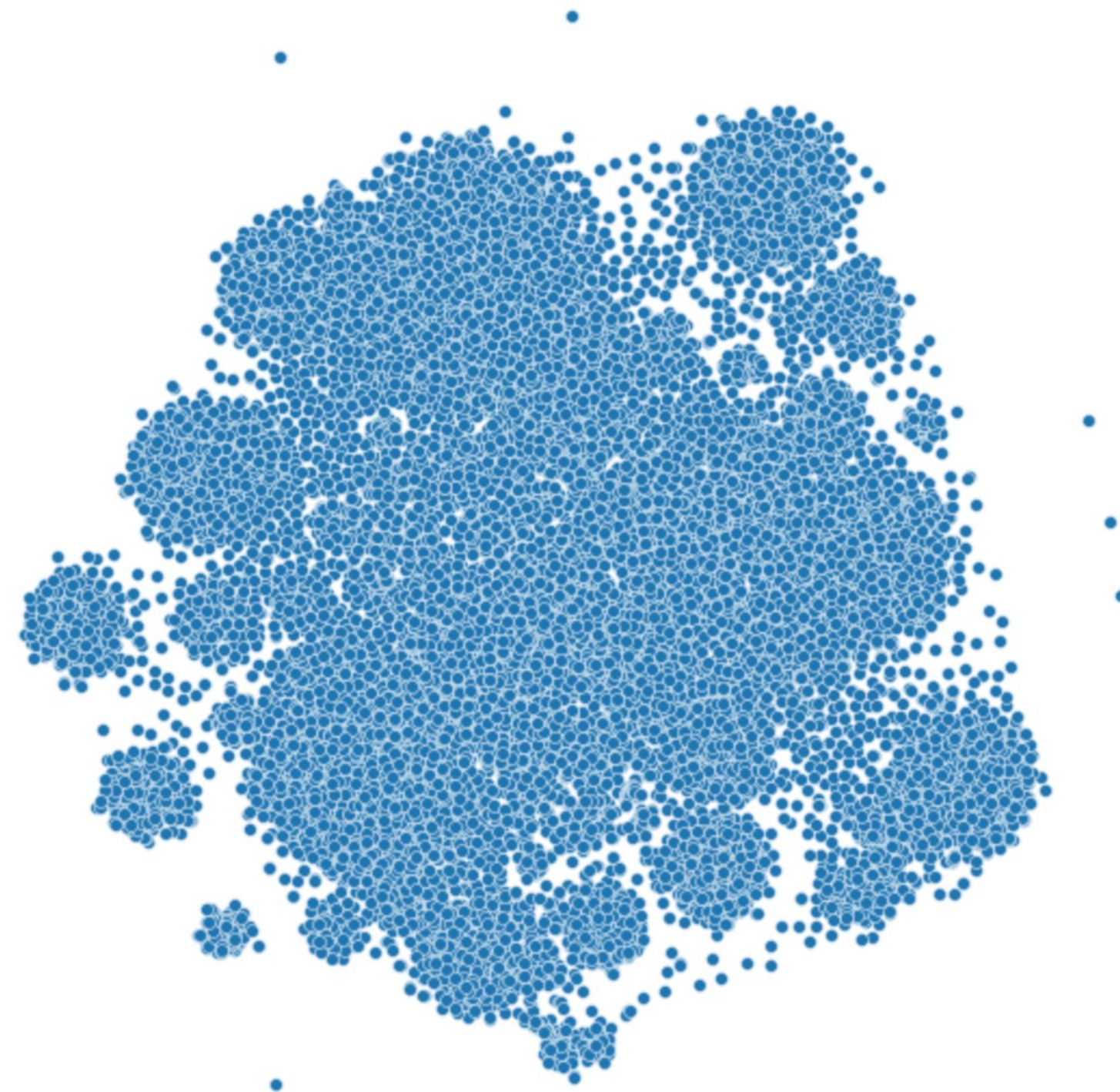
# 3.2 Visualization: Overview

**POI2VEC**



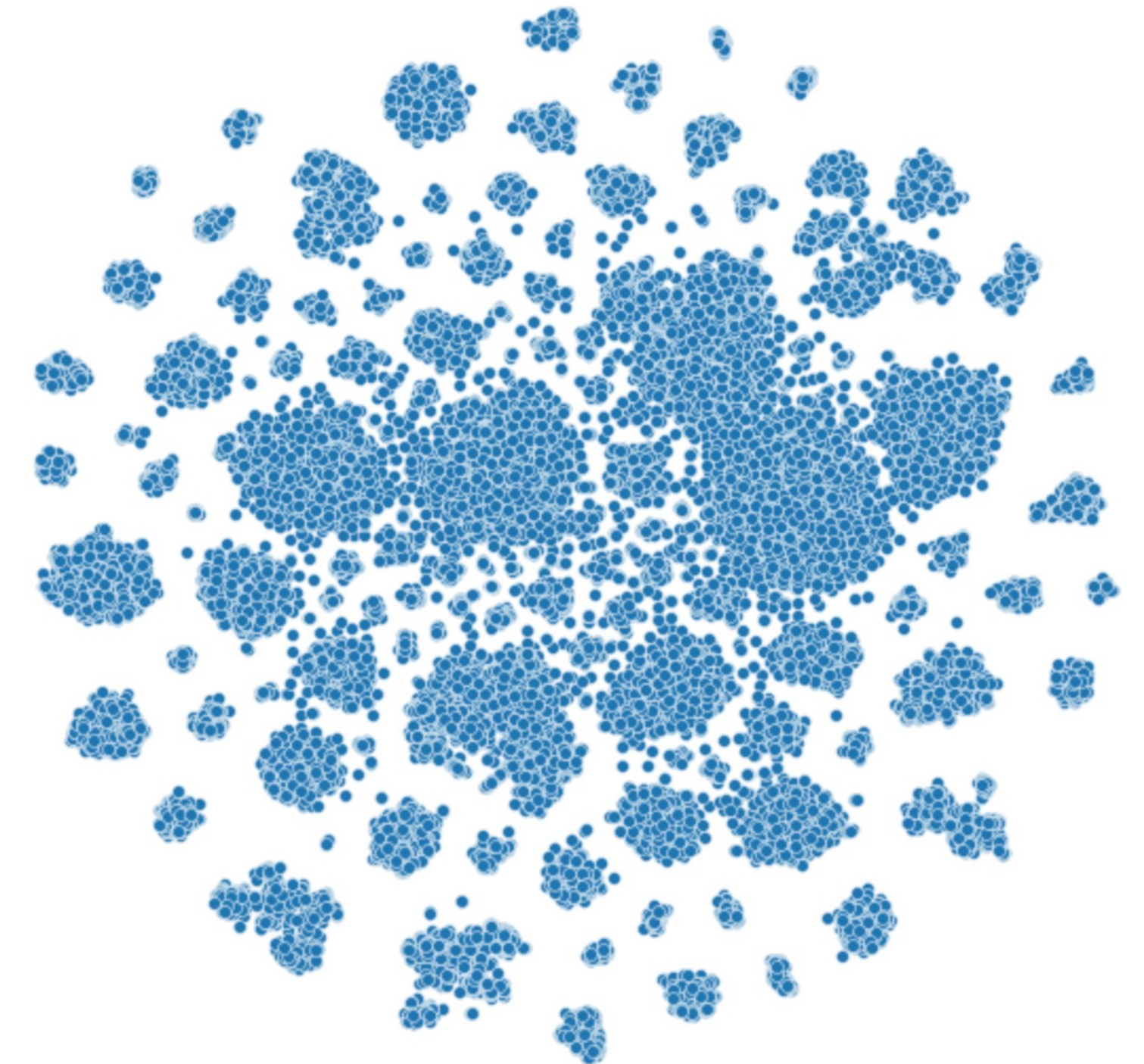
근처, 동일 탐색 의도를 가진  
POI끼리 유사하도록

**POIMeta2vec**



같은 Metadata를 가지는  
POI끼리 유사하도록

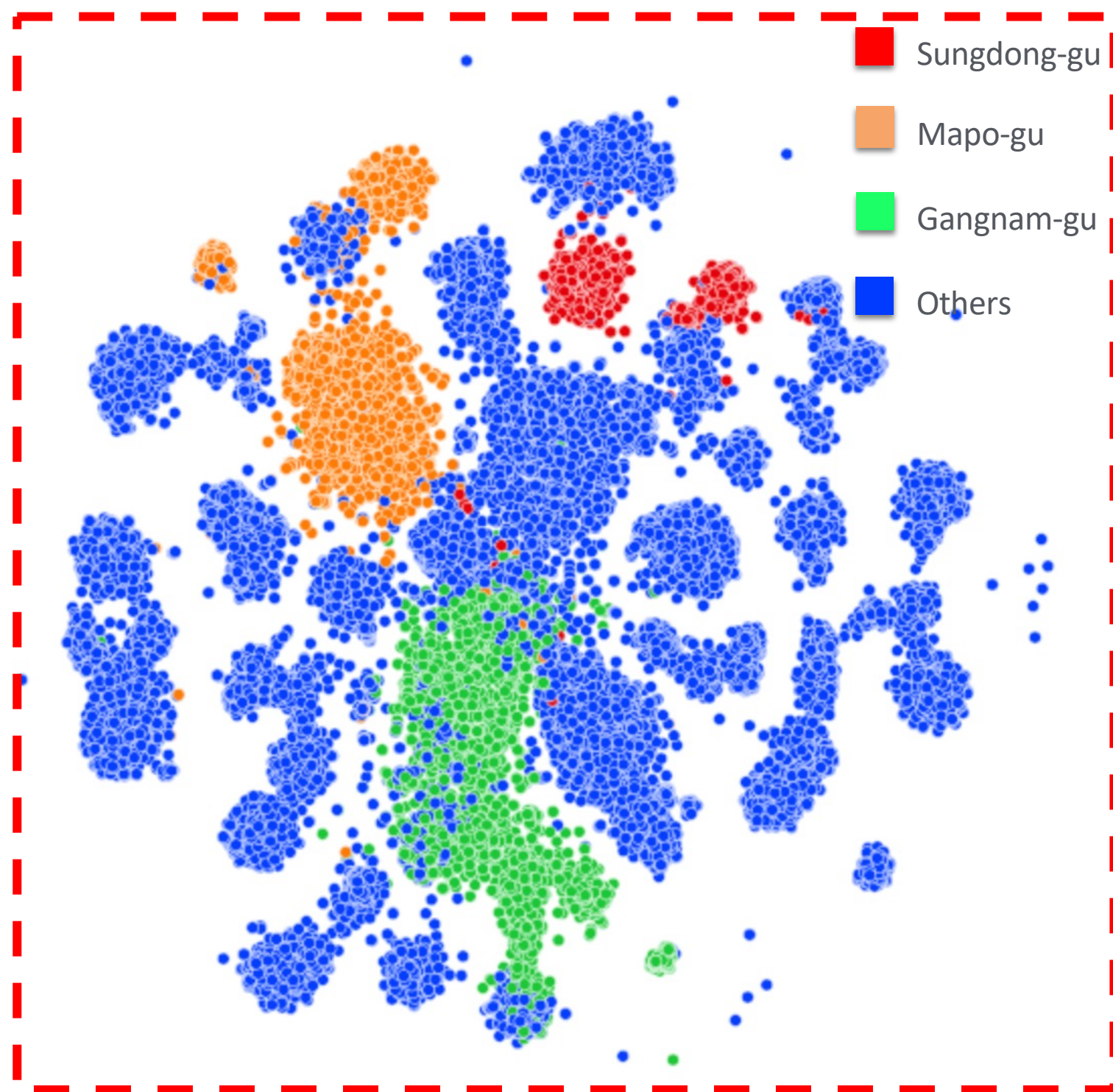
**POISAGE**



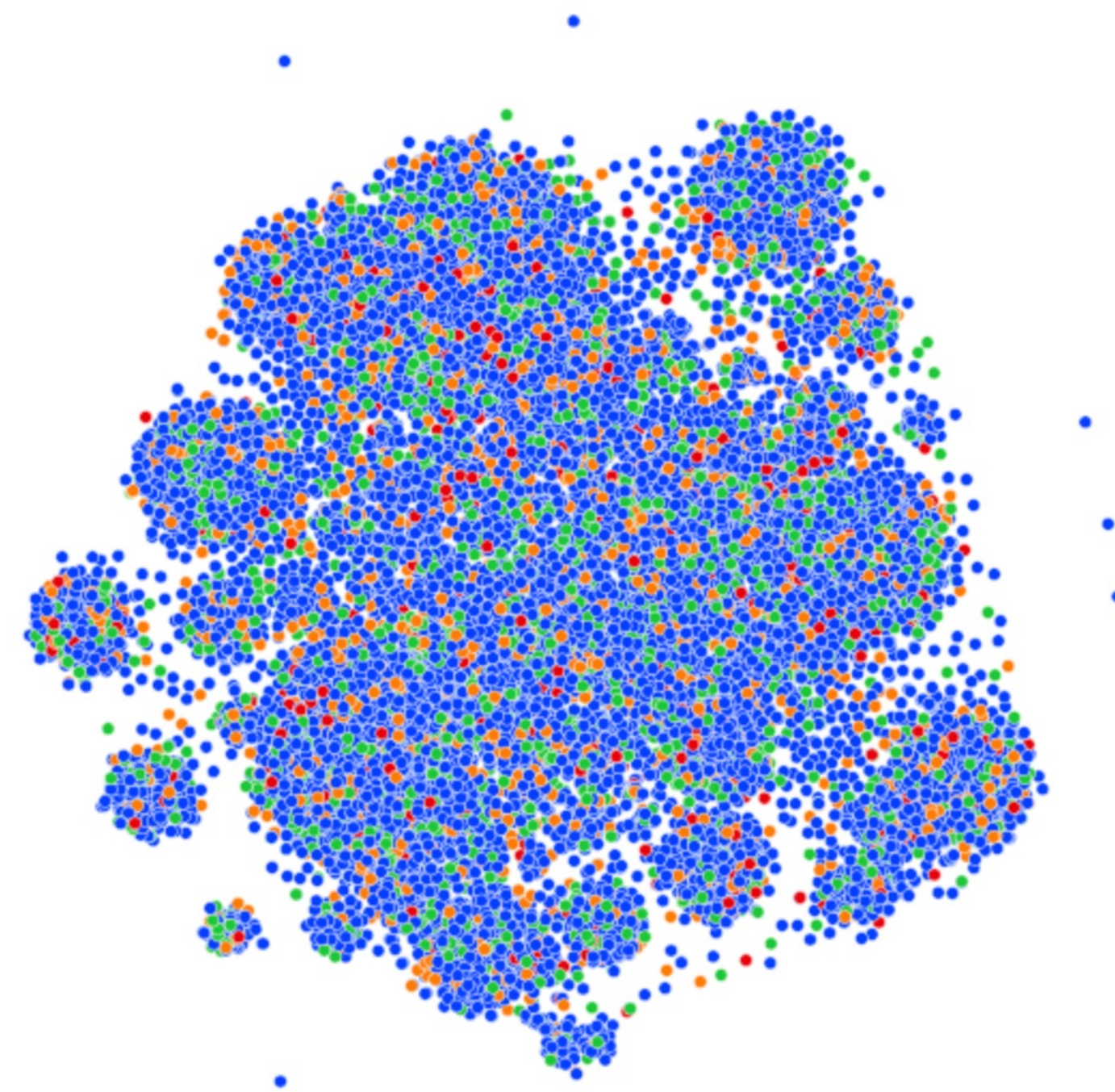
유사한 성향을 보이는 유저들이 방문한  
POI끼리 유사하도록

# 3.2 Visualization: Details

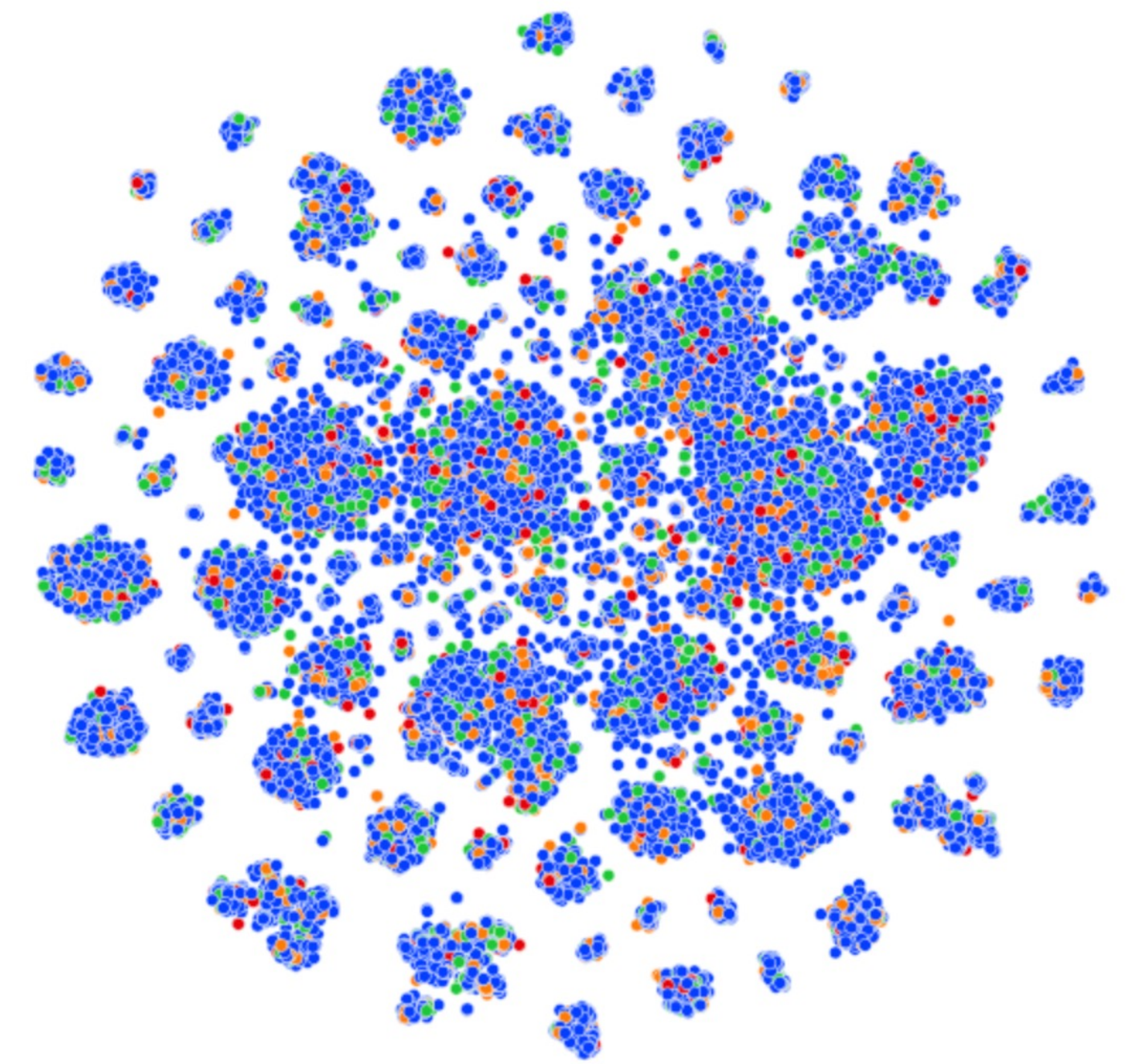
같은 지역 (e.g. 강남구, 성동구, 마포구) 끼리 뭉쳐있는 것이 아닐까?



**POI2VEC**



**POIMeta2vec**

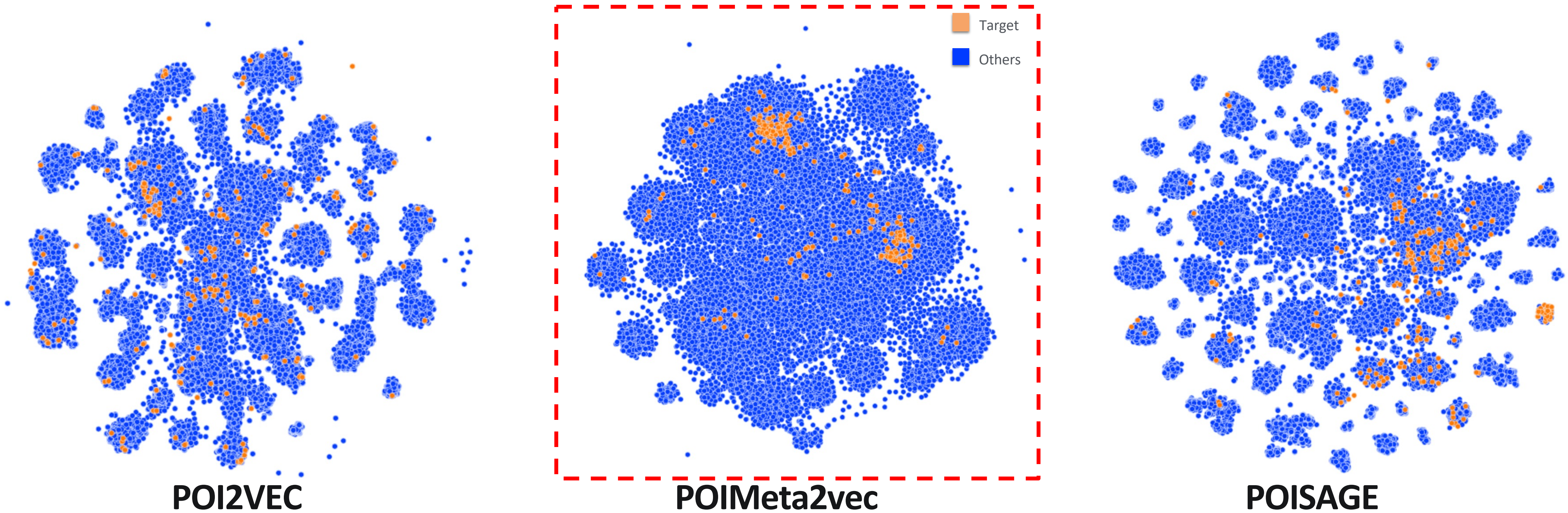


**POISAGE**



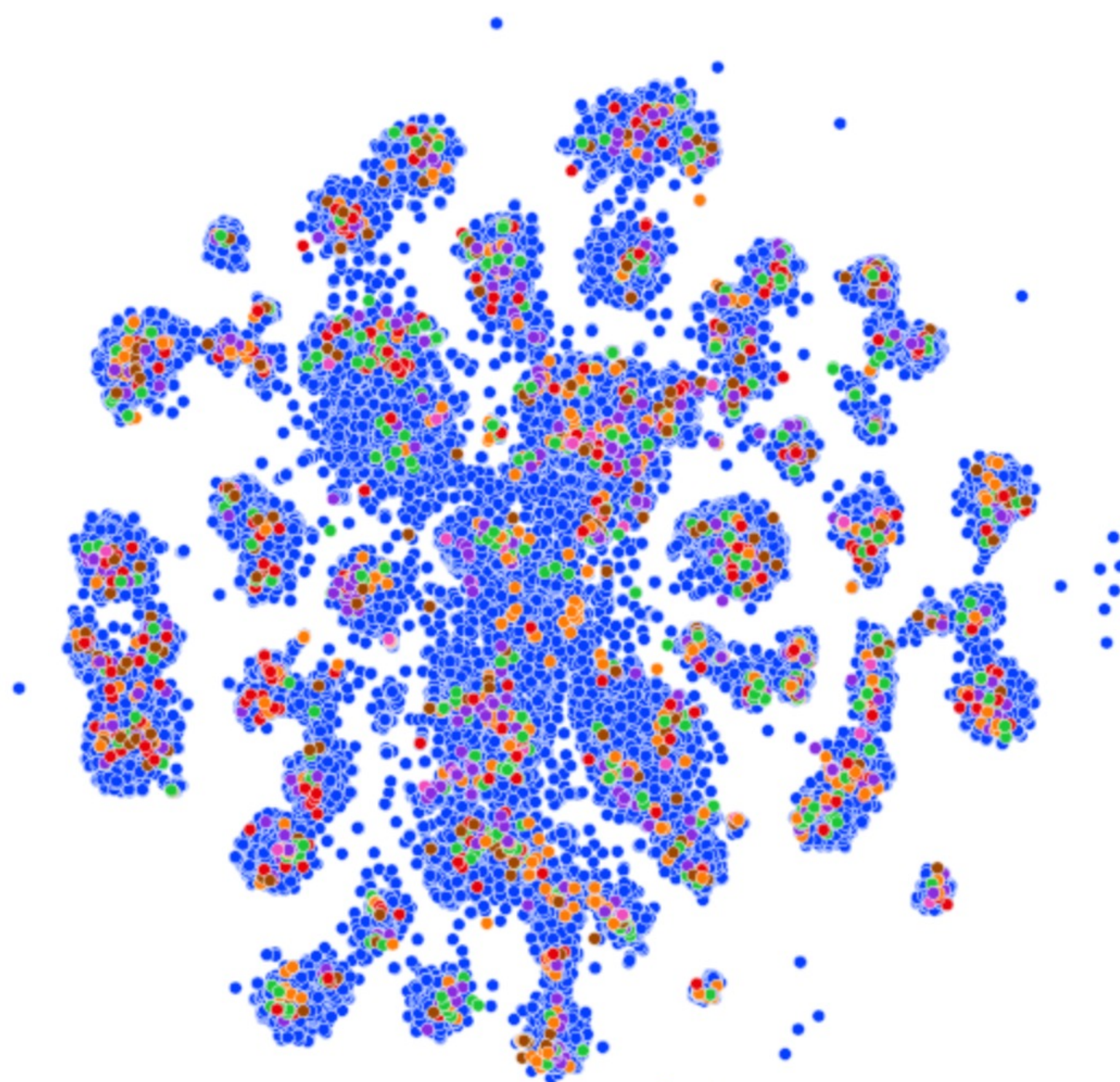
# 3.2 Visualization: Details

같은 Metadata (목적, 분위기 e.g. 데이트+사진찍기+분위기좋은) 끼리 뭉쳐있는 것이 아닐까?

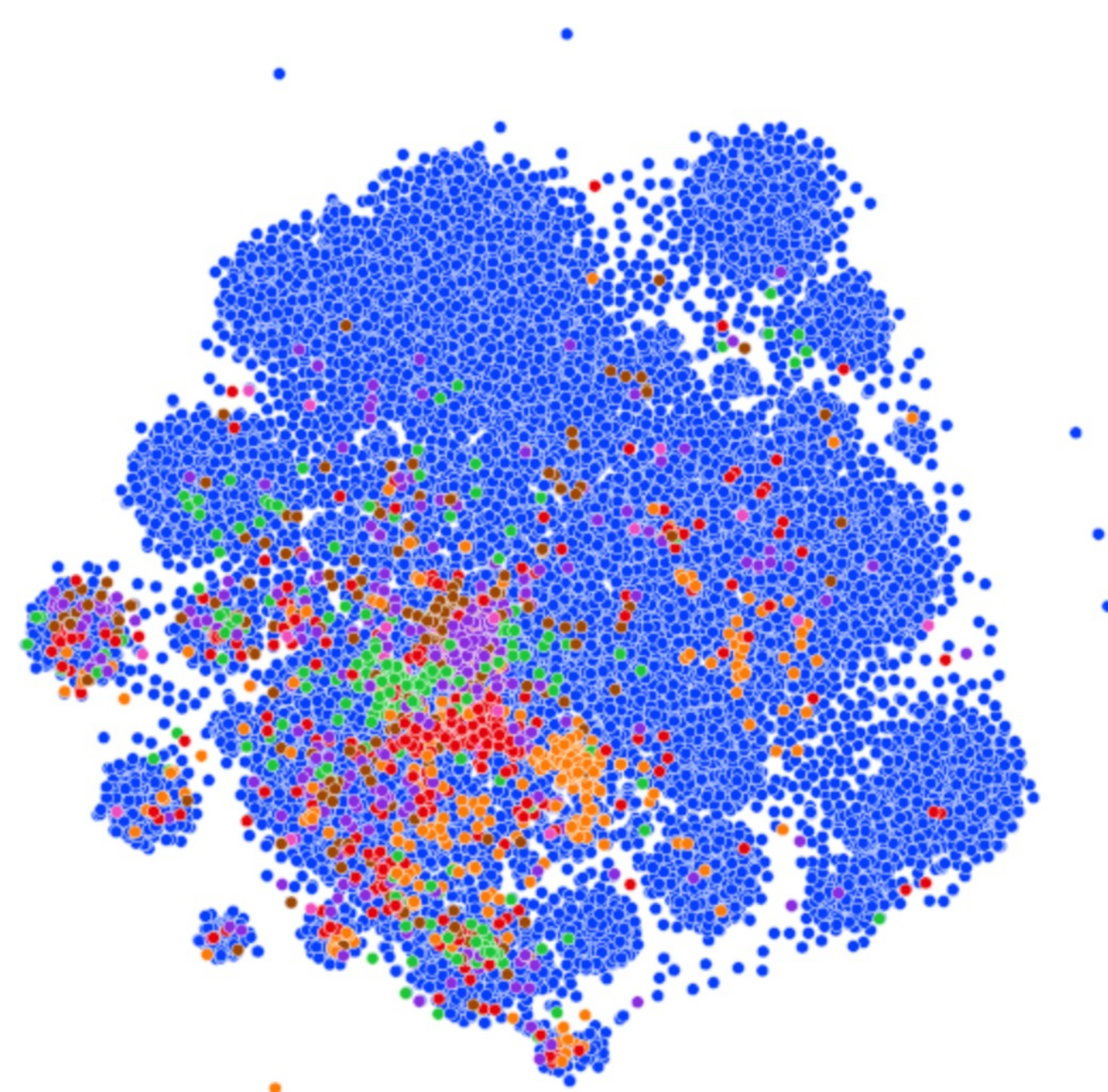


# 3.2 Visualization: Details

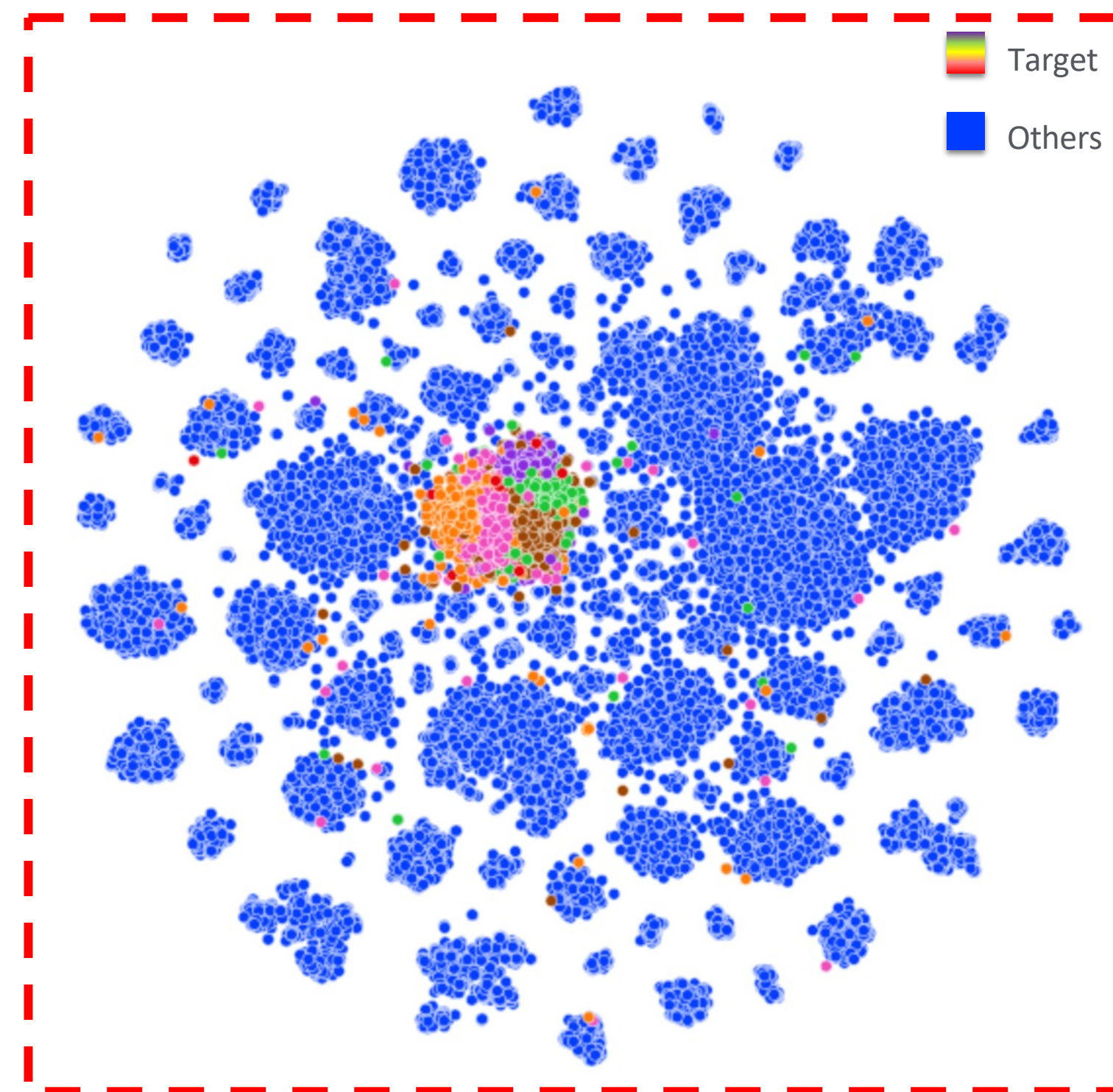
유사한 성향의 유저 (카테고리+소비수준 e.g. 한식+소비수준) 끼리 뭉쳐있는 것이 아닐까?



**POI2VEC**



**POIMeta2vec**



**POISAGE**

## 3.3 Precision

**Context  
Visualization**

각 모델들의 목적을  
잘 녹여냈을까?

**Precision**

어떤 상황에서  
얼마나 정확하게 추천할 수 있을까?

**Value of  
Personalization**

얼마나 개인화의 가치를 잘 담아냈을까?

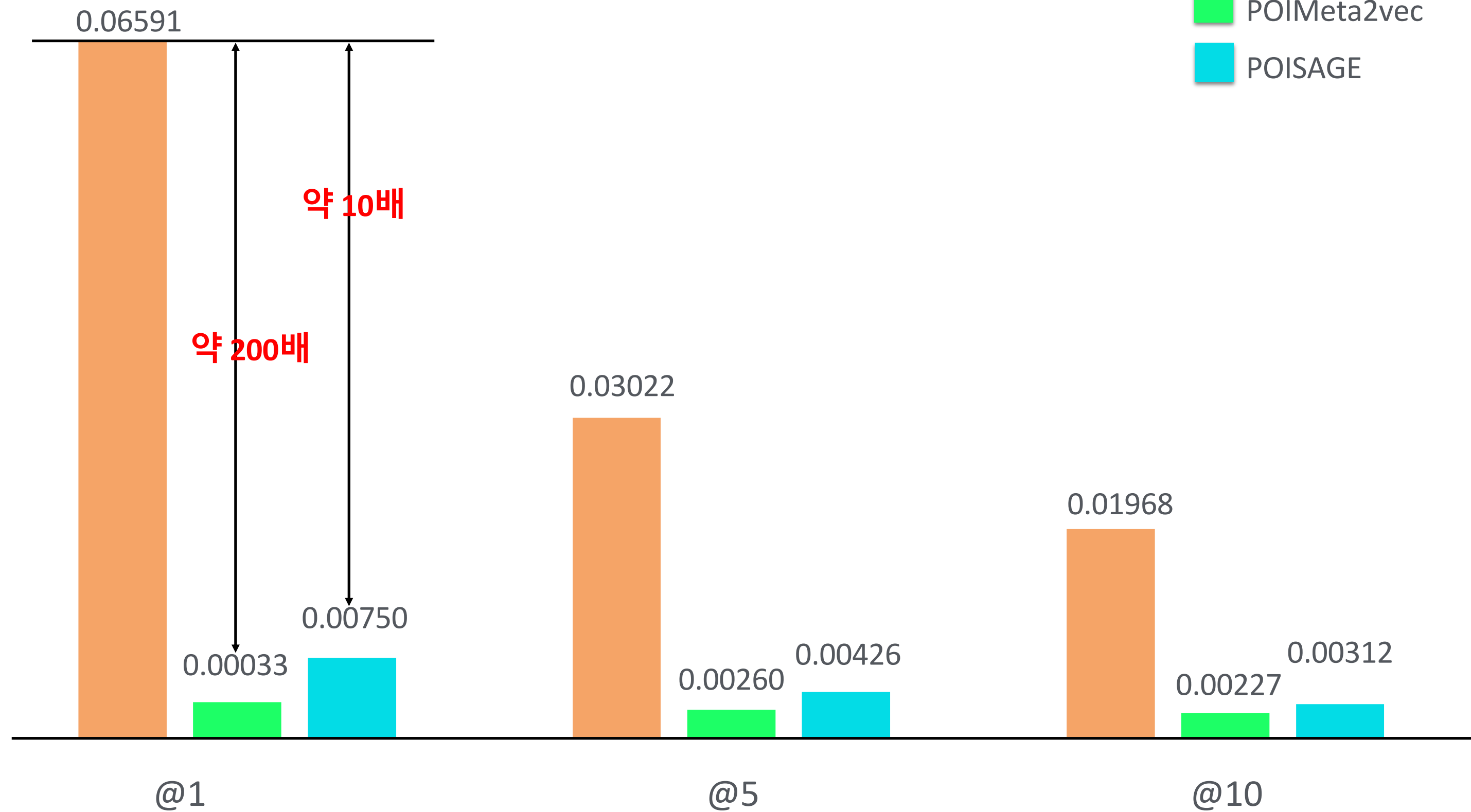


# 3.3 Precision: Same Intent

얼마나 정확하게 같은 지역 내, 동일 의도를 가진 업체를 추천해줄까?

Precision @K

- POI2VEC
- POIMeta2vec
- POISAGE



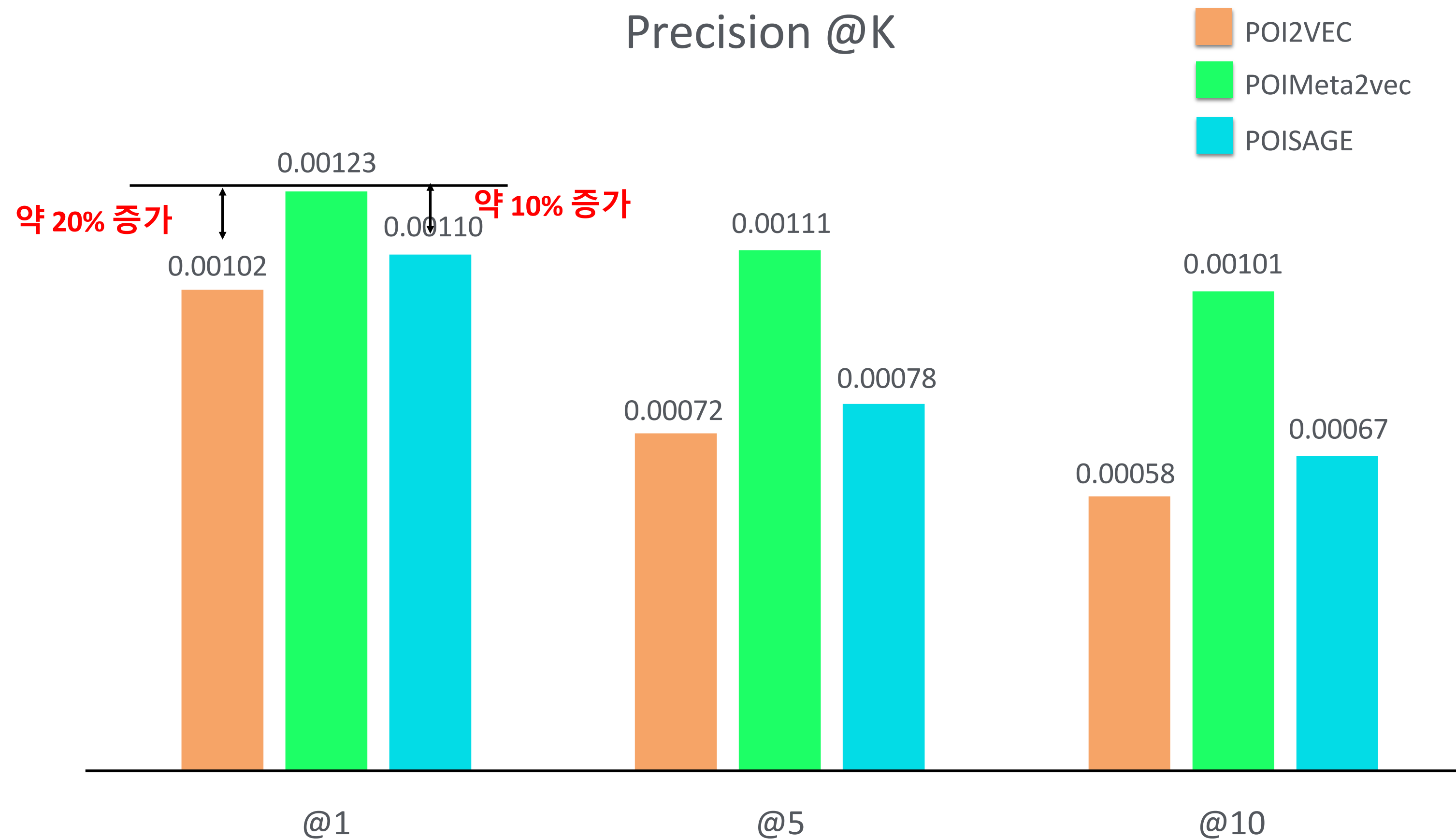
Dataset Description

- Session 내 클릭 기반 데이터
- ※ 업체를 검색 하는 동안 동일한 의도를 가질 것

→ POI2VEC이 같은 지역 내, 동일 의도의 업체를 가장 잘 추천!

# 3.3 Precision: Taste – Save

얼마나 정확하게 다른 지역에서, 유저의 취향과 유사한 성격의 업체를 추천해줄까?



### Dataset Description

- 취향 데이터 기반(Save)

※ 취향 데이터란, 유저가 Explicit 하게 POI에 대해 피드백 한 것으로, 좋아요(Like), 저장(Save)이 있음

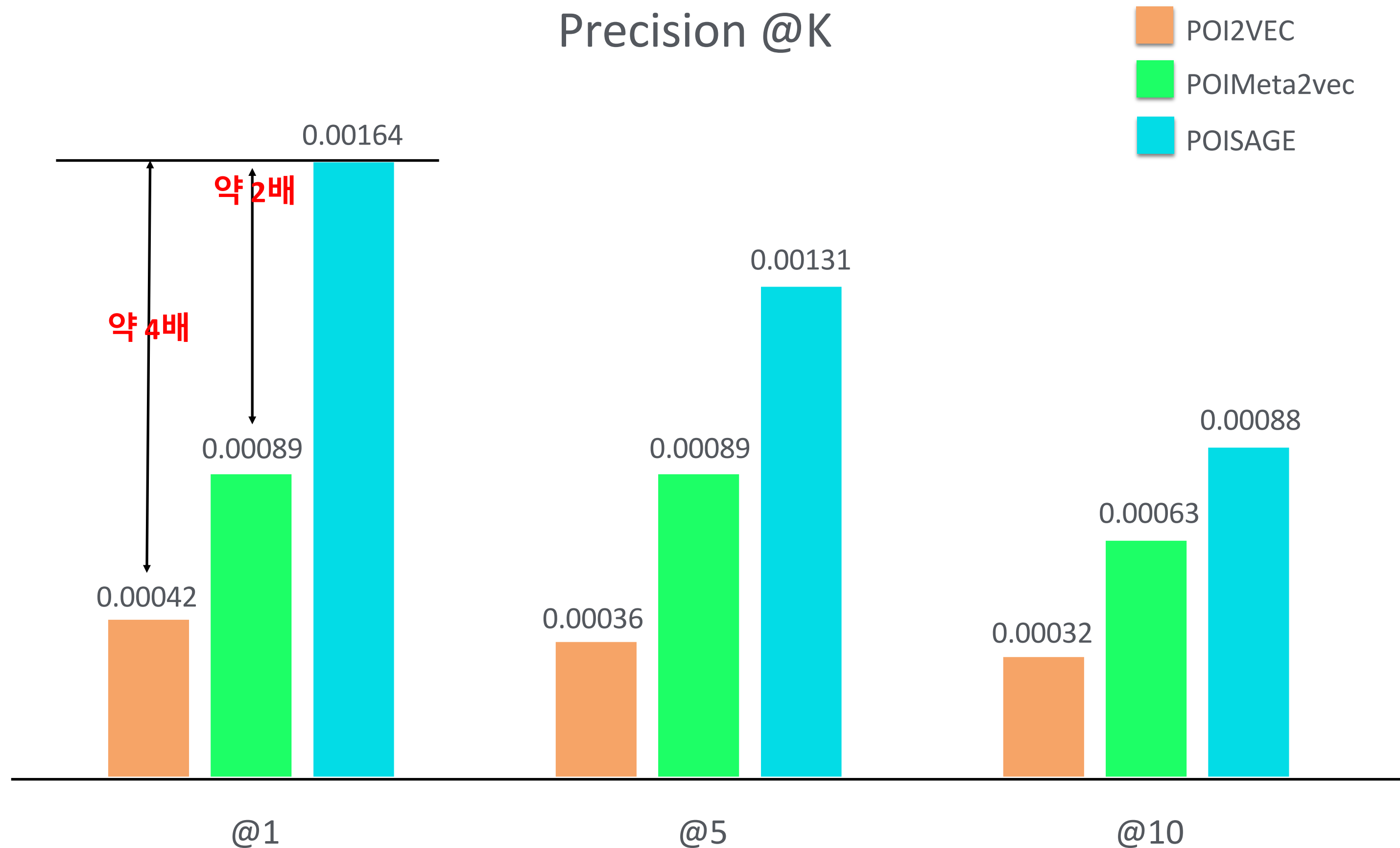
※ 좋아요(Like)는 업체를 방문한 뒤 남기는 피드백으로 강한 취향의 성격을 띠

※ 저장(Save)는 제약없이 자유롭게 남기는 피드백으로 약한 취향의 성격을 띠

→ POIMeta2vec 는 유저의 약한 취향을 잘 반영!

# 3.3 Precision: Taste – Like

얼마나 정확하게 다른 지역에서, 유저의 취향과 유사한 성격의 업체를 추천해줄까?



### Dataset Description

- 취향 데이터 기반(Like)
  - ※ 취향 데이터란, 유저가 Explicit 하게 POI에 대해 피드백 한 것으로, 좋아요(Like), 저장(Save)이 있음
  - ※ 좋아요(Like)는 업체를 방문한 뒤 남기는 피드백으로 강한 취향의 성격을 띠
  - ※ 저장(Save)는 제약없이 자유롭게 남기는 피드백으로 약한 취향의 성격을 띠

→ POISAGE 는 유저의 강한 취향을 잘 반영!

# 3.4 Value of Personalization



**Context  
Visualization**

각 모델들의 목적을  
잘 녹여냈을까?



**Precision**

어떤 상황에서  
얼마나 정확하게 추천할 수 있을까?



**Value of  
Personalization**

얼마나 개인화의 가치를 잘 담아냈을까?



## 3.4 Value of Personalization: Preview

업체는 얼마나 다양하게 추천 되는가?

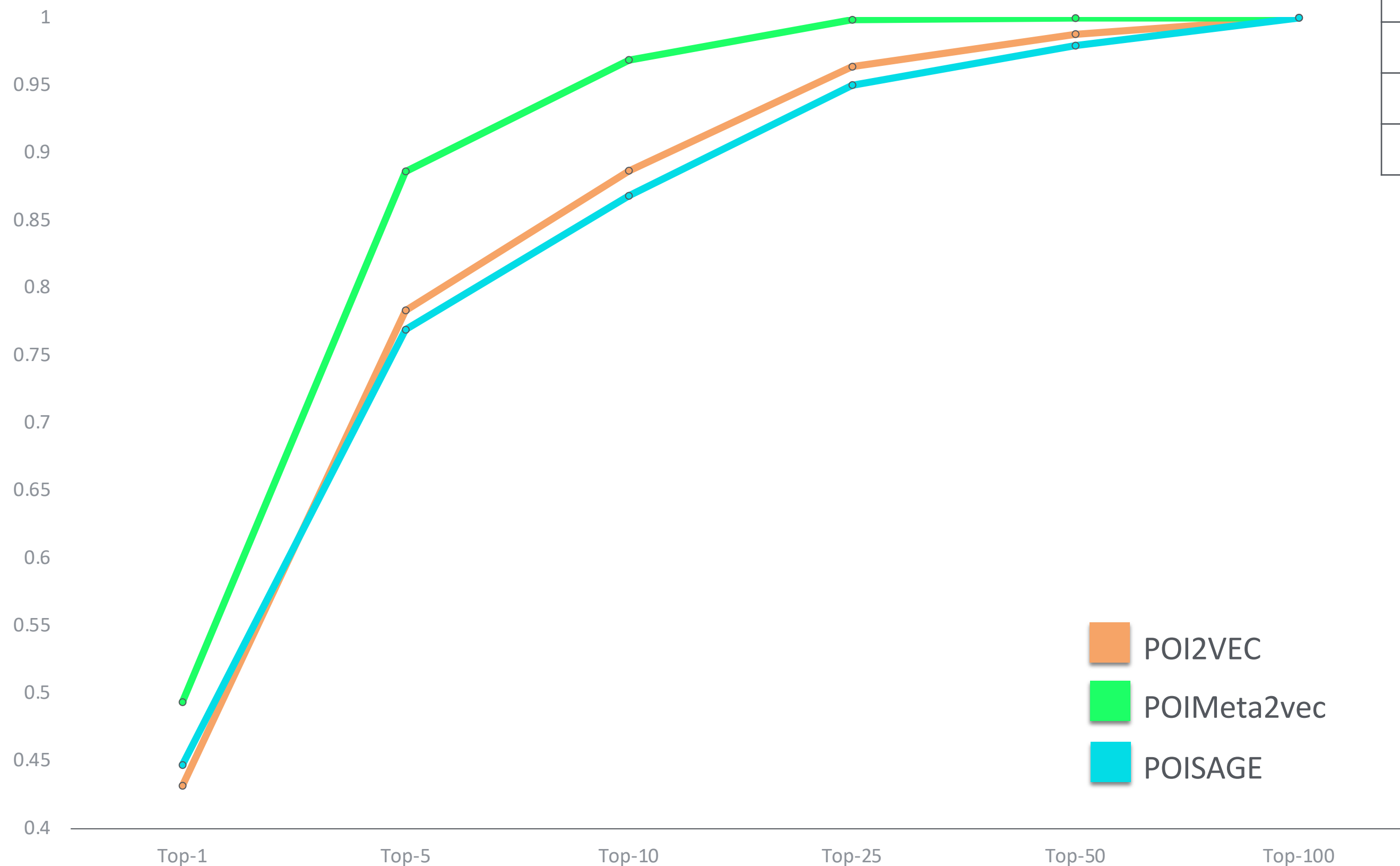
SME(Small and Medium Enterprise)를 높일 수 있을까?

실제로 추천이 잘 될까?: Case By Case

# 3.4 Value of Personalization: Diversity

업체는 얼마나 다양하게 추천 되는가?

POI Coverage(%)



	Top-1	Top-5	Top-10	Top-25	Top-50	Top-100
POI2VEC	0.43	0.78	0.89	0.96	0.99	1
<b>POIMeta2vec</b>	<b>0.49</b>	<b>0.89</b>	<b>0.97</b>	<b>1</b>	<b>1</b>	<b>1</b>
POISAGE	0.45	0.77	0.87	0.95	0.98	1

Coverage Description

- Coverage =  $\frac{\text{Top-}k \text{ 추천 리스트 업체 갯수}}{\text{전체 업체 갯수}}$

→ 기본적인 추세를 봤을 때,

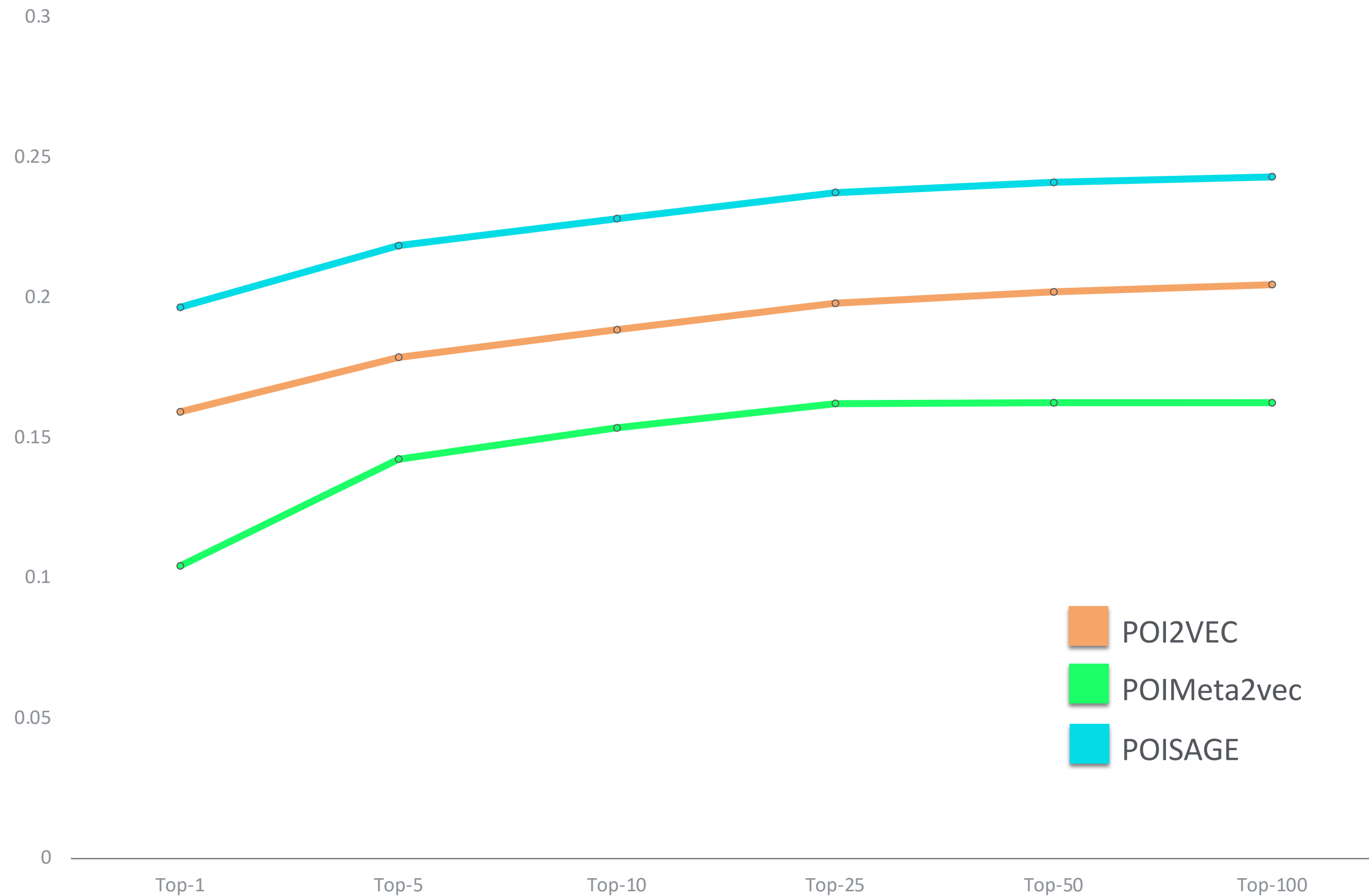
**POIMeta2vec > POISAGE > POI2VEC** 순으로

다양한 업체를 추천

# 3.4 Value of Personalization: SME

SME(Small and Medium Enterprise, 소상공인)를 높일 수 있을까?

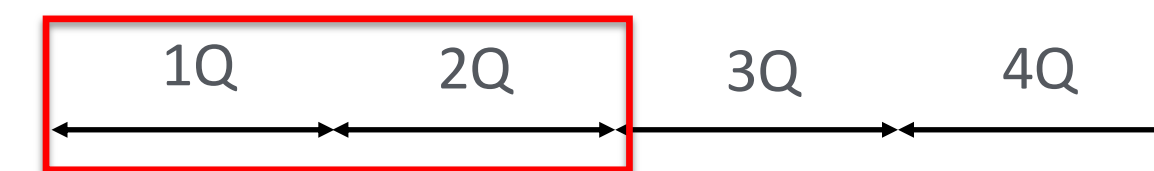
TOP @K



	Top-1	Top-5	Top-10	Top-25	Top-50	Top-100
POI2VEC	0.15933	0.17876	0.18858	0.19802	0.20212	0.20467
POIMeta2vec	0.10437	0.14245	0.15363	0.16227	0.16260	0.16259
<b>POISAGE</b>	<b>0.19663</b>	<b>0.21857</b>	<b>0.22821</b>	<b>0.23753</b>	<b>0.24117</b>	<b>0.24311</b>

SME Description

- 업체 추천 리스트에, 리뷰 수의 백분위수 기준으로 1Q ~ 2Q에 속하는 업체의 비율



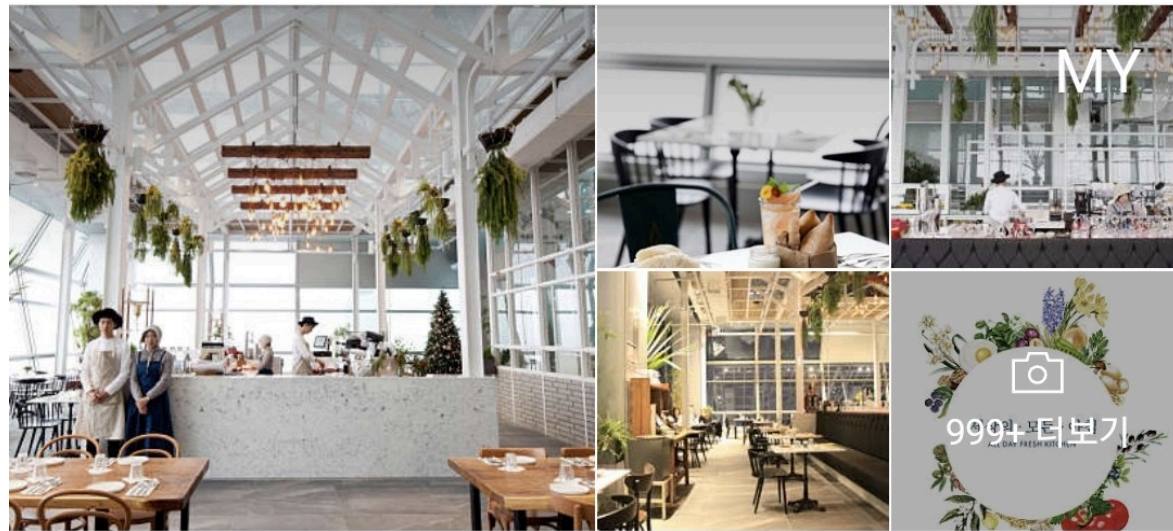
→ 월등하게 **POISAGE > POI2VEC > POIMeta2vec** 순으로 SME를 잘 추천!

## 3.4 Value of Personalization: Case by Case

실제로 추천이 잘 될까?

# 3.4 Value of Personalization: Case by Case

사용자 이력 (인기 업체)



**POI2VEC**



위치: 서울 영등포구 국제금융로10

콘래드서울 버티고 바(BAR)  
★ 4.28/5 · 방문자리뷰 784 · 블로그리뷰 590

세상의모든아침 브런치

★ 4.27/5 · 방문자리뷰 1,975 · 블로그리뷰 3,270

창가석 전망 뷰가 멋진 여의도 브런치카페 레스토랑

위치: 서울 영등포구 여의대로24

**POIMeta2vec**



위치: 서울 마포구 양화로3길

오스테리아샘킴 양식  
★ 4.26/5 · 방문자리뷰 237 · 블로그리뷰 692

<b>POI</b>	<b>메뉴</b>	<b>분위기</b>	<b>목적</b>
브런치	봉골레 리조또	분위기 좋은 사진찍기 좋은	재방문 데이트
⋮	⋮	⋮	⋮

<b>POI</b>	<b>메뉴</b>	<b>분위기</b>	<b>목적</b>
양식	봉골레 리조또	분위기 좋은 사진찍기 좋은	재방문 데이트
⋮	⋮	⋮	⋮

**POISAGE**



위치: 서울 동작구 동작대로27가길

37.5 동작이수점 브런치  
★ 4.62/5 · 방문자리뷰 1,216 · 블로그리뷰 119  
골드 대리석 인테리어가 예쁜 브런치카페

<b>POI</b>	<b>메뉴</b>	<b>분위기</b>	<b>목적</b>	<b>소비 수준</b>
브런치	리조또 파스타	친절한 분위기 사진찍기 좋은	재방문 데이트	
⋮	⋮	⋮	⋮	

# 3.4 Value of Personalization: Case by Case

사용자 이력 (SME)



무국적주점 거실 요리주점

★ 5/5 · 방문자리뷰 24 · 블로그리뷰 33

위치: 서울 성동구 성덕정17길

- POI**  
요리주점
- 메뉴**  
튀김 타파스
- 분위기**  
분위기 좋은 사진찍기 좋은

POI2VEC



위치: 서울 성동구 연무장길12

이로우 이자카야

★ 4.33/5 · 방문자리뷰 152 · 블로그리뷰 204

POIMeta2vec



위치: 서울 마포구 어울마당로123

새벽 요리주점

★ 4.36/5 · 방문자리뷰 98 · 블로그리뷰 44

홍대 분위기 좋고 안주 맛있는 술집

- POI**  
요리주점
- 메뉴**  
짬뽕
- 목적**  
재방문 데이트  
⋮

POISAGE



위치: 서울 동작구 동작대로27가길

무국적 식당 주철 요리주점

★ 4.67/5 · 방문자리뷰 87 · 블로그리뷰 66

안주가 다양하고 맛있는 송파구 문정동 술집

- POI**  
요리주점
- 메뉴**  
순대 감자전
- 분위기**  
친절한 분위기
- 소비 수준**

# 3.5 Summary

	<b>POI2VEC</b>	<b>POIMeta2vec</b>	<b>POISAGE</b>
<b>Context Visualization</b>	같은 지역의 POI끼리 유사하도록	같은 Metadata를 가지는 POI끼리 유사하도록	유사한 성향의 유저들이 방문한 POI끼리 유사하도록
<b>Coverage(Diversity)</b>	모든 면에서 낮음	업체의 다양성 측면에서 월등함	SME 측면에서 월등함
<b>Case by Case</b>	지역성이 강하게 보임	카테고리에 한정되지 않은, 다양한 추천을 보임	뾰족한 추천을 보임

# 4. Future Works & Conclusion



# 4.1 Future Works



**Explainable  
Recommendation**

추천된 장소에 대한 신뢰성



**Multi-Intent  
Deeply**

사용자의 의도 파악



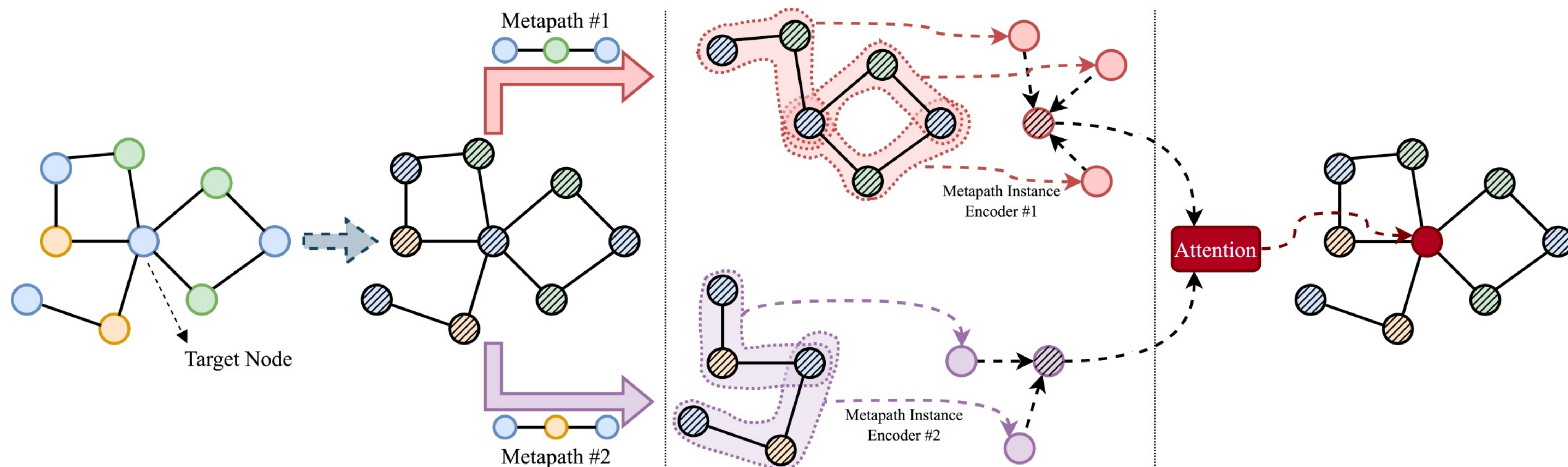
**Multimodal**

다양한 데이터로 Embedding을 풍부하게

# 4.1 Future Works: Explainable

## MAGNN

- Node Content Transformation
- Intra-metapath Aggregation
- Inter-metapath Aggregation

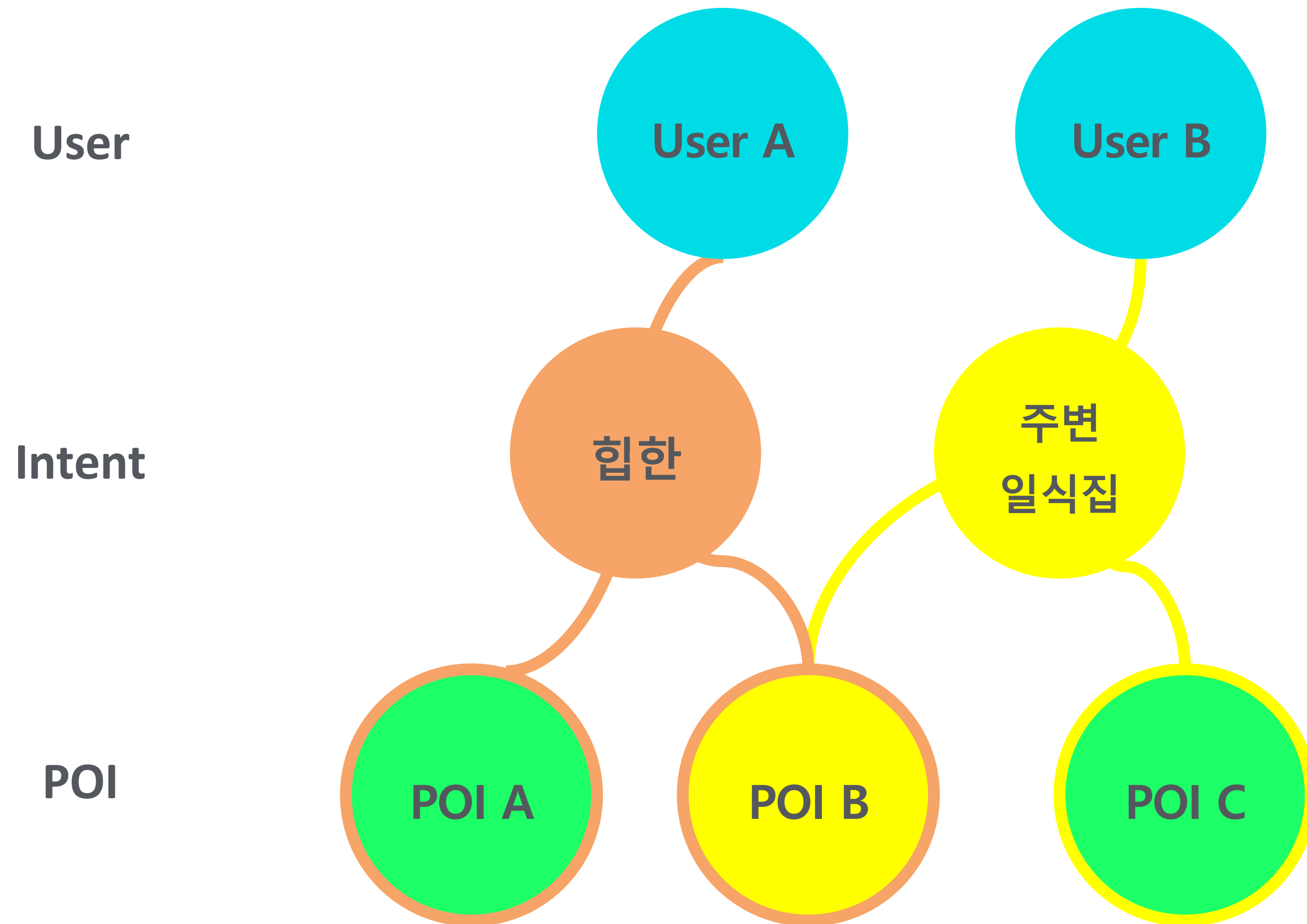


**(a) Node Content Transformation**

**(b) Intra-metapath Aggregation**

**(c) Inter-metapath Aggregation**

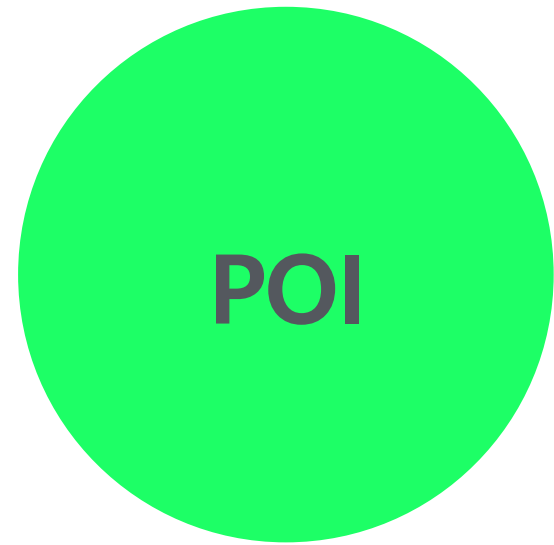
# 4.1 Future Works: Multi-Intent



## HyperGraph

- Understanding Multi-Intents from Multi POIs
- Understanding POIs by Hyper Edges

# 4.1 Future Works: Multimodal



## 다양한 데이터로 표현

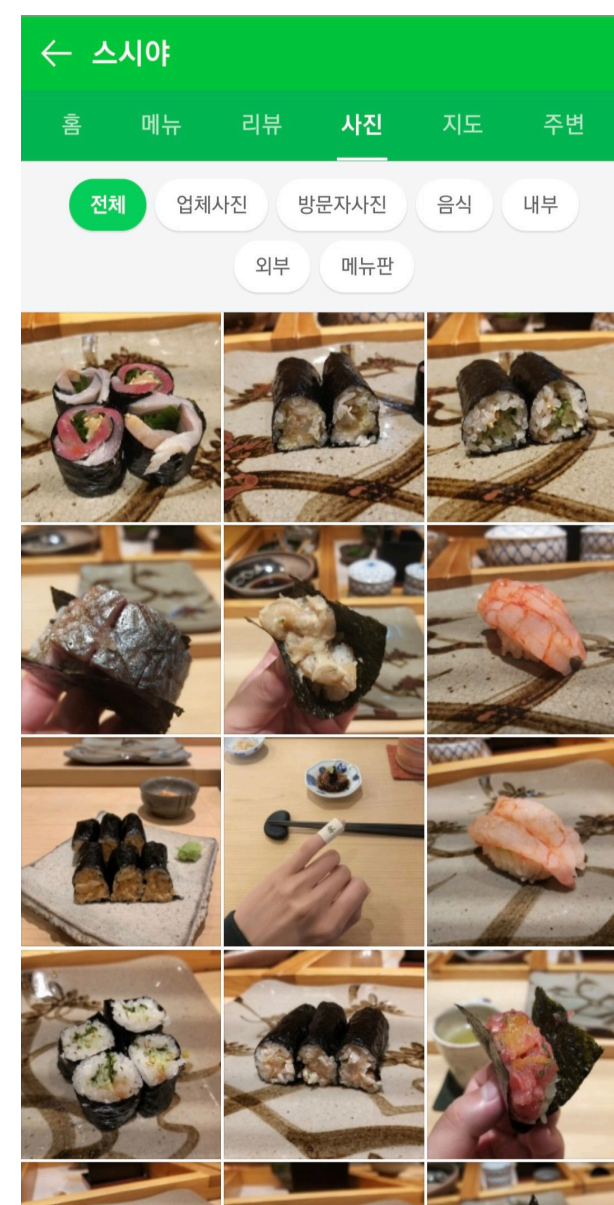
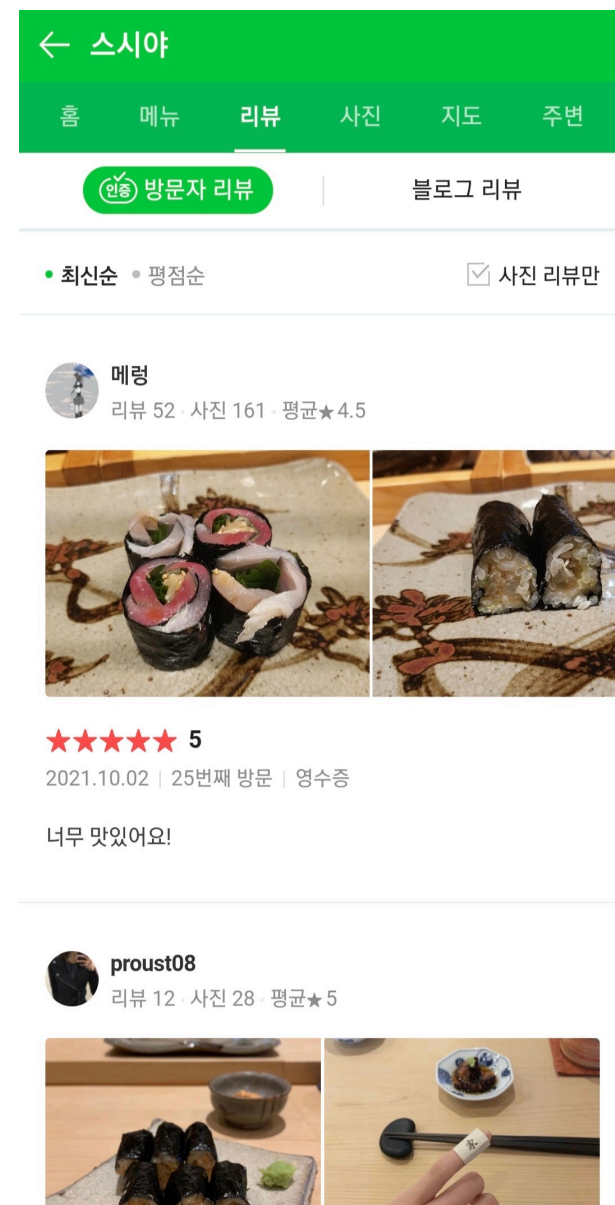
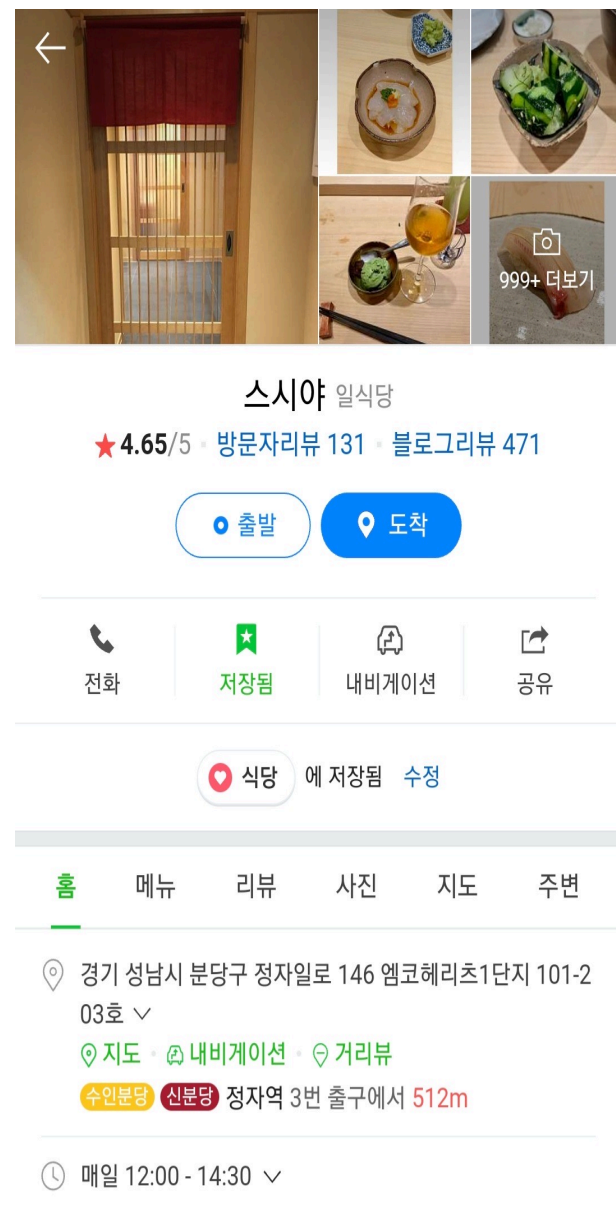
- Text (  $x_W$  , e.g. 업체명, 리뷰 ...)
- Image (  $x_I$  , e.g. 리뷰, 업체 대표 사진 ...)
- ⋮

By Multimodal

$$f_T(x_W)$$

$$f_I(x_I)$$

**Deep Joint Representation**

$$Z = [f_T(x_W), f_I(x_I)]$$


## 4.2 Conclusion

1. 풀고자 하는 문제 상황 및 Context 를 정의하는 것이 중요하다.
2. Knowledge Graph는 특정 도메인을 표현하는데 있어 강력하다.
3. 문제 상황 및 Context에 맞는 Knowledge Graph Design & Modeling 이 중요하다.
4. Embedding의 특성을 잘 파악하면, 각 특성에 맞는 적합한 추천 서비스 및 시스템을 생성할 수 있다.

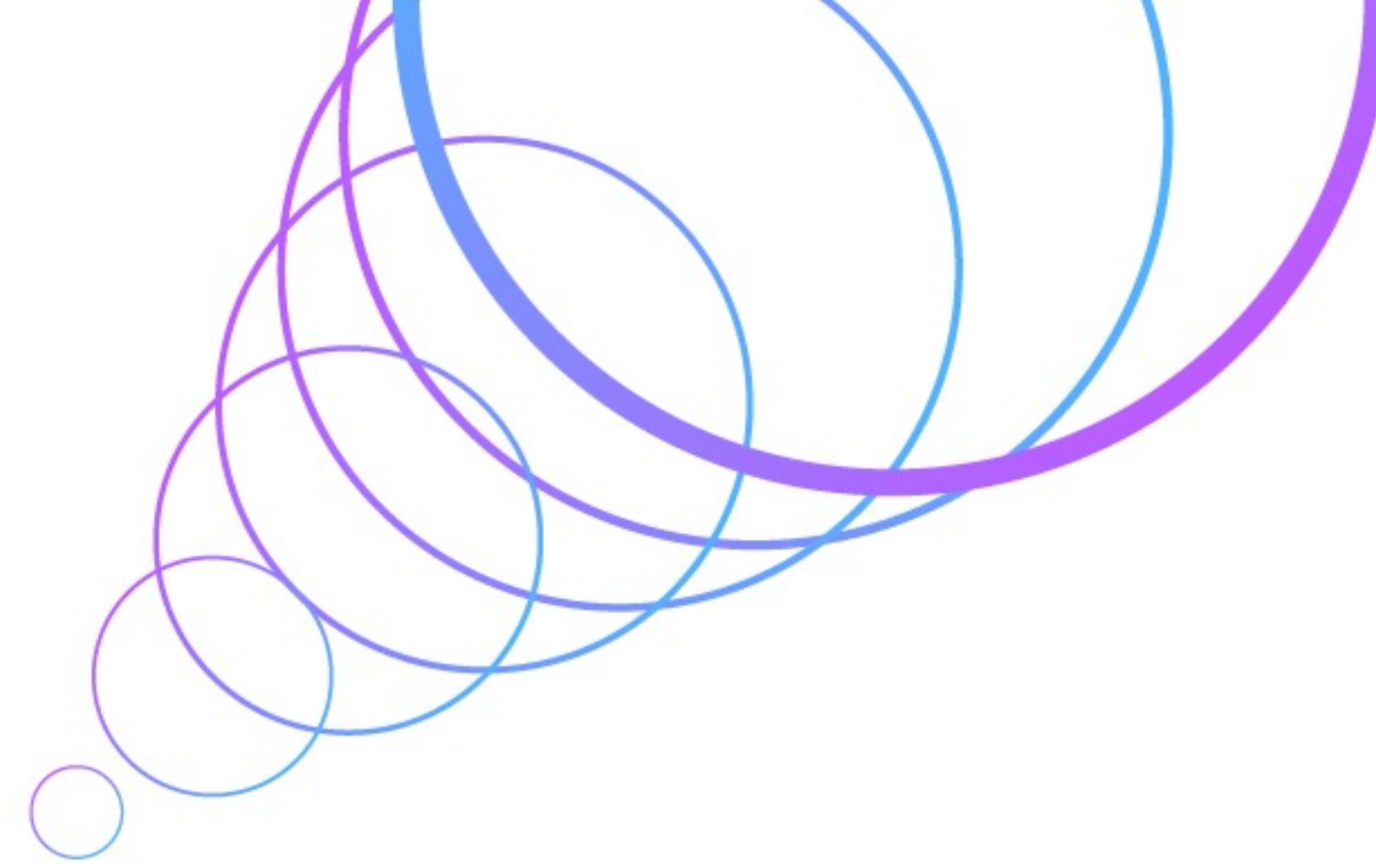
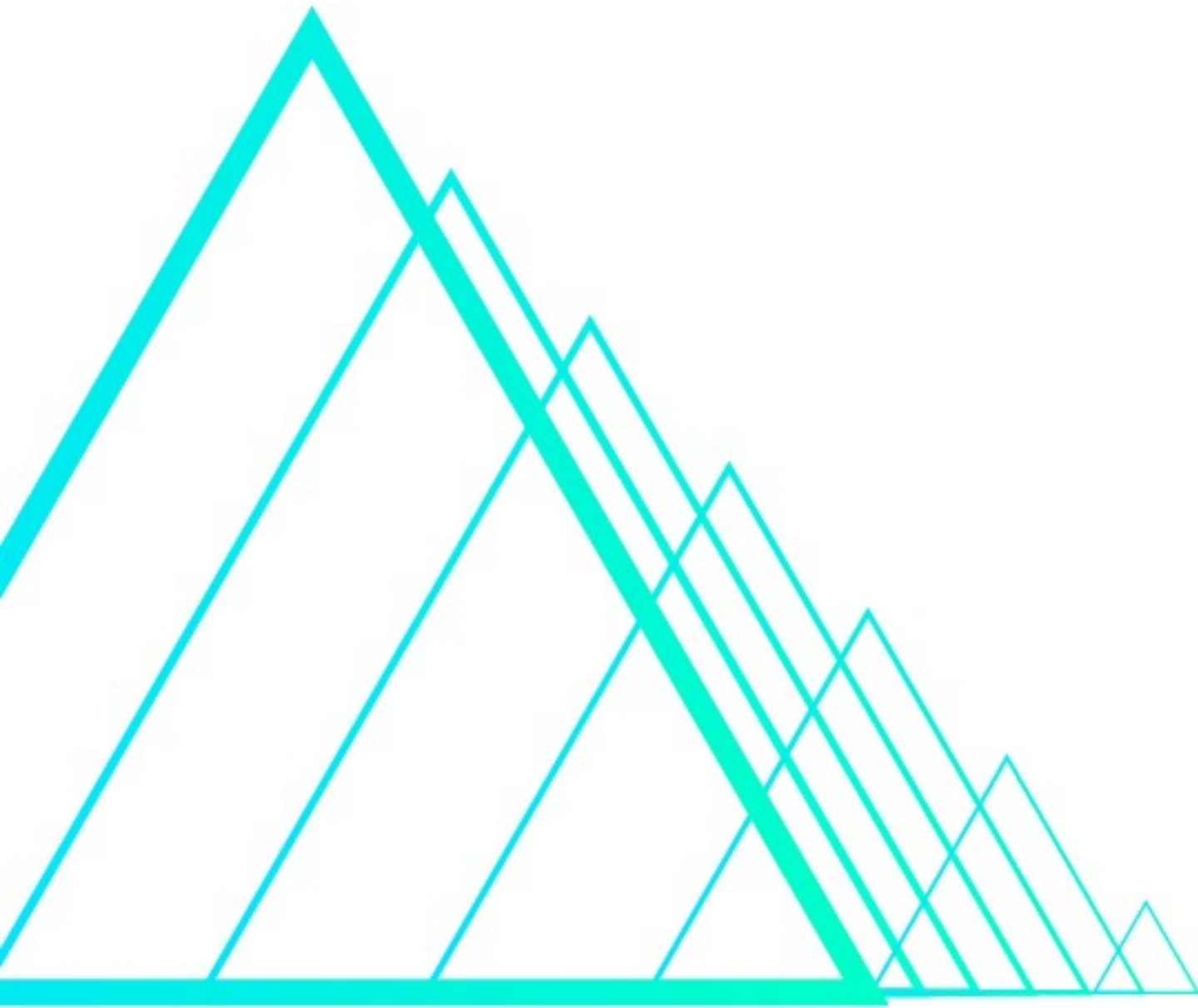


# We are Hiring!

AirSPACE (신입/경력)

<https://naver-career.gitbook.io/kr/service/search/local/airspace>





**Thank You**

