

Context-Aware Co-attention Neural Network for Service Recommendations

<https://arxiv.org/html/2506.10329v1>

0. Introduction

- 추천 시스템에서 컨텍스트 정보(시간, 장소, 동행자 등)가 추천 정확도 높이는 데 중요 함.
- 사용자 리뷰 텍스트는 유저 선호, 아이템 속성, 암묵적 컨텍스트까지 담고 있어 활용 가치 큼.
- 기존 연구는 유저-아이템 또는 아이템-컨텍스트만 다루는 경우가 많아서 세 요소를 동시에 모델링하는 접근 부족했음.
- 이 논문은
 - 유저-컨텍스트와 아이템-컨텍스트의 상호작용을 동적으로 계산하는 CCANN(Co-attention 기반 추천 모델) 제안함
 - 리뷰 텍스트로 유저·아이템·컨텍스트·단어를 함께 임베딩하는 Entity2Vec 기법 제 안함
 - 대규모 TripAdvisor 데이터에서 기존 모델보다 좋은 성능 보였다고 주장함

1. Overview

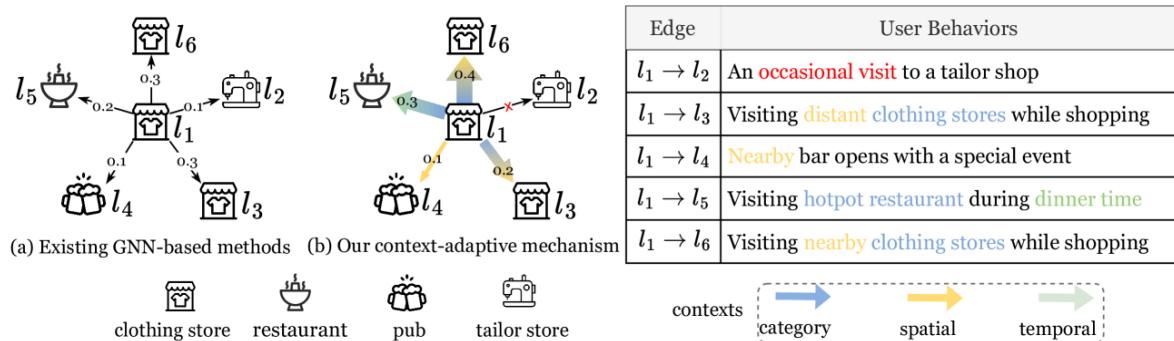
- 핵심 아이디어는 유저, 아이템, 컨텍스트 세 가지 요소를 동시에 고려해 상황별 맞춤 추천을 강화하는 것임.
- 기본 구조
 - 유저의 컨텍스트별 선호도와 아이템의 컨텍스트 측면을 각각 MLP로 생성함
 - Co-attention으로 두 표현 간 매칭 점수 계산함
 - attention 가중치로 유저/아이템 표현 강화함

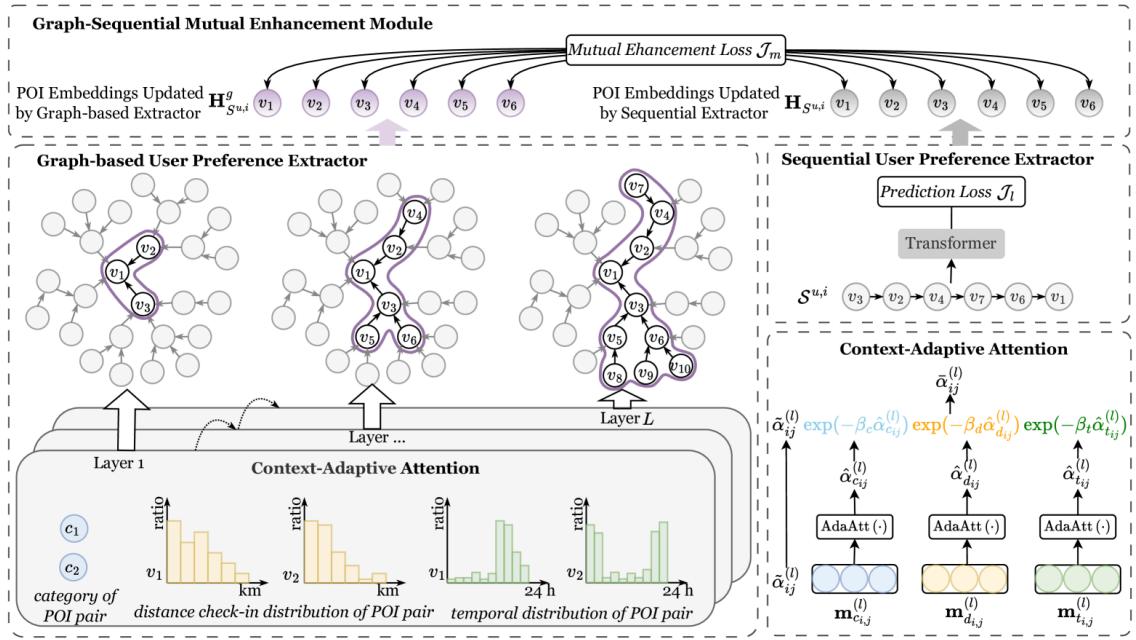
- Factorization Machine(FM)으로 최종 평점 예측함
- 데이터는 TripAdvisor 호텔 리뷰 사용함.
- 목표는 컨텍스트 반영 + 리뷰 기반 임베딩으로 추천 성능과 설명력 함께 높이는 것임.

2. Challenges

- 컨텍스트 변수가 여러 개일 때 이를 모두 반영해 유저·아이템 관계를 모델링하는 게 복잡함.
- 리뷰 텍스트에서 암묵적 컨텍스트 추출하는 일 쉬운 작업 아님.
- 기존 모델은 세 가지(entity) 상호작용을 깊게 표현하지 못함.
- 리뷰 기반 모델은 보통 복잡해 학습 비용이 높은데, 이를 더 효율적으로 처리해야 하는 문제 있음.

3. Method





3.0 CCANN

- 유저 컨텍스트 선호도:
 - 유저 임베딩 + 특정 컨텍스트 값 concat → MLP → 유저-컨텍스트 선호 벡터 생성함
- 아이템 컨텍스트 측면:
 - 아이템 임베딩 + 동일 컨텍스트 concat → MLP → 아이템-컨텍스트 측면 벡터 생성함
- Co-attention:
 - 유저-컨텍스트 벡터와 아이템-컨텍스트 벡터의 유사도를 attention score로 계산함
 - 컨텍스트별 중요도가 달라지는 구조임
- 강화된 표현:
 - attention 가중치로 유저·아이템 벡터 각각 가중합
- 예측:
 - 두 벡터 concat → FM에 넣어 평점 예측
 - 손실은 MSE 기반으로 최적화함

3.1 Entity2Vec

- 리뷰 텍스트 기반으로 단어, 유저, 아이템, 컨텍스트를 같은 공간에서 임베딩함.

- CBOW 방식처럼 슬라이딩 윈도우 내 엔티티 평균 → 타깃 단어 예측 방식으로 학습함.
- 이렇게 학습된 임베딩이 CCANN 초기값으로 들어가 성능을 올리는 역할 함.

4. Experiments / Data

Dataset	#Users	#POIs	#Check-ins	Density
PHO	2,946	7,247	47,980	0.22%
NY	16,387	56,252	511,431	0.06%
SIN	8,648	33,712	355,337	0.12%

- TripAdvisor 리뷰 사용함(도시: 홍콩, 뉴욕, 런던).
- 구성 요소: userID, itemID(호텔), 평점, 리뷰 텍스트, 명시적 컨텍스트(time, companion, purpose 등).
- 데이터 분할은 train/valid/test = 8:1:1.
- 평가 지표는 RMSE 사용함.
- 비교 대상
 - 비컨텍스트 모델: PMF, FM
 - 리뷰 기반 모델: ConvMF+, DeepCoNN
 - 컨텍스트 기반 모델: NFM, AIN
- 실험은 여러 번 반복해 평균 측정함.

5. Results

Metric	RNN-based			Transformer-based			Graph-based				CAGNN	Improve
	ST-RNN	ATST-LSTM	PLSPL	CTLE	CFPRec	CLSPRec	LightGCN	SGRec	AGRAN	DCHL		
PHO	HR@1	0.0419	0.0517	0.0846	0.1439	0.1842	<u>0.2521</u>	0.0775	0.1295	0.1937	0.2057	0.2945***
	HR@5	0.1240	0.1579	0.1775	0.2632	0.3421	<u>0.5217</u>	0.2563	0.2897	0.4573	0.4962	0.5611**
	HR@10	0.2028	0.2377	0.2569	0.3605	0.4253	<u>0.5936</u>	0.3151	0.3401	0.5248	0.5714	0.6271**
	N@1	0.0419	0.0517	0.0846	0.1439	0.1842	<u>0.2521</u>	0.0775	0.1295	0.1937	0.2057	0.2945***
	N@5	0.0802	0.1033	0.1285	0.1995	0.2432	<u>0.3806</u>	0.1881	0.2048	0.3685	0.3336	0.4347***
	N@10	0.1229	0.1385	0.1538	0.2068	0.2730	<u>0.4126</u>	0.2194	0.2249	0.3962	0.3638	0.4566***
NY	HR@1	0.0327	0.0396	0.0426	0.0649	0.0718	<u>0.1797</u>	0.0527	0.0674	0.1296	0.1498	0.2065***
	HR@5	0.1347	0.1667	0.1741	0.2421	0.2771	<u>0.3518</u>	0.1752	0.1891	0.2732	0.3292	0.4023***
	HR@10	0.1826	0.2031	0.2413	0.3205	0.3606	<u>0.4388</u>	0.2229	0.2443	0.3746	0.3904	0.4880***
	N@1	0.0327	0.0396	0.0426	0.0649	0.0718	<u>0.1797</u>	0.0527	0.0674	0.1296	0.1498	0.2065***
	N@5	0.0593	0.0912	0.0961	0.1513	0.1971	<u>0.2583</u>	0.1035	0.1089	0.2097	0.2217	0.3076***
	N@10	0.1303	0.1638	0.1825	0.1841	0.2190	<u>0.2792</u>	0.1209	0.1877	0.2308	0.2457	0.3423***
SIN	HR@1	0.0439	0.0478	0.0513	0.0722	0.0744	<u>0.2132</u>	0.0823	0.0921	0.1552	0.1732	0.2312**
	HR@5	0.0959	0.1296	0.1447	0.2041	0.2310	<u>0.3564</u>	0.2165	0.2310	0.2955	0.3573	0.3890**
	HR@10	0.1370	0.1933	0.1719	0.2784	0.3085	<u>0.4077</u>	0.2691	0.2953	0.3339	0.4045	0.4375**
	N@1	0.0439	0.0478	0.0513	0.0722	0.0744	<u>0.2132</u>	0.0823	0.0921	0.1552	0.1732	0.2312**
	N@5	0.0655	0.1027	0.1126	0.1315	0.1588	<u>0.2763</u>	0.1263	0.1530	0.2576	0.2670	0.3155**
	N@10	0.0794	0.1476	0.1384	0.1556	0.1836	<u>0.2939</u>	0.1335	0.1739	0.2697	0.2856	0.3313**

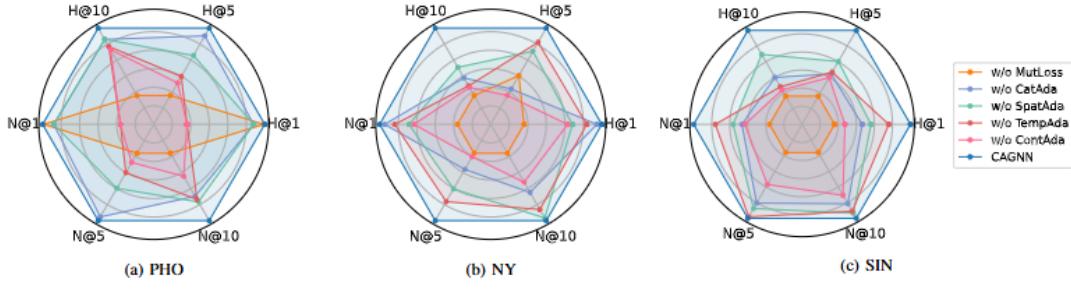


Fig. 3: Performance comparison of different CAGNN variants on the three datasets in terms of HR ('H') and NDCG ('N').

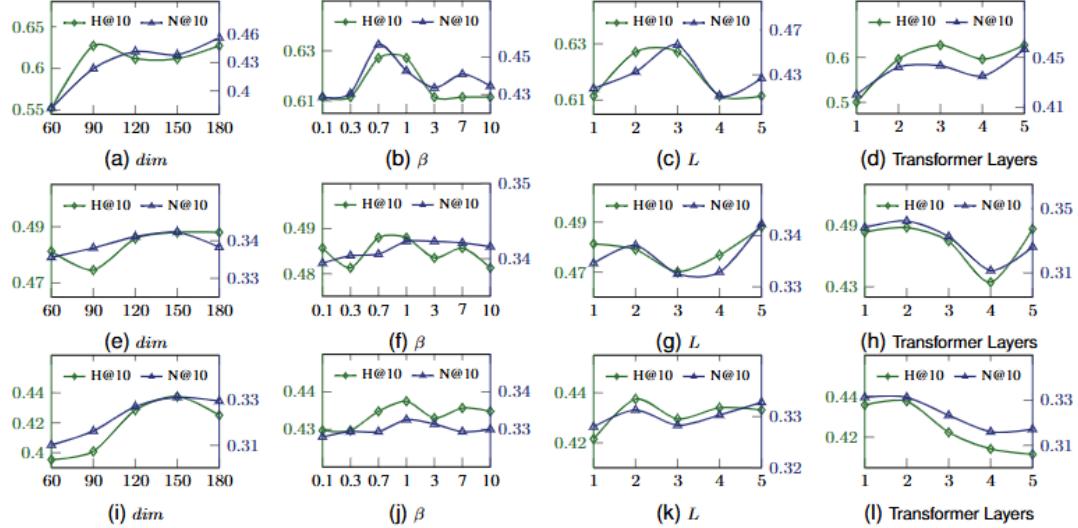
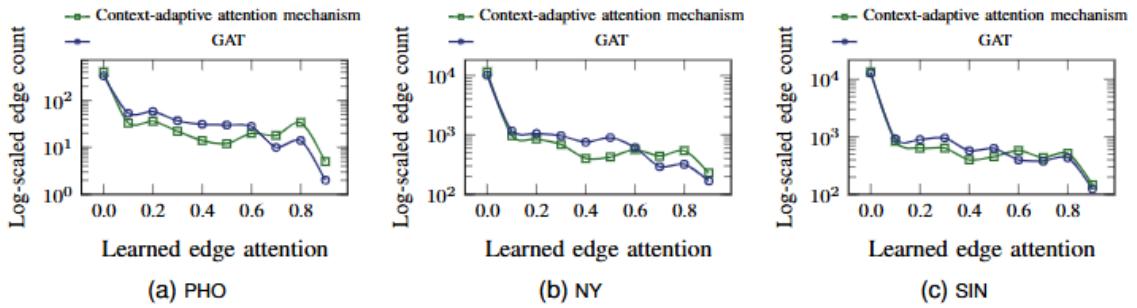


Fig. 4: Parameter sensitivity analysis of CAGNN on three datasets regarding HR ('H') and NDCG ('N'). Subfigures (a)–(d) correspond to PHO, (e)–(h) to NY, and (i)–(l) to SIN.



- CCANN이 대부분 baseline보다 RMSE 낮아서 성능 우수함.
- 컨텍스트 반영 모델이 비컨텍스트 모델보다 전반적으로 더 좋은 성능 보임.
- 리뷰 기반 모델도 강하지만 컨텍스트 정보까지 반영한 CCANN이 더 안정적임.
- Entity2Vec 임베딩 초기화 모델이 랜덤 초기화보다 성능 더 좋았음.
- 구조 자체가 MLP + FM라서 리뷰 기반 딥러닝 모델 대비 계산 효율 높음.
- attention weight를 분석하면 어떤 컨텍스트가 추천에 중요한지 해석 가능함.

6. Insight

- 유저·아이템·컨텍스트를 동시에 고려해야 현실세계 추천 상황을 더 잘 반영할 수 있음.
- 리뷰를 단순 텍스트가 아닌 의미 기반 임베딩으로 활용하면 추천 성능 크게 개선됨.
- Co-attention 구조는 추천의 "이유"를 설명하는 데 유용함.
- 컨텍스트 상호작용 모델링
- 리뷰 기반 임베딩 활용
- 계산 비용 낮음
- 설명력 확보
- 리뷰 적은 도메인에서는 성능 개선 제한됨
- 명시적 컨텍스트 없는 데이터에서는 적용 힘듦
- 컨텍스트가 많아질수록 attention 계산 복잡해짐
- 여행, 호텔, 음식점처럼 상황에 따라 선호 바뀌는 서비스에서 특히 효과 큼
- 리뷰 텍스트가 풍부한 플랫폼일수록 효과 커짐