

OPEN ACCESS

편집자:

마틴 아츠윌러, 오
스나브르크 대학교, 독일

검토자:

센트럴 플
로리다 대학교, Yanjie Fu,
미국 주화 후,
워싱턴 대학교,
미국

*통신: Nitesh V.
Chawla
nchawla@nd.edu

스페셜티 섹션:

이 기사는 다음에 제출되었습니다.

빅 데이터 네트워크,
빅 데이터의 프론티어
저널의 한 섹션

¹ 미국 노트르담 대학교 컴퓨터 과학 및 공학부 및 루시 패밀리 데이터 및 사회 연구소, 미국 노트르담, 인디애나주; ² 브랜다이스 대학교 컴퓨터 과학부, 미국 매사추세츠주 윌섬, 매사추세츠

레시피 추천 시스템은 사람들이 관심 있고 자신의 식습관에 맞는 레시피를 찾는 데 중요한 역할을 합니다. 콘텐츠 기반 또는 협업 필터링 접근 방식을 사용하여 레시피를 추천하기 위해 개발된 것과 달리 사용자, 레시피, 음식 항목 간의 관계 정보에 대한 탐색은 미흡합니다. 본 논문에서는 레시피 추천에 관계형 정보를 활용하고 이를 해결하기 위한 그래프 학습 접근법을 제안합니다. 특히 레시피 추천을 위한 새로운 계층적 그래프 주의 네트워크인 HGAT를 제안합니다. 제안 모델은 유형별 변환, 노드 수준 주의, 관계 수준 주의 등 여러 신경망 모듈을 통해 사용자 이력 행동, 레시피 콘텐츠, 관계 정보를 포착할 수 있습니다. 또한 랭킹 기반 목적 함수를 도입하여 모델을 최적화합니다. 철저한 실험을 통해 HGAT가 수많은 기존 방법보다 성능이 뛰어나다는 것이 입증되었습니다.

키워드: 레시피 추천, 사용자 행동 모델링, 레시피 그래프, 음식, 그래프 관심 네트워크

1. 소개

대규모 음식 데이터는 음식에 대한 풍부한 지식을 제공하며 인류 사회의 여러 핵심 문제를 해결하는 데 도움이 될 수 있습니다(Mouritsen 외., 2017; Min 외., 2019; Tian 외., 2021). 특히 레시피 웹사이트는 개인이 자신이 만든 레시피를 온라인에서 공유하기를 원하기 때문에 대량의 음식 데이터를 포함하고 있습니다(Teng et al., 2012). 이는 다른 사용자가 평가하고 댓글을 달 수 있는 기회를 제공



하여 사람들이 무엇을 먹을지 결정할 때 이러한 웹사이트를 참조하는 습관을 형성하는 데 도움이 됩니다(Ueda et al., 2011). 세계에서 가장 큰 레시피 공유 웹사이트 중 하나인 Food.com⁽¹⁾은 50만 개가 넘는 레시피를 수집합니다. 이러한 방대한 양의 데이터는 레시피 제공 서비스에 대한 수요가 많다는 것을 반영합니다(Ueda et al., 2011). 특히 레시피가 재료, 조리법, 영양소, 사용자 피드백 등 다양한 이질적인 콘텐츠와 연관되어 있는 경우, 이 압도적인 양의 온라인 레시피 리소스를 파헤쳐 만족스러운 레시피를 찾는 것은 항상 어려운 일입니다(Britto et al., 2020). 하지만 레시피 추천 시스템은 사용자가 수많은 온라인 레시피 데이터를 탐색하고 사용자의 선호도 및 이력 행동에 맞는 레시피를 추천할 수 있는 강력한 기능을 갖추고 있습니다(Khan et al., 2019).

기존의 레시피 추천 접근 방식은 대부분 레시피 간의 유사성을 기반으로 합니다(Yang et al., 2017; Chen et al., 2020). 일부 접근법에서는 사용자 정보를 고려하려고 시도했지만(Freyne and Berkovsky, 2010; Forbes and Zhu, 2011; Ge et al., 2015; Vivek et al., 2018; Khan et al., 2019; Gao et al., 2020), 사용자 간의 관계 정보를 무시한 채 사용자 간의 중복된 평가 레시피만을 기반으로 유사한 사용자를 정의했습니다,

계층적 그래프 관심 네트워크를 통한 레시피 추천

이준 티엔¹, 장추수², 로널드

Front. 빅 데이터 4:778417.

DOI:
10.3389/fdata.2021.778417

¹<https://www.food.com>

또는 재료. 그럼에도 불구하고 음식에 대한 사용자 선호도는 복잡합니다. 사용자는 재료, 맛 또는 친구의 추천으로 인해 새로운 레시피를 시도하기로 결정할 수 있습니다. 따라서 신중한 레시피 추천은 이러한 모든 요소를 고려해야 합니다. 따라서 레시피 추천을 위해서는 관계형 정보를 인코딩하고 사용자, 레시피, 재료 간의 관계를 심층적으로 이해하는 것이 중요합니다.

이 논문에서는 관계형 정보를 레시피 추천에 활용하고자 합니다. 먼저 노드 간의 관계를 공식화하기 위해 이질적인 레시피 그래프를 구성합니다. 특히 각 레시피에 재료, 조리법, 사용자 평가가 포함된 레시피 코퍼스를 수집하는 것으로 시작합니다. 그런 다음 이 레시피 집합을 세 가지 유형의 노드(즉, 재료, 레시피, 사용자)와 이를 연결하는 네 가지 유형의 관계(즉, 재료-재료, 레시피-재료, 레시피-레시피, 사용자-레시피 관계)가 있는 레시피 그래프로 변환합니다. 구축된 레시피 그래프의 그림은 **그림 1**에 나와 있습니다. 레시피 그래프를 구축한 후, 관계 정보를 자연스럽게 추천에 반영하는 그래프 학습 접근법을 사용하여 레시피 추천 문제를 해결하고자 합니다.

특히, 계층적 그래프 관심 네트워크(Hierarchical Graph Attention Network)의 약자인 새로운 이기종 레시피 그래프 추천 모델인 *HGAT*를 제안합니다. *HGAT*는 사용자의 기록 행동과 선호도에 맞는 레시피를 추천할 수 있습니다. 특히 여러 신경망 모듈을 활용하여 레시피 히스토리, 레시피 콘텐츠, 관계 정보를 인코딩합니다. 구체적으로, 먼저 유형별 변환 매트릭스를 적용하여 각 노드와 관련된 이질적인 콘텐츠(예: 지침 및 영양소)를 모델링하고 이를 공유 임베딩 공간으로 변환합니다. 그런 다음 노드 수준의 관심 모듈을 설계하여 서로 다른 가중치로 이웃 노드를 인코딩합니다. 여러 유형의 노드와 에지가 있다는 점을 고려하여 이 모듈은 각 관계에 대해 개별적으로 실행되어 각 유형의 이웃 노드 정보를 포함하는 관계별 임베딩을 공식화합니다. 예를 들어, 레시피-재료, 레시피-레시피, 레시피-사용자 등 3개의 연결된 관계를 가진 레시피 노드가 주어지면 노드 수준 주의 모듈은 각 유형의 노드를 개별적으로 인코딩하고 3개의 관계별 임베딩을 학습합니다. 다음으로, 관계 수준 주의 모듈을 개발하여 생성된 모든 관계별 임베딩을 서로 다른 가중치로 결합하고 각 노드에 대해 업데이트된 임베딩을 얻습니다. 동일한 예시를 통해 설명하자면, 관계 수준 주의 모듈은 학습된 세 가지 관계별 임베딩을 하나로 병합하여 주어진 레시피 노드의 최종 임베딩을 나타

냅니다. 따라서 학습된 임베딩에는 이웃 노드의 정보뿐만 아니라 연결된 관계 정보도 포함됩니다. 마지막으로 학습된 사용자 및 레시피 임베딩을 기반으로 점수 예측자와 랭킹 기반 목적 함수를 도입하여 모델을 최적화합니다. 본 논문의 주요 기여를 요약하면 다음과 같습니다:

- 관계형 정보는 레시피에 대한 사용자 선호도를 이해하는 데 중요하다고 주장합니다. 또한

*HGAT*를 개발했습니다. *HGAT*는 노드 콘텐츠와 관계형 정보를 모두 캡처하여 적절한

권장 사항. *HGAT*는 유형별 변환, 노드 수준 주의, 관계 수준 주의 등 여러 신경망 모듈로 구성되어 있습니다.

- 저희는 모델의 성능을 평가하기 위해 광범위한 실험을 수행합니다. 그 결과 다음과 같은 우수성이 입증되었습니다.

레시피 추천을 위한 여러 기준 방법과 비교하여 *HGAT*를 제안합니다.

이 백서의 나머지 부분은 다음과 같이 구성됩니다. 섹션 2에서는 관련 연구를 검토합니다. 섹션 3에서는 제안한 모델을 설명합니다. 섹션 4에서는 레시피 추천에 대한 다양한 모델의 실험 결과를 제시하고 섹션 5에서 결론을 내립니다.

2. 관련 작업

이 연구는 음식 추천, 레시피 추천, 그래프 표현 학습 연구와 밀접한 관련이 있습니다.

음식 추천. 음식 추천은 레스토랑, 개별 음식, 식사, 레시피 등 사용자의 선호도와 개인화된 니즈를 충족하는 음식 목록을 제공하는 것을 목표로 합니다(Trattner and Elswailer, 2017a; Min et al., 2020). 음식 추천은 비교적 덜 연구된 연구 문제임에도 불구하고 상당한 양의 문헌이 존재합니다(Trattner and Elswailer, 2017a). 예를 들어, Sano 등(2015)은 실제 식료품점의 거래 데이

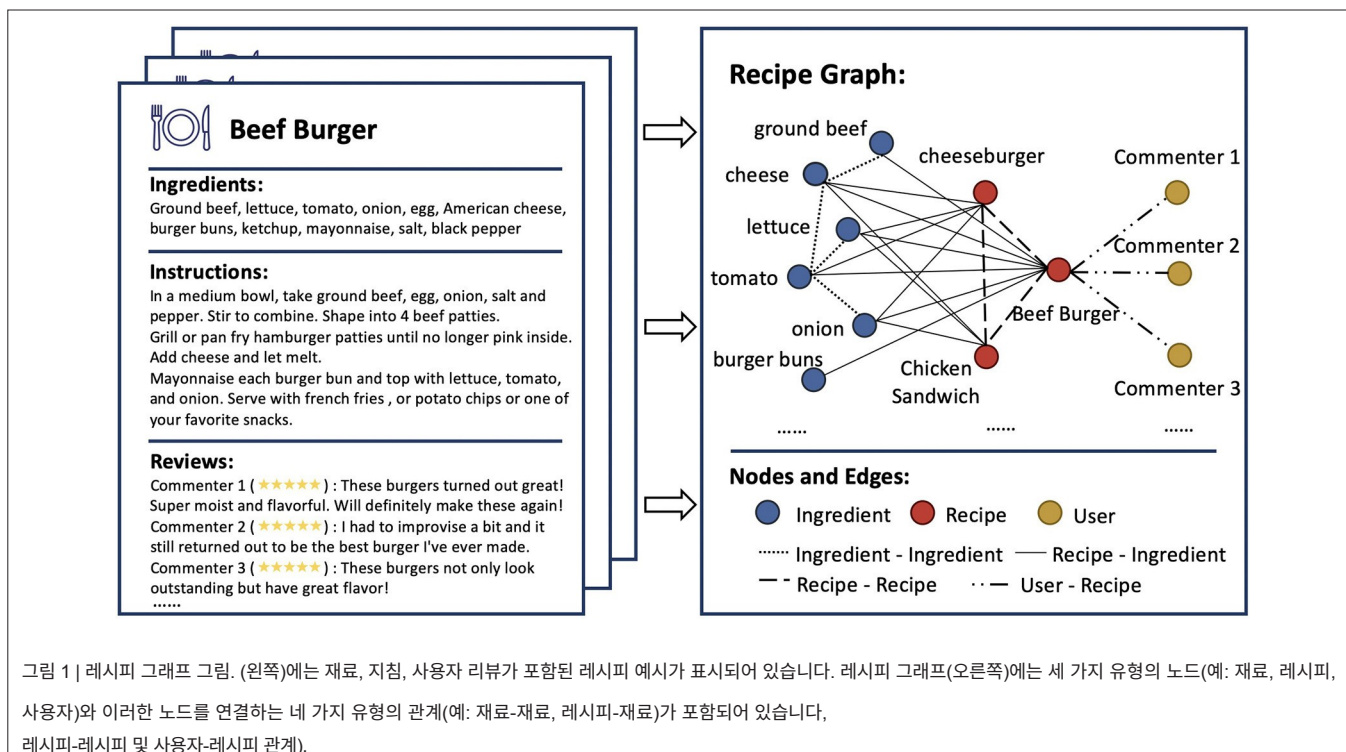
이 정보를 레시피 추천에 활용

문제를 발견하고 이를 해결하기 위한 그래프 학습 접근법을 제안했습니다.

Elsweiler(2017b)는 LibRec² 프레임워크의 9가지 저명한 추천 알고리즘을 사용하여 식사 계획과 레시피를 추천했습니다. 저희는 일상 생활과 가장 밀접한 관련이 있는 레시피 추천에 주로 초점을 맞췄습니다.

기존의 레시피 추천 접근 방식은 대부분 콘텐츠 기반, 즉 레시피 간의 유사성을 기반으로 레시피를 추천하는 방식입니다(Yang et al., 2017; Chen et al., 2020). 제안된 접근법 중 일부는 추천 절차에 사용자 정보를 포함시킵니다(예: 협업 필터링). 하지만 이러한 접근 방식은 사용자, 레시피 또는 재료 간의 관계 정보를 무시하고 평가가 겹치는 레시피를 기준으로 유사한 사용자만 고려했습니다(Freyne and Berkovsky, 2010; Forbes and Zhu, 2011; Ge et al., 2015; Vivek et al., 2018; Khan et al., 2019; Gao et al., 2020). 예를 들어, Yang 등(2017)은 항목별 및 쌍별 레시피 이미 지 비교를 기반으로 음식 선호도를 학습하는 프레임워크를 개발했습니다. Ge et al. (2015)는 레시피 추천을 위해 사용자 평가와 태그를 융합하는 행렬 인수분해 접근법을 활용했습니다. 한 편, 구축된 그래프를 기반으로 레시피 추천을 시도한 연구도 있지만, 사용자 정보는 포함되지 않았습니다(Li et al., 2010; Teng et al., 2012; Adaji et al., 2018). 예를 들어, Adaji 등(2018)은 같은 사람이 두 개의 레시피를 리뷰한 경우 두 레시피가 연결되는 그래프를 기반으로 사용자에게 레시피를 추천했습니다. Li 등(2010)은 노드가 요리 동작 또는 재료인 요리 그래프를 구성했습니다.

²<https://github.com/guoguibing/librec>



를 생성하고 유사성을 기반으로 레시피를 추천합니다. (2012)는 두 가지 유형의 재료 그래프를 구축하여 재료의 대체 또는 보완을 기반으로 레시피 쌍을 예측합니다. (2019)는 지식 기반 질문 답변 방식을 활용하여 재료를 기반으로 레시피를 추천했습니다. 본 연구에서는 사용자 정보가 포함된 이질적인 레시피 그래프를 통해 관계 정보를 모델링하려고 합니다. 따라서 그래프 학습 접근 방식은 관계 정보를 자동으로 인코딩하고 그에 따라 사려 깊은 추천을 할 수 있습니다.

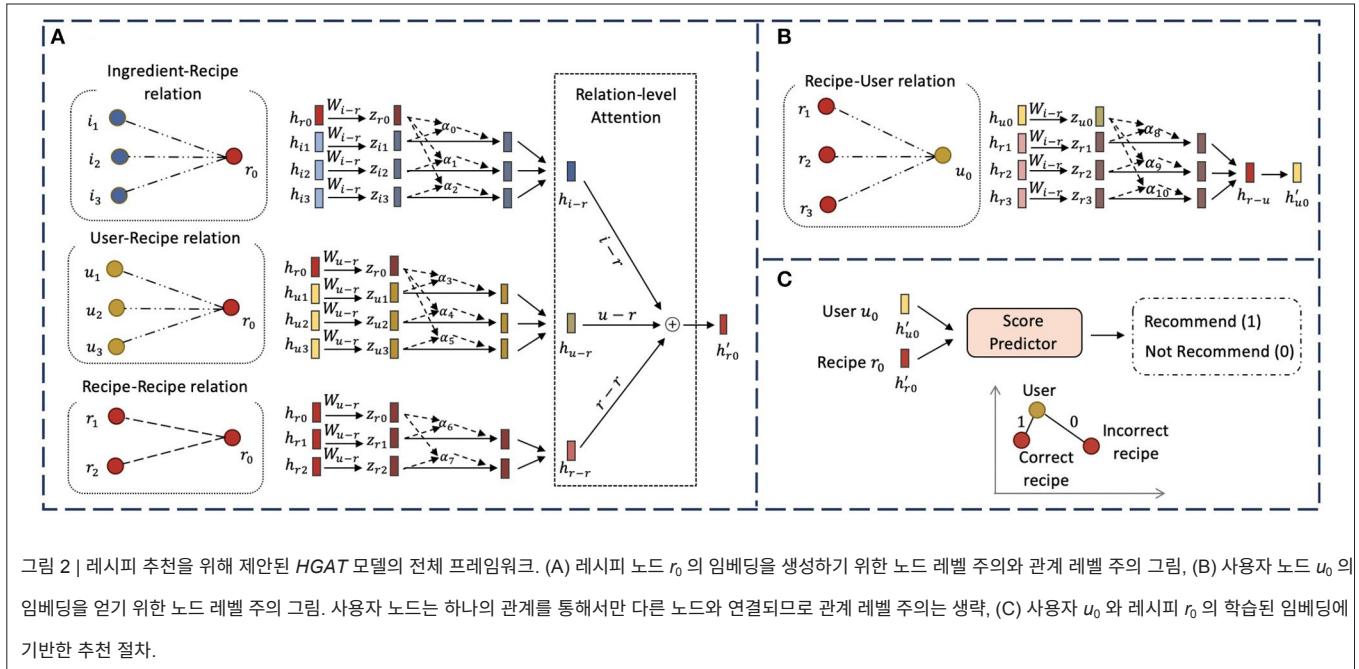
사용자 행동 모델링. 사용자 행동 모델링은 추천 영역에서 널리 연구되고 있습니다. 예를 들어, Zhou 등(2018)은 추천을 위해 모든 유형의 행동을 여러 잠재 의미 공간에 투영하는 주의 기반 사용자 행동 모델링 프레임워크를 제안했습니다. Elkahky 등(2015)은 웹 검색 기록과 검색어를 포함한 풍부한 기능 세트를 사용하여 사용자를 대표하여 콘텐츠 기반 추천 시스템을 제안했습니다. Abel 등(2011)은 사용자 프로필이 시맨틱 보강을 통해 어떤 이점을 얻는지 분석하고 개인화된 뉴스 추천 시스템에서 다양한 사용자 모델링 전략을 비교했습니다. 음식 추천 분야에서는 Zhang 등(2016)이 식사 행동에 대한 사용자 피드백을 레스토랑 추천에 활용했습니다. Musto 등(2020)은 사용자의 체질량지수(BMI), 식이 제약과 같은 개인적 요인에 초점을 맞춰 음식 및 사용자 건강 관

련 특성에 대한 지식을 기반으로 한 추천 전략을 개발했습니다. 이 연구에서는 사용자 이력 행동과 레시피에 대한 평점과 같은 사용자 피드백을 통합하여 사용자의 관심사와 선호도를 고려한 추천을 제공합니다. **그래프 표현 학습.** 그래프 표현 학습은 가장 인기 있는 데이터 마이닝 주제 중 하나가 되었습니다.

에 대한 연구가 활발히 진행되고 있습니다(Wu et al., 2021). 그래프 구조 데이터를 인코딩하기 위해 많은 그래프 표현 학습 접근법(Perozzi et al., 2014; Dong et al., 2017; Hamilton et al., 2017; Kipf and Welling, 2017; Schlichtkrull et al., 2018; Velickovic et al., 2018)이 제안되었습니다. 이들은 각 노드와 연관된 콘텐츠 정보와 그래프의 관계 정보를 활용하여 벡터화된 임베딩을 학습하고, 이를 추천과 같은 다양한 그래프 마이닝 작업에 활용합니다. 예를 들어, DeepWalk(Perozzi et al., 2014)는 무작위 걷기 세트를 SkipGram 모델에 입력하여 노드 임베딩을 학습했습니다(Mikolov et al., 2013). 그래프의 노드는 인접 노드가 컨텍스트 정보로 사용되는 각 보행에 대해 동시에 훈련되었습니다. matapath2vec(Dong et al., 2017)은 메타 경로 기반 랜덤 워크를 수행하고 이질적인 그래프를 임베딩하기 위해 SkipGram 모델을 활용했습니다. GAT(Velickovic et al., 2018)는 메시지 전달에 주의 메커니즘을 활용하여 서로 다른 주의도를 가진 이웃 노드의 특징을 집계하여 동종 그래프에서 노드 임베딩을 학습했습니다. 본 연구에서는 이기종 그래프에 적용하여 뛰어난 성능을 달성할 수 있는 다양한 수준(즉, 노드 수준과 관계 수준)의 주의 메커니즘을 활용하는 계층적 그래프 주의 네트워크를 제안합니다.

3. 제안 모델

이 섹션에서는 새로운 계층적 그래프 주의 네트워크 모델인 HGAT에 대해 설명합니다. 그림 2에서 볼 수 있듯이, 이 모델에는 몇 가지 주요 구성 요소가 포함되어 있습니다. 먼저 유형별 변환 행렬을 적용하여 다양한 입력 특징을 취하고 다음과 같이 투영합니다.



를 공유 임베딩 공간으로 이동합니다. 그런 다음 노드 수준의 주의 모듈을 설계하여 각 관계로 연결된 이웃 노드를 인코딩하여 관계별 임베딩을 학습합니다. 다음으로, 모든 관계별 임베딩을 결합하고 각 노드에 대해 업데이트된 임베딩을 얻기 위해 관계 수준 주의 모듈을 개발합니다. 마지막으로, 점수 예측기와 순위 기반 목적 함수를 사용하여 모델을 최적화합니다.

3.1. 유형별 변환

이전 연구(살바도르 외, 2017; 마린 외, 2019)는 다음과 같습니다. 조리 지침을 제공하는 데 필요한 것으로 나타났습니다. 요리 과정에 대한 변별적 이해. Marin 등의 연구(Marin et al., 2019)에 따라 사전 학습된 건너뛰기 명령 임베딩을 명령 표현으로 사용한 다음, 이러한 표현을 평균화하여 각 레시피에 대해 하나의 입력 원시 특징을 얻습니다. 예를 들어, M 문장이 포함된 레시피가 있다고 가정합니다.

인스트럭션 임베딩 $\{x_{ins,1}, x_{ins,2}, \dots, x_{ins,M}\}$ 에 대한 입력 기능으로 평균 명령어 임베딩 x_{ins} 을 계산합니다.

레시피 노드:

$$x_{ins} = \frac{1}{M} \sum_{i=1}^M x_{ins,i}, \quad (1)$$

또한 각 성분의 영양소를 벡터로 공식화한 다음 이 영양소 벡터를 성분 노드의 콘텐츠 정보로 사용합니다. 이 콘텐츠 정보를 성

서로 다른 노드 유형(예: 레시피, 재료, 사용자)에 대한 입력 특징의 이질성으로 인해 유형 φ_i 의 노드 v_i 가 주어지면 입력 특징을 동일한 임베딩 공간에 투영하기 위해 유형별 변환 행렬 $W_{\varphi i}$ 를 도입합니다. 구체적으로, 각 노드 v_i 에 대해 투영 프로세스는 다음과 같이 공식화됩니다:

$$\begin{aligned} x_i &= \begin{cases} x_{ins} & \text{if } \varphi_i = \text{성분} \\ x_{u0} & \text{if } \varphi_i = \text{사용자} \end{cases} \\ h_i &= W_{\varphi i} x_i, \end{aligned} \quad (2)$$

여기서 $W_{\varphi i} \in \mathbb{R}^{d \times d_{\varphi i}}$ 는 변환 행렬이고, x_i 는 차원 $d_{\varphi i}$ 를 갖는 v_i 의 입력 특징이며, h_i 는 차원 d 를 갖는 v_i 의 투영 특징입니다. 즉, 이 유형별 변환 작업을 사용하면 명령, 성분 및 사용자 임베딩은 공유 임베딩 공간에 있으므로 모델은 임의의 유형의 입력 피처를 사용할 수 있습니다.

3.2. 노드 수준 주의

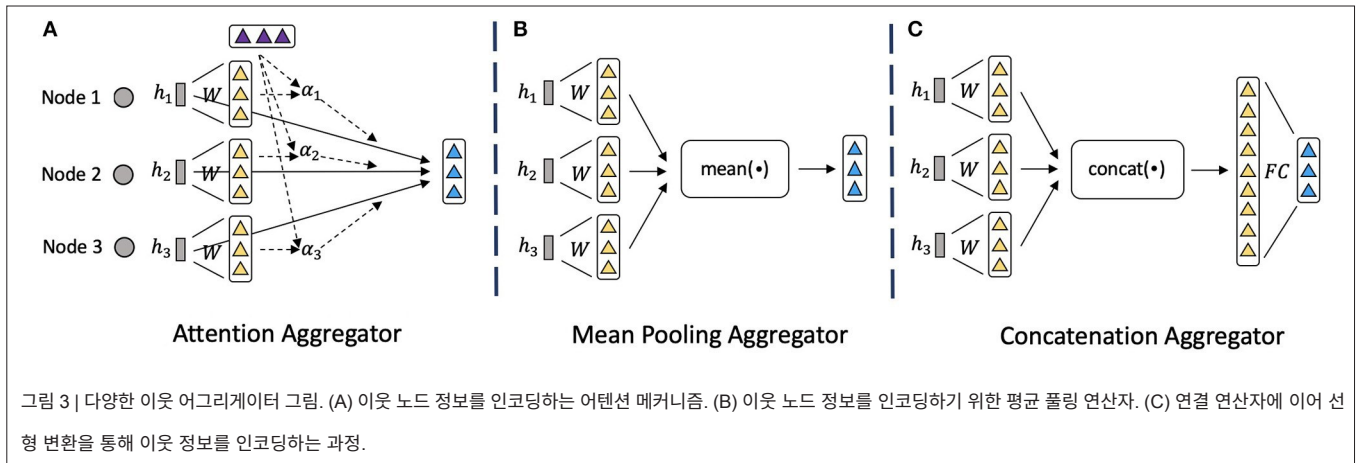
이웃 노드 정보를 인코딩하고 융합하기 위해 다음과 같이 제안합니다.

노드 수준의 관심도 모듈로, 그림 3A와 같이 관심도 애그리게이터를 기반으로 합니다. 평균과 비교

본 노드에 대한 입력 기능 x_{ing} 로 표시합니다. 사용자 노드를 표현하기 위해 Food.com의 대부분의 사용자는 자신에 대한 자세한 정

보(예: 설명, 위치, 선호도 또는 인구통계학적 정보)를 제공하지 않았으며 이러한 개별 정보를 크롤링하면 개인정보 보호정책을 위반할 수 있으므로 사용자 노드의 입력 기능으로 xavier normal random initialized feature(Glorot and Bengio, 2010)를 사용하며, 이를 x_{user} 로 표시합니다.

폴링 어그리게이터(그림 3B)와 연결 어그리게이터(그림 3C)가 널리 사용되어 왔지만(Hamilton et al., 2017), 그 특징들을 간단히 결합한 관심도 어그리게이터는 각 이웃 노드의 중요성을 학습하고 현명하게 융합할 수 있습니다. 구체적으로, 노드 v_i 에 대해 먼저 노드 수준 주의도를 사용하여 v_i 에 연결되는 각 관계 r 에 대한 각 관계별 임베딩 $h_{i,r}$ 을 계산합니다. 관계별 임베딩 $h_{i,r}$ 을 얻는 방법을 설명하기 위해 먼저 단일 노드 수준 주의도를 설명하여 각 관계에 대한 노드 수준 주의도를 계산할 것입니다. 노드입니다. 레이어 $l + 1$ 에서 $h_{i,r}$ 를 계산하기 위해 노드 수준 관심 레이어의 입력은 레이어의 노드 임베딩 집합입니다.



$l: \{h_1, h_2, \dots, h_{N_{i,r}}\} \in \mathbb{R}^{d_l}$, 여기서 $N_{i,r}$ 은 관계 r 을 통해 v_i 에 연결되는 이웃 노드의 수를 나타내고, d_l 는 레이어 l 의 임베딩 차원입니다.

입력 피처를 상위 레벨 피처로 변환하기에 충분한 표현력 $z_i \in \mathbb{R}^{d_{l+1}}$, 여기서 d_{l+1} 는 레이어 $l+1$ 의 임베딩 차원이며, 관계 r 에 대한 공유 선형 변환 가중치 행렬 $W_r \in \mathbb{R}^{d_l \times d_{l+1}}$ 이 적용됩니다:

$$z_i = W_r \cdot h_i. \quad (3)$$

노드 v_i 와 v_j 에 대해 각각 중개 기능 z_i, z_j 을 사용하여 v_i 와 v_j 사이에서 정규화되지 않은 관심도 점수 e_{ij} 를 계산하여 v_j 와 v_i 의 중요성을 나타냅니다. 계산 프로세스는 다음과 같이 정의됩니다:

$$e_{ij} = \text{LeakyReLU}(W_{ij} \cdot (z_i \parallel z_j)), \quad (4)$$

여기서 $W_{ij} \in \mathbb{R}^{2d_{l+1}}$ 은 연결 연산자를 나타냅니다. 이는 v 사이의 관심을 나타내는 가중치 벡터입니다.

와 v_j . 이론적으로 우리 모델은 관계 정보를 고려하지 않고 모든 노드의 다른 모든 노드에 대한 관심도를 계산할 수 있습니다. 메시지 전달 프로토콜을 고려하여 그래프 구조를 인정하고 그래프에서 v_i 와 v_j 사이에 에지가 존재할 경우에만 e_{ij} 를 계산하는 마스크 어텐션(Velickovic et al., 2018)을 수행합니다. 즉, v_i 의 1차 이웃 노드(v_i 포함)에만 집중합니다. 여러 노드에서 계수를 쉽게 비교할 수 있도록 소프트맥스 함수를 사용하여 e_{ij} 를 정규화합니다. 정규화된 노드 수준의 관심도 벡터 α_{ij} 는 다음과 같이 계산됩니다:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N_{i,r}} \exp(e_{ik})}, \quad (5)$$

여기서 σ 는 비선형 활성화 함수입니다(실험에서는 ReLU 를 사용했습니다). 이전 연구(Vaswani et al., 2017)에서 영감을 받아 단순히 단일 주의 함수를 수행하는 대신 노드 수준의 주의력을 다중 헤드 주의력으로 확장하여 모델과 훈련 프로세스를 더욱 안정적으로 만듭니다. 특히, 노드 수준의 주의력을 M 회 병렬로 계산한 다음, 그 결과물을 연결하여 최종 학습된 관계별 임베딩 $h_{i,r}$ 에 투영합니다. 계산 과정은 다음과 같이 공식화됩니다:

$$h_{i,r} = \sum_{m=1}^M \sigma \left(\sum_{j \in N_{i,r}} \alpha_{ij} \cdot z_j \right) \cdot W_o, \quad (7)$$

여기서 $W_o \in \mathbb{R}^{K_d \times m \times d_{l+1}}$ 은 학습 가능한 가중치 행렬이고, d_m 는 주의력 헤드의 차원으로 $d_m = d_{l+1} / M$ 입니다. 따라서 각 헤드의 치수를 줄이면 총 비용이 감소합니다.

의 다중 헤드 주의 계산은 차원 d 가 있는 단일 헤드 주의의 계산과 유사합니다.

3.3. 관계 수준 관심

그래프에서 노드는 여러 유형의 관계(예: 레시피와 사용자, 레시피와 재료, 레시피와 레시피)를 통해 다른 노드와 연결되어 있지만, 각 관계별 노드 임베딩은 하나의 관점에서만 노드 정보를 나타낼 수 있습니다. 따라서 보다 포괄적인 노드 임베딩을 학습하려면 노드 주변의 관계별 노드 임베딩을 융합해야 합니다. 관계를 선택하고 여러 관계별 노드 임베딩을 융합하는 문제를 해결하기 위해, 서로 다른 관계의 중요성을 학습하고 자동으로 융합하는 관계 수준 주의 모듈을 제안합니다.

를 사용합니다. 특히 관계 수준 관심 모듈을 사용하여 다음과 같은

그런 다음 α_{ij} 를 계수로 사용하여 이웃 노드 피처를 선형적으로 결합하고 관계별 임베딩 $h_{i,r}$ 을 생성합니다. 이 과정은 다음과 같이 공식화됩니다:

$$h_{i,r} = \sigma \sum_{j \in N_{i,r}} \alpha_{ij} z_j, \quad (6)$$

최종 노드 임베딩 H_i .

먼저 공유 비선형 가중치 행렬 W_R 를 사용하여 관계별 노드 임베딩을 변환합니다. 그런 다음 관계 수준 중개 벡터 q 를 사용하여 다음과의 유사도를 계산합니다.

변환된 임베딩을 계산하며, 이는 각 관계별 노드 임베딩의 중요도로도 사용됩니다. 그런 다음 모든 관계별 노드 임베딩의 중요도를 평균하여 관계에 대한 중요도를 계산합니다.

r 를 사용하여 노드 v_i 에 대한 중요도 점수 $w_{i,r}$ 를 생성합니다(i). 프로세스는 다음과 같습니다:

$$w_{i,r} = \frac{1}{|V_r|} \sum_{v_r \in V_r} q^T - \tanh(W_R - h_{i,r} + b), \quad (8)$$

여기서 $W_R \in \mathbb{R}^{d+1 \times d+1}$ 은 비선형 가중치 행렬이고, $b \in \mathbb{R}^{d+1}$ 입니다.

는 바이어스 벡터, $q \in \mathbb{R}^{d+1}$ 은 관계 수준 중개자입니다.

벡터, V_r 는 관계 r 의 노드 집합을 나타냅니다. 서로 다른 관계에

서 계수를 비교할 수 있도록 하기 위해 다음을 수행합니다.

$w_{i,r}$ 를 정규화하여 관계 수준 관심도 벡터 $\beta_{i,r}$ 를 얻습니다.

소프트맥스 함수를 사용하여 각 관계 R 를 정규화합니다. 정규화 프로세스는 다음과 같이 공식화됩니다:

$$\beta_{i,r} = \frac{\exp(w_{i,r})}{\sum_{R \in R} \exp(w_{i,r})}, \quad (9)$$

여기서 R_i 은 노드 v_i 의 연관 관계를 나타냅니다. 여기서 생성된 관심도 벡터 $\beta_{i,r}$ 는 노드 v_i 에 대한 관계 r 의 기여도로 설명할 수 있습니다. $\beta_{i,r}$ 가 높을수록 관계 r 이 더 중요하다는 것을 알 수 있습니다. 관계마다 학습 목표에 기여하는 정도가 다를 수 있으므로 각 관계에 대한 관계 수준 관심도 벡터는 그에 따라 다른 가중치를 가질 수 있습니다. 따라서 관계별 노드 임베딩(i,r)과 관계 수준 관심도를 융합하여 최종 노드 임베딩(i)을 얻습니다. 이 과정은 다음과 같이 설명되어 있습니다:

$$h_i = \sum_{r=1}^R \beta_{i,r} \cdot h_{i,r}. \quad (10)$$

여기서 최종 노드 임베딩 h_i 은 관계별 노드 임베딩의 최적 가중치 조합으로 해석할 수 있으며, 각 관계별 노드 임베딩은 동일한 관계를 공유하는 노드 임베딩의 최적 가중치 조합으로 해석할 수 있습니다.

3.4. 레시피 추천

위에서 우리는 레이어 $l+1$ 에서 노드 임베딩 h_i 을 전파하고 학습하는 방법에 대해 설명했습니다. 그러나 이러한 정보 노드 임베딩을 활용하고 레시피를 만드는 방법은 다음과 같습니다.

추천은 여전히 어려운 과제입니다. 이 섹션에서는 이러한 임베딩을 활용하여 적절한 추천을 만드는 방법을 소개합니다. 구체적으로, 노드 수준 및 관계 수준 어텐션이 모두 인코딩된 GNN의 L 계층을 통해 노드 임베딩을 전파하고 업데이트한다고 가정하면, 여

연결은 서로 다른 관점에서 노드 정보를 나타냅니다. 따라서 노드 정보를 반영하는 데 기여하는 바가 다릅니다. 따라서 다음과 같이 연결합니다.

를 사용하여 각 노드에 대한 최종 임베딩을 개발합니다. 연결 프로세스는 다음과 같이 나타낼 수 있습니다:

$$\hat{h} = \sum_{i=1}^L \parallel h_{ins}^i \varphi_i = \text{레시피} \quad (12)$$

$\parallel h_{ins}^i \varphi_i = \text{성분}$
 $\parallel h_{ins}^i \varphi_i = \text{사용자}$
 $\parallel h_{ins}^i \varphi_i = \text{항상}$

여기서 L 은 레이어를 나타내며 연결 연산입니다. 따라서 저희는 노드 임베딩의 마지막 레이어를 강화할 뿐만 아니라 이전 전파 계층의 임베딩을 통해 전파 범위를 감독할 수 있습니다. 여기서 L 은 하이퍼파라미터이며, 각 계층에서 각 노드에 대해 생성된 L 표현을 얻게 됩니다. 예를 들어, 유형이 φ_i 인 노드 v_i 가 주어지면 학습된 노드 임베딩 \hat{h} 는 $\{h^1, h^2, \dots, h^L\}$ 로 나타낼 수 있습니다.

여기서는 단순성과 효율성을 위해 서로 다른 임베딩 레이어를 결합하는 데 연결만 적용합니다(Xu et al., 2018). 하지만 최대 풀링, LSTM(Hochreiter and Schmidhuber, 1997) 또는 가중 평균과 같은 다른 연산도 활용할 수 있습니다. 이러한 애그리게이터들은 임베딩을 집계할 때 서로 다른 가정을 제시합니다. 이 부분은 향후 작업에서 살펴볼 수 있도록 남겨두었습니다.

마지막으로, 점수 예측자를 활용하여 학습된 사용자 임베딩과 레시피 임베딩을 기반으로 추천을 합니다. 특히 $임베딩_u$ 인 사용자 u 와 $임베딩_r$ 인 레시피가 주어지면 점수 예측기는 이를 입력으로 받아 모델이 이 레시피를 사용자에게 추천해야 하는지 여부를 나타내는 점수를 생성할 수 있습니다. 점수는 0에서 1 사이의 범위이며, 점수가 1에 가까울수록 레시피를 추천해야 하며 그 반대는 추천하지 않아야 함을 의미합니다. 추천 프

여기서 $\varphi_i \in \{ins, ing, user\}$. 프로세스는 다음과 같이 공식화할 수 있습니다. 다음과 같이 설정합니다:

$$\begin{aligned} \hat{h}_i &= \begin{cases} \{h_{ins}^1, h_{ins}^2, \dots, h_{ins}^L\}, & \text{if } \varphi_i = \text{레시피} \\ \{h_{ing}^1, h_{ing}^2, \dots, h_{ing}^L\}, & \text{if } \varphi_i = \text{성분} \\ \{h_{user}^1, h_{user}^2, \dots, h_{user}^L\}, & \text{if } \varphi_i = \text{user} \end{cases} \quad (11) \end{aligned}$$

서로 다른 레이어에서 생성된 표현은 서로 다른 순서를 통해 전달된 메시지의 조합을 강조하기 때문에

로세스는 다음과 같이 시연됩니다:

$$s_{u,r} = sp(h_u, h_r) \quad (13)$$

여기서 sp 는 점수 예측자이고 $s_{u,r}$ 는 사용자 u 에 대한 레시피 r 의 추천 점수입니다. 내부 곱, 코사인 유사성, 다층 퍼셉트론 등 여러 함수를 sp 에 대해 추가로 비교합니다. 가장 높은 성능을 렌더링하는 모델(즉, 내적 곱)이 선택됩니다. 자세한 내용은 섹션 4.7.3에 설명되어 있습니다.

모델 파라미터를 학습하기 위해 순위 기반 목적 함수를 사용하여 모델을 훈련합니다. 특히, 사용자-레시피 관계로 연결된 노드 간의 추천 점수를 임의의 사용자 및 레시피 노드 쌍 간의 점수와 비교하는 것이 목표입니다. 예를 들어, 사용자 노드 u 와 레시피 노드 r 을 연결하는 에지가 주어졌을 때, u 와 r 사이의 추천 점수가 무작위로 샘플링된 네거티브 레시피 노드 r' 사이의 점수.

목표 L 을 다음과 같이 공식화합니다:

$$L = \sum_{u \in U, r, r' \in N_u} \max(0, 1 - s_{u,r} + s_{u,r'}), \quad (14)$$

여기서 $s_{u,r}$ 와 $s_{u,r'}$ 는 사용자 간의 추천 점수입니다. u 와 올바른 레시피 r 및 잘못된 레시피 r' 을 각각 입력하세요, U 는 사용자 집합을 나타내고, N_u 은 사용자 u 의 레시피 이웃을 나타냅니다.

4. 실험

이 섹션에서는 다음과 같은 연구 질문에 답하기 위해 광범위한 실험을 수행합니다:

- **RQ1:** 레시피 추천에 있어 다양한 기준 방법과 비교했을 때 *HGAT*의 성능은 어떤가요?
- **RQ2:** 노드 수준 관심도 또는 관계 수준 관심도와 같은 다양한 구성 요소는 모델 성능에 어떤 영향을 미치나요?
- **RQ3:** 임베딩 차원 수 및 전파 레이어 수와 같은 다양한 하이퍼 파라미터가 모델 성능?

4.1. 데이터 세트

Reciptor(Salvador et al., 2017; Li and Zaki, 2020)에서 레시피를 추출하여 포괄적인 정보(즉, 각 재료에 대한 양과 단위가 표시된)가 있는 레시피만 사용합니다. 이러한 모든 레시피는 온라인에서 가장 큰 레시피 공유 플랫폼 중 하나인 *Food.com*에서 수집합니다. 실제 사용자와 레시피 간의 상호 작용을 공식화하기 위해 사용자 평가를 크롤링합니다.

플랫폼의 각 레시피에 대해 이 정보를 활용합니다. 레시피 추천 작업에 사용합니다. 각 재료에 대해

가 데이터 세트에 나타나면, 이를 USDA 영양 데이터 세트(미국 농무부, 2019)와 매칭하여 영양 정보를 얻습니다. 다음으로 레시피, 사용자, 재료를 각각 '레시피', '사용자', '재료' 유형을 가진 노드로 변환하여 레시피 그래프를 구축합니다. 그런 다음 이러한 노드 사이에 네 가지 유형의 에지를 구축하여 다음을 수행합니다.

연결합니다. 특히, 먼저 각 레시피와 해당 레시피의 가장자리가 있는 재료는 레시피-재료 관계로 표시되며, 각 재료의 무게는 가장자리 무게로 사용됩니다.

그런 다음 레시피 노드를 FoodKG(Haussmann et al., 2019)의 유사도에 따라 연결하고, 그 점수는 Reciptor(Li and Zaki, 2020)에서와 같이 에지 가중치로 사용됩니다. 또한 FlavorGraph(Park et al., 2021)의 정규화된 점 상호 정보(NPMI)(Bouma, 2009)를 사용하여 재료 노드들을 공동 발생 확률에 따라 연결하며, 이는 *Kitchenette*(Park et al., 2019)에서 볼 수 있듯 재료 노드들을 연결합니다. 또한 상호작용을 기반으로 사용자와 레시피 간의 에지를 구성하고, 평점은 에지 가중치로 처리합니다. 구축된 레시피 그래프의 통계는 표 1에 나와 있습니다.

표 1 | 빌드 레시피 그래프 통계.

구성 요소	이름	번호	데이터 소스
노드	사용자	7,959	Food.com
	레시피	68,794	수신자 [리와 자키 (2020)]
	재료	8,847	수신자 [리와 자키 (2020)]
가장자리	사용자 레시피	135,353	Food.com
	레시피 레시피	647,146	FoodKG [Haussmann 외. (2019)]
	레시피 재료	463,485	수신자 [리와 자키 (2020)]
	성분 - 성분	146,188	FlavorGraph [Park et al. (2021)]

$$Recall@K = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \frac{|RE(u) \cap GT(u)|}{|GT(u)|} \quad (15)$$

여기서 U_{test} 는 평가용 테스트 데이터의 사용자 집합이고, $RE(u)$ 는 사용자 u 에 대한 상위 K 추천 목록을 나타내며, $GT(u)$ 는 사용자 u 에 대한 기준 데이터 레시피 집합을 나타냅니다.

- **MRR@K.** 추천 목록의 순위 품질을 측정하며, 이는 다음과 같이 정의됩니다:

$$MRR@K = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \frac{1}{|GT_K(u)|} \sum_{v \in GT_K(u)} \frac{1}{r(v)} \quad (16)$$

4.2. 실험 설정

We employ the leave-one-out method to evaluate the model performance, which is widely utilized in existing recommendation studies (He et al., 2016, 2017; Bayer et al., 2017; Jiang et al., 2019). 구체적으로, 각 사용자에게 대해 하나의 긍정 레시피를 테스트 데이터로, 하나의 긍정 레시피를 검증 데이터로 남겨두고 나머지 긍정 레시피를 학습에 사용했습니다. 테스트 기간에는 각 사용자에게 대해 100개의 부정 레시피를 무작위로 샘플링하고 회상률과 평균 상호 순위(MRR) 지표를 사용하여 모델 성능을 평가했습니다. K 는 1에서 10까지의 범위에서 $top@K$ 로 성능을 보

- **Recall@K.** 이 값은 상위@K 추천 목록에서 올바른 레시피가 검색되는 비율을 계산하여 보여줍니다.

여기서 $GT K(u)$ 는 사용자 u 에 대한 상위@K 추천 목록에 나타나는 기준 데이터 레시피를 나타내고, $r(v)$ 는 추천 목록에서 해당 레시피의 순위 위치를 나타냅니다.

4.3. 기준 방법

기준 추천 방식, 레시피 표현 학습 방식, 그래프 임베딩 모델을 포함한 7가지 기준 방법과 HGAT를 비교합니다.

- **BPR**(Rendle et al., 2009): 추천을 위한 경쟁적 쌍 행렬 인수분해 모델로, 레시피에 널리 사용되는 최신 알고리즘 중 하나입니다.
추천 작업을 수행합니다(Trattner and Elswiler, 2019).
- **IngreNet**(Teng et al., 2012): 인기 있는 재료 목록을 구축된 데이터에서 추출한 동시 발생 횟수로 대체하여 레시피를 추천하는 접근 방식입니다.
재료 네트워크. 공정한 비교를 위해 사용자 레시피 그래프에 GCN 계층을 적용하여 임베딩을 공동으로 학습합니다.
- **NeuMF**(He et al., 2017): 신경망을 사용하는 최첨단 신경 협업 필터링 모델 중 하나입니다.
사용자 및 항목 임베딩을 사용하여 비선형 기능 상호 작용을 캡처합니다.
- **matapath2vec** (Dong et al., 2017): 메타에 의해 유도되는 랜덤 워크 기반의 이기종 그래프 임베딩 방법.
경로를 추가합니다. 여기서는 메타 경로 *user-recipe-ingredient-recipe-user*를 사용합니다.

표 2 | Top@K 레시피 추천에 대한 다양한 모델의 성능.

메트릭	모델	K									
		1	2	3	4	5	6	7	8	9	10
Recall@K	BPR	2.78	4.06	5.21	6.45	7.53	8.36	9.18	10.20	11.01	12.15
	IngreNet	9.49	13.56	16.22	18.38	19.89	21.72	23.04	24.60	26.05	27.24
	NeuMF	2.46	4.64	6.37	8.19	10.06	11.57	13.10	14.61	15.96	17.29
	메타패스2백	6.96	11.67	15.76	19.06	21.31	23.53	25.23	26.82	28.43	29.97
	그래프 세이지	2.55	5.40	8.23	10.91	13.70	15.99	18.62	21.08	23.38	25.56
	GAT	2.89	5.91	8.96	11.82	14.55	17.49	20.02	22.58	25.12	27.33
	수신자	8.63	12.33	14.1	15.22	16.04	16.85	17.77	18.61	19.2	19.91
	HGAT	14.91	18.88	22.62	25.47	28.32	30.71	33.11	35.24	37.49	39.50
MRR@K	BPR	2.78	3.42	3.80	4.11	4.33	4.46	4.58	4.71	4.80	4.91
	IngreNet	9.49	11.52	12.41	12.95	13.25	13.56	13.75	13.94	14.10	14.22
	NeuMF	2.46	3.55	4.13	4.58	4.96	5.21	5.43	5.62	5.77	5.90
	메타패스2백	6.96	9.32	10.68	11.50	11.95	12.32	12.57	12.77	12.94	13.10
	그래프 세이지	2.55	3.98	4.92	5.59	6.15	6.53	6.90	7.21	7.47	7.68
	GAT	2.89	4.40	5.42	6.13	6.68	7.17	7.53	7.85	8.13	8.35
	수신자	8.63	10.48	11.07	11.35	11.51	11.65	11.78	11.88	11.95	12.02
	HGAT	14.91	16.90	18.14	18.86	19.43	19.82	20.17	20.43	20.68	20.89

- **GraphSAGE**(해밀턴 외., 2017): 로컬 이웃의 특징을 집계하고 샘플링하여 임베딩을 학습하는 그래프 신경망 모델입니다.
- **GAT**(Velickovic et al., 2018): 주의 메커니즘을 활용하여 집계하는 그래프 주의 네트워크 모델입니다. 동종 그래프의 이웃 정보.
- **리시버**(리와 자키, 2020): 세트 트랜스포머에 기반한 최첨단 레시피 임베딩 모델 중 하나이며 지침-재료 유사성 손실 및 지식 그래프 기반 삼중 손실. 공정한 비교를 위해 사용자와 레시피 임베딩을 공동으로 학습하기 위해 각 관계에 GCN 계층을 적용합니다.

각 노드에 뿌리를 둔 도보 수를 5, 음의 샘플 수를 5로 설정했습니다. DeepWalk, GAT, GraphSage를 포함한 동종 그래프 표현 학습 접근법의 경우 노드의 이질성을 무시하고 전체 그래프에 대해 알고리즘을 수행합니다. 공정한 비교를 위해 원래 설정을 따르고 임베딩 차원으로 600을 사용하기 때문에 수신자를 제외한 위의 모든 모델에 대해 임베딩 차원을 128로 설정했습니다.

4.4. 구현 세부 정보

제안된 모델 *HGAT*의 경우 학습률을 0.005로, 노드 수준의 주의 헤드 수를 4로, 숨겨진 크기를 128로, 스킵 명령 임베딩의 입력 차원을 1,024로, 성분 임베딩의 입력 차원을 46으로, 배치 크기를 1,024로, 훈련 에포크를 100으로 설정했습니다. Adam을 사용하여 모델을 최적화하고 (Kingma and Ba, 2014) 학습률은 매 에포크마다 $\gamma = 0.95$ 씩 기하급수적으로 증가합니다. 다음을 포함한 랜덤 워크 기반 그래프 표현 학습 알고리즘의 경우

DeepWalk와 메타패스2vec에서는 창 크기를 5, 도보 길이를 30,

4.5. 성능 비교(RQ1)

평가 지표로 리콜과 MRR을 사용합니다. 모든 모델의 성과는 표 2에 나와 있습니다. 가장 우수한 결과는 굵은 글씨로 강조 표시되어 있습니다. 표에 따르면, 모든 경우에서 HGAT 모델이 모든 기준선보다 우수한 성능을 보이는 것을 알 수 있습니다. 특히, BPR 및 NeuMF와 같은 기존의 협업 필터링 추천 접근 방식은 숨겨진 관계형 정보나 각 레시피와 연관된 재료를 고려하지 않기 때문에 성능이 좋지 않습니다. 레시피 추천 모델 IngreNet은 관계형 정보를 활용하기 위해 GCN을 통합한 후 괜찮은 성능을 얻었습니다. 리시버는 사용자 정보를 간과한 채 레시피에 대한 표현만 학습하기 때문에 성능이 좋지 않습니다. 사용자 임베딩을 공동으로 학습하기 위해 GCN 레이어를 적용하더라도 정보를 완전히 인코딩할 수 없습니다. 그러나 동종 그래프 표현 학습 접근법(예: GraphSage 및 GAT)은 Recall에서는 만족스러운 결과를 얻지만 MRR에서는 성능이 떨어집니다. 이는 그래프 구조를 모델링할 때 노드 유형 정보가 중요하기 때문입니다. 이 정보를 무시하면 모델이 포괄적인 노드 임베딩을 학습하지 못하고 반환된 추천 목록 내에서 올바른 레시피의 순위를 더 높게 지정하지 못합니다. 반대로, 이질적인 그래프 표현 학습 알고리즘인 메타패스2vec은 리콜과 MRR 모두에서 우수한 성능을 보입니다. 마지막으로, 제안된 모델인 HGAT는 레시피 콘텐츠, 고차 상호작용, 관계 정보를 통합하고 주의 메커니즘을 활용하여 다양한 유형의 노드와 관계를 인코딩함으로써 모든 기준 방법과 비교하여 가장 우수한 성능을 달성했습니다. 일반적으로 최상의 기준선과 비교했을 때, HGAT는 리콜 점수를 +5.42%, +5.32%, +6.40%, +6.41%, +7.01% 향상시킵니다, 7.18%, +7.88%, +8.42%, +9.06%, +9.53% 범위에서 +7.18%, +7.88%, +8.42%, +9.06%, +9.53% 를 각각 1부터 10까지 점수를 매겼습니다. MRR 점수의 경우,

표 3 | $Top@K$ 권장 사항에 대한 다양한 모델 변형 비교.

메트릭	모델	K									
		1	2	3	4	5	6	7	8	9	10
Recall@K	$HGAT_{mean}$	12.50	17.33	20.76	23.46	25.79	28.03	30.41	32.49	34.33	36.07
	$HGAT_{pool}$	13.67	18.26	21.52	24.59	27.06	29.26	31.24	33.50	35.34	37.24
	$HGAT_{nAtt}$	14.63	18.87	22.02	25.38	28.04	30.29	32.52	34.92	36.90	38.84
	$HGAT$	14.91	18.88	22.62	25.47	28.32	30.71	33.11	35.24	37.49	39.50
MRR@K	$HGAT_{mean}$	12.50	14.91	16.06	16.73	17.20	17.57	17.91	18.17	18.38	18.55
	$HGAT_{pool}$	13.67	15.96	17.05	17.82	18.31	18.68	18.96	19.24	19.45	19.64
	$HGAT_{nAtt}$	14.63	16.75	17.80	18.42	18.93	19.30	19.62	19.92	20.14	20.34
	$HGAT$	14.91	16.90	18.14	18.86	19.43	19.82	20.17	20.43	20.68	20.89

$HGAT$ 는 +5.42%, +5.38%, +5.73%, +5.91%씩 점수를 향상시킵니다, +6.18%, +6.26%, +6.42%, +6.49%, +6.58%, +6.67%에서 k 의 범위는 각각 1에서 10까지입니다. 이는 $HGAT$ 가 다른 모델에 비해 더 나은 레시피 추천을 얻을 수 있음을 보여줍니다.

4.6. 절제 연구(RQ2)

$HGAT$ 는 여러 신경망 모듈로 구성된 공동 학습 프레임워크입니다. 서로 다른 구성 요소가 모델 성능에 어떤 영향을 미칠까요? 이 질문에 답하기 위해 다음과 같은 여러 모델 변형의 성능을 평가하기 위한 제거 연구를 수행합니다:

- $HGAT_{mean}$: 노드 수준의 관심도나 관계 수준의 관심도 모두 사용하지 않는 모델 변형입니다. 대신 평균 연산자를 사용하여 이웃 노드 피처와 관계 기능.
- $HGAT_{pool}$: 노드 수준 관심도나 관계 수준 관심도를 사용하지 않는 모델 변형입니다. 대신 풀링 연산자를 사용하여 이웃 노드 특징과 평균을 결합합니다. 연산자를 사용하여 관계 기능을 결합합니다.
- $HGAT_{nAtt}$: 노드 수준의 주의력을 사용하여 이웃 노드 기능을 융합하고 평균 연산자를 사용하여 관계 기능을 결합하는 모델 변형입니다.
- $HGAT$: 노드 수준 주의와 관계 수준 주의를 모두 활용하는 제안된 모델입니다.

결과는 표 3에 나와 있습니다. 이 표에서

- $HGAT_{pool}$ 가 $HGAT_{mean}$ 보다 성능이 더 좋다는 것은 이웃 노드 정보를 집계하는 방식이 중요하며, 단순히 평균 연산자를 사

용하여 결합하는 방식이 중요하다는 것을 나타냅니다.

이웃 노드 메시지에서 일부 정보가 손실될 수 있습니다.

- $HGAT_{nAtt}$ 는 노드 수준 주의의 효과를 입증하고 평균을 사용하여 주의 메커니즘의 발전을 설명하는 $HGAT_{pool}$ 및 $HGAT_{mean}$ 보다 성능이 우수합니다.

또는 풀링 연산자를 사용하여 이웃 노드 정보를 집계합니다.

- 제안된 $HGAT$ 는 다음을 포함한 모든 모델 변형보다 성능이 뛰어납니다. $nAtt$, 통합이 다음을 보여줍니다.

관계 수준의 관심도를 더욱 향상시킬 수 있습니다.

성능. 이는 관계 수준 관심의 효과를 보여줍니다.

4.7. 파라미터 감도(RQ3)

하이퍼 파라미터에 대한 제안된 모델의 민감도를 추정하기 위해, 다양한 하이퍼 파라미터 설정에서 *HGAT*의 성능을 측정하기 위해 여러 가지 대비 실험을 수행했습니다. 먼저 데이터 모델링에서 일반적으로 중추적인 역할을 하는 임베딩 차원이 미치는 영향을 살펴보았습니다. 그런 다음 전파 계층 번호의 영향을 분석하여 관계형 정보 모델링의 중요성을 보여줍니다. 또한 다양한 점수 예측자가 추천 성과에 어떤 영향을 미치는지 연구합니다.

4.7.1. 다양한 임베딩 치수의 영향

그림 4에서 임베딩 차원 수에 따른 Recall@5, MRR@5, Recall@10, MRR@10의 성능을 보고합니다. 구체적으로 {16, 32, 64, 128, 256, 512} 내에서 임베딩 차원 수를 검색하고 제안된 HGAT와 두 가지 최상의 기준선(즉, 메타패스2vec 및 GAT)의 성능을 평가합니다. 그림에서:

- 임베딩 차원 수를 늘리면 모델 성능이 향상됩니다. 256개 차원을 사용할 때 모든 모델이 2개 지표에서 가장 높은 점수를 받았습니다. 이는 차원이 많을수록 노드 콘텐츠를 표현할 수 있는 용량이 더 커질 수 있기 때문입니다.
- 임베딩 차원 수를 512로 더 늘리면 과적합이 발생합니다. 이는 표현을 고차원 공간으로 변환하는 과정에서 노이즈가 발생했습니다.
- 임베딩 치수 수를 변경할 때 *HGAT*는 다양한 설정에서 다른 방법보다 일관되게 우수합니다. 이는 다른 접근 방식과 비교하여 *HGAT*의 기능을 보여줍니다.

4.7.2. 다양한 전파 계층의 영향

모델 깊이를 변경하고 성능을 테스트하여 제안된 *HGAT*가 여러 개의 임베딩 전파 레이어를 통해 이점을 얻을 수 있는지 조사합니다. 구체적으로 {1, 2, 3, 4}에서 레이어 번호를 검색합니다. 실험 결과는 다음에서 보고됩니다.

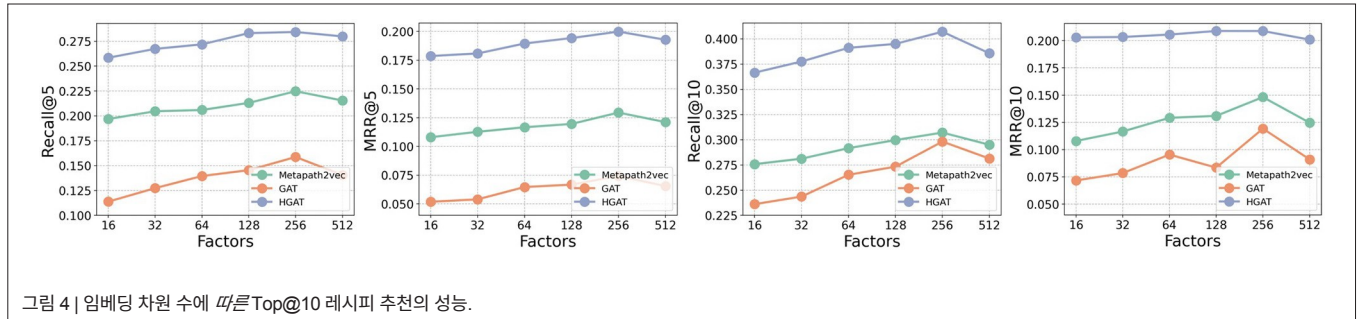


표 4 | 전파 계층에 따른 Top@K 권장 사항에 대한 HGAT의 성능.

메트릭	레이어	K									
		1	2	3	4	5	6	7	8	9	10
Recall@K	HGAT-1	14.57	17.62	19.15	20.47	21.70	22.92	24.24	25.54	26.95	28.13
	HGAT-2	14.91	18.88	22.62	25.47	28.32	30.71	33.11	35.24	37.49	39.50
	HGAT-3	15.19	19.71	23.47	26.45	29.17	31.90	34.55	36.86	39.26	41.30
	HGAT-4	14.73	19.50	23.28	26.80	29.45	31.71	34.33	36.54	38.62	41.09
MRR@K	HGAT-1	14.57	16.09	16.61	16.94	17.18	17.39	17.57	17.74	17.89	18.01
	HGAT-2	14.91	16.90	18.14	18.86	19.43	19.82	20.17	20.43	20.68	20.89
	HGAT-3	15.19	17.45	18.70	19.45	19.99	20.45	20.83	21.12	21.38	21.59
	HGAT-4	14.73	17.11	18.37	19.25	19.78	20.16	20.53	20.81	21.04	21.29

표 5 | 점수 예측 변수에 대한 Top@K 레시피 추천에 대한 HGAT의 성능.

메트릭	예측 변수	K									
		1	2	3	4	5	6	7	8	9	10
Recall@K	Cosine	14.89	17.67	19.96	22.01	24.04	26.08	28.19	30.10	32.23	34.15
	MLP	14.21	18.53	22.16	24.89	27.48	29.89	32.16	34.41	36.69	38.71
	내부 제품	14.91	18.88	22.62	25.47	28.32	30.71	33.11	35.24	37.49	39.50
MRR@K	코사인	14.89	16.33	17.09	17.61	18.01	18.35	18.65	18.89	19.13	19.32
	MLP	14.21	16.37	17.58	18.26	18.78	19.18	19.51	19.79	20.04	20.24
	내부 제품	14.91	16.90	18.14	18.86	19.43	19.82	20.17	20.43	20.68	20.89

표 4에서 HGAT-2는 2개의 임베딩 전파 레이어가 있는 모델을 나타내며, 그 외에는 유사한 표기가 사용됩니다. 표 4를 분석하면 다음과 같은 사실을 알 수 있습니다:

- HGAT에 더 많은 전파 계층을 장착하면 추천 성능이 크게 향상됩니다. 분명히 HGAT-2와 HGAT-3는 HGAT-1에 비해 지속적으로 개선되었습니다.

1을 초과하는 모든 경우에서 HGAT-1은 1차 이웃 노드만 고려합니다. 이러한 개선은 관계형 정보의 효과적인 모델링에 기인합니다. 관계형 정보는 2차 및 3차 연결에 의해 전달되며, 이러한 상호 작용을 인코딩하여 관계형 정보를 모델링할 수 있기

때문입니다.

- HGAT-3 위에 전파 계층을 더 쌓을 경우, HGAT-4가 데이터 세트에 과적합을 유발하는 것으로 나타났습니다. 이 너무 심층적인 아키텍처를 적용하면

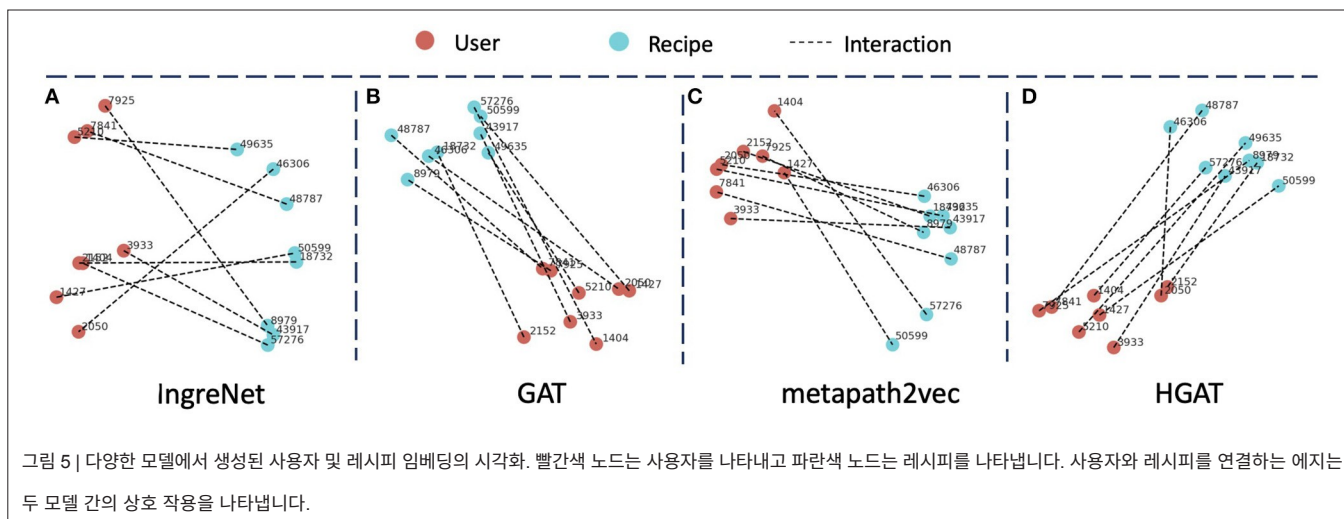
노이즈를 모델링에 추가합니다. *HGAT-3*와 유사한 결과를 통해 3개의 전파 계층을 수행하면 관계 정보를 캡처하는 데 충분하다는 것을 확인할 수 있습니다.

- 표 4의 결과와 표 2(*HGAT-2* 성능을 보고하는 표)를 비교하면 HGAT가 일관되게 다음과 같은 성능을 보입니다.

다른 방법보다 우수합니다. 이는 고차 상호작용과 관계 정보를 명시적으로 모델링하면 모델링을 크게 용이하게 하고 추천 작업의 성능을 더욱 향상시킬 수 있음을 실증적으로 보여줌으로써 *HGAT*의 효과를 다시 한 번 입증합니다.

4.7.3. 다양한 점수 예측 인자의 영향

모델에서 사용된 점수 예측자의 효과를 보여주기 위해 다양한 점수 예측자의 효과를 테스트하고 그 결과를 표 5에 보고합니다. 특히



코사인 유사도, 다층 퍼셉트론(MLP), 내적 곱 등 다양한 유사도 함수를 사용합니다. $K=1$ 에서 10 사이의 범위인 Top@K 추천에 대해 다양한 함수의 성능을 평가합니다. 표를 참조하세요:

- 내적 곱은 유사도 점수를 계산하는 데 가장 좋은 함수이며 코사인은 가장 나쁜 함수입니다. 이는 코사인 유사도가 2 사이의 각도 차이만 고려하기 때문일 수 있습니다. 벡터는 학습된 임베딩의 복잡성을 포착하지 못하는 반면, 내적 곱은 각도와 크기를 모두 고려합니다.
- MLP는 리콜에 따라 내부 제품에서 비슷한 성능을 달성하지만 MRR을 사용하여 평가할 때는 여전히 성능이 떨어집니다, 이는 논문(Rendle et al., 2020)의 연구 결과와 일치합니다. 이는 내부 제품의 호환성을 더욱 잘 보여줍니다.

4.8. 사례 연구: 비주얼라이제이션 임베딩

보다 직관적인 이해와 비교를 위해 8개의 사용자-레시피 상호작용 쌍을 무작위로 선택하고 t-SNE를 사용하여 임베딩의 시각화를 생성합니다(van der Maaten and Hinton, 2008). 그림 5에서 볼 수 있듯이 IngreNet의 성능이 좋지 않다는 것을 알 수 있습니다. 이 모델은 사용자와 레시피를 대략적으로 왼쪽과 오른쪽으로 구분할 수 있지만, 각 클러스터 내에는 간격이 존재합니다. 또한 클러스터 간의 경계가 무질서합니다. 이상적으로는 각 사용자-레시피 쌍 사이에 명확한 매핑을 개발해야 합니다. 즉, 각 쌍을 연결할 때 연결 선은 서로 평행해야 합니다["왕-남자=여자" 관계(Mikolov 외., 2013)와 유사]. GAT는 사용자와 레시피를 두 개의 응축 클러스터로 성공적으로 형성할 수 있지만, 그 사이의 평행선을 구성하는 데는 실패합니다. 메타패스2vec은 사용자와 레시피

피에 대해 두 개의 클러스터를 구축하고 어떤 의미에서 클러스터 간에 평행선을 설정합니다. 예를 들어 "1404-57276" 및 "1427-50599" 사용자-레시피 쌍이 있습니다. 그러나 형성된 클러스터는 응축되어 있지 않으며 일부 선만 서로 평행합니다. 마지막으로, 우리의 모델 HGAT는 사용자와 레시피를 2개의 응축된 클러스터로 쉽게 형성할 수 있으며 거의 모든

사용자-레시피 쌍입니다. 이는 우리 모델의 우수성을 더욱 입증합니다.

5. 결론

본 논문에서는 관계형 정보를 레시피 추천에 활용하는 방법을 제안합니다. 이를 위해 문제 해결을 위한 새로운 계층적 그래프 주의 네트워크인 *HGAT*를 설계합니다. *HGAT*는 여러 신경망 모듈을 통해 사용자 이력 행동, 레시피 콘텐츠, 관계 정보를 수집할 수 있습니다. 또한 점수 예측자와 순위 기반 목적 함수를 도입하여 모델을 최적화합니다. 광범위한 실험을 통해 *HGAT*가 다양한 기준 접근법보다 성능이 우수하다는 것이 입증되었습니다. 앞으로 더 많은 정보를 통합하고 *HGAT*를 개선할 계획입니다. 사용자 리뷰와 레시피의 건강 상태 요인 등 사용할 수 있는 유용한 정보 구성 요소가 여전히 많다는 것을 확인했습니다. 한 가지 유망한 방향은 사용자 선호도와 건강에 대한 우려에 맞는 레시피를 추천하는 방법을 연구하는 것입니다.

데이터 가용성 설명

이 글의 결론을 뒷받침하는 원시 데이터는 저자가 과도한 유보 없이 제공할 것입니다.

작성자 기여

YT, CZ, RM, NC가 연구의 전반적인 설계에 기여했습니다. YT가 실험을 수행했습니다. CZ는 결과 해석을 수행했습니다. YT가 원고의 초안을 작성했습니다. 모든 저자가 원고 수정에 기여하고 제출된 버전을 승인했습니다.

펀딩

이 연구는 미국 농무부 국립식량농업연구소의 농업 및 식품 연구 이니셔티브 보조금 번호 2021-67022-33447/프로젝트 가입 번호 1024822의 지원으로 수행되었습니다.

참고 자료

- Abel, F., Gao, Q., Houben, G.-J., Tao, K. (2011). "개인화된 뉴스 추천을 위한 트위터 사용자 모델링 분석", *UMAP* (지로나).
- Adaji, I., Sharmaine, C., Debrowney, S., Oyibo, K., Vassileva, J. (2018). "레시피 네트워크 그래프를 사용한 성격 기반 레시피 추천", *SCSM*(라스베이거스, 네바다).
- Bayer, I., He, X., Kanagal, B., Rendle, S. (2017). "암시적 피드백으로부터 학습하기 위한 일반 좌표 하강 프레임워크", *WWW*(퍼스, WA).
- Bouma, G. (2009). "배열 추출에서 정규화 된 (포인트 별) 상호 정보", *GSCL* (포츠담).
- 브리토, L., 파시피코, L., 올리베이라, E., 루더미르, T. (2020). "식품 제한 식별을 위한 요리 레시피 다중 라벨 분류 접근법", *ENIAC*(리오그란데).
- Chen, M., Jia, X., Gorbonos, E., Hoang, C. T., Yu, X., Liu, Y. (2020). 더 건강한 식생활: 더 건강한 레시피 추천을 위한 영양 정보 탐색. *Inf. Process. Manag.* 57, 102051.
- Dong, Y., Chawla, N. V., & Swami, A. (2017). "메타패스2백: 이기종 네트워크를 위한 확장 가능한 표현 학습", *KDD* (헬리팩스, NS).
- Elkahky, A. M., Song, Y., He, X. (2015). "추천 시스템에서 크로스 도메인 사용자 모델링을 위한 멀티뷰 딥 러닝 접근 방식", *WWW* (피렌체).
- Forbes, P., & Zhu, M. (2011). "추천 시스템을 위한 콘텐츠 부스트 행렬 인수분해: 레시피 추천 실험", *RecSys*(일리노이주 시카고).
- Freyne, J., & Berkovsky, S. (2010). "지능형 음식 계획 : 개인화 된 레시피 추천", *IUI* (일리노이 주 시카고).
- Gao, X., Feng, F., He, X., Huang, H., Guan, X., Feng, C., et al. (2020). 시각적 인식 식품 추천을 위한 계층적 주의 네트워크. *IEEE Trans. 멀티미디어* 22, 1647-1659. doi: 10.1109/TMM.2019.2945180
- Ge, M., Elahi, M., FernandezTobias, I., Ricci, F., Massimo, D. (2015). "음식 추천 시스템에서 태그 및 잠재 요인 사용", *ICDH* (피렌체).
- 글로롯, X., 및 벤지오, Y. (2010). "심층 피드포워드 신경망 훈련의 어려움에 대한 이해", *AISTATS*(사르데냐).
- 해밀턴, W. L., 잉, R., 및 레스코백, J. (2017). "큰 그래프에 대한 귀납적 표현 학습", *NeurIPS* (캘리포니아 롱비치).
- Hausmann, S., Seneviratne, O., Chen, Y., Ne'eman, Y., Codella, J., Chen, C.-H., 외. (2019). "Foodkg: 음식 추천을 위한 의미론 기반 지식 그래프", *ISWC* (오클랜드).
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S. (2017). "신경 협업 필터링", *WWW* (퍼스, WA).
- He, X., Zhang, H., Kan, M.-Y., Chua, T.-S. (2016). "암시적 피드백을 통한 온라인 추천을 위한 빠른 행렬 인수분해", *SIGIR*(피사).
- Hochreiter, S., & Schmidhuber, J. (1997). 긴 단기 기억. *Neural Comput.* 9, 1735-1780. doi: 10.1162/neco.1997.9.8.1735
- Jiang, H., Wang, W., Liu, M., Nie, L., Duan, L.-Y., Xu, C. (2019). "Market2dish: 건강을 고려한 음식 추천 시스템", *ACM-MM* (니스).
- Khan, M. A., Rushe, E., Smyth, B., 및 Coyle, D. (2019). 개인화 된 건강 인식 레시피 추천 : 양상불 주제 모델링 기반 접근 방식. *arXiv 사전 인쇄물* arXiv: 1908.00148.
- Kingma, D., 및 Ba, J. (2014). "Adam: 확률적 최적화를 위한 방법", 에서 *ICLR*(캘리포니아주 샌디에이고).
- Kipf, T. N., 및 Welling, M. (2017). "그래프 컨볼 루션 네트워크를 사용한 반지도 분류", *ICLR* (틀루).
- Li, D., & Zaki, M. J. (2020). "수신자: 레시피 표현 학습을 위한 효과적인 사전 훈련 모델", *KDD*(가상 이벤트, 캘리포니아).
- Li, Q., Chen, W., Yu, L. (2010). "P2P 네트워크에서 커뮤니티 기반 레시피 추천 및 적응", *ICUIMC* (수원).
- Marin, J., Biswas, A., Ofli, F., Hynes, N., Salvador, A., Aytar, Y., et al. (2019). Recipe1m+: 요리 레시피 및 음식 이미지에 대한 교차 모드 임베딩 학습을 위한 데이터 세트. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 187-203. doi: 10.1109/TPAMI.2019.2927476
- Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). "벡터 공간에서 단어 표현의 효율적인 추정", *ICLR*(스코츠데일).
- Min, W., Jiang, S., Jain, R. (2020). 음식 추천: 프레임워크, 기존 솔루션 및 과제. *IEEE Trans. 멀티미디어* 22, 2659-2671. doi: 10.1109/TMM.2019.2958761

Min, W., Jiang, S., Liu, L., Rui, Y., Jain, R. (2019). 식품 컴퓨팅에 대한 설문 조사.

arXiv 사전 인쇄 arXiv:1808.07202.

Mouritsen, O. G., Edwards-Stuart, R., Ahn, Y.-Y., 및 Ahnert, S. E. (2017). 음식 인식, 준비, 소비 및 문화 연구를 위한 데이터 기반 방법. *Front. ICT* 4, 15. doi: 10.3389/fict.2017.00015

Musto, C., Trattner, C., Starke, A., 및 Semeraro, G. (2020). "전체론적 사용자 모델을 활용하는 지식 인식 식품 추천 시스템을 향하여", *UMAP* (제노바).

Park, D., Kim, K., Kim, S., and Spranger, M. (2021). Flavorgraph: 음식 표현을 생성하고 음식 페어링을 추천하기 위한 대규모 식품-화학 그래프. 11, 931. doi: 10.1038/s41598-020-79422-8

Park, D., Kim, K., Park, Y., Shin, J., and Kang, J. (2019). "Kitchenette: 삼 신경망을 이용한 식재료 페어링 예측 및 순위 지정", *IJCAI* (마카오).

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). "Deepwalk: 사회적 표현의 온라인 학습", *KDD* (뉴욕, NY).

Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. (2009). "Bpr: 암시적 피드백을 통한 베이저안 개인화 순위", *UAI* (몬트리올, QC). Rendle, S.,

Krichene, W., Zhang, L., Anderson, J. R. (2020). 신경 협업 필터링 대 행렬 인수분해 재검토. *arXiv 사전 인쇄물* arXiv: 2005.09683.

살바도르, A., 하인즈, N., 아이타르, Y., 마린, J., 오폴리, F., 웨버, I., 토랄바, A. (2017). "요리 레시피 및 음식 이미지에 대한 교차 모드 임베딩 학습", *CVPR* (호놀룰루, HI).

Sano, N., Machino, N., Yada, K., & Suzuki, T. (2015). 데이터 희소성을 고려한 식료품점 추천 시스템. *Procedia Comput.* 60, 1406-1413. doi: 10.1016/j.procs.2015.08.216

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., 및 Welling, M. (2018). "그래프 컨볼 루션 네트워크를 사용한 관계형 데이터 모델링", *에서 ESWC* (헤라클리온).

Teng, C., Lin, Y., Adamic, L. A. (2012). "재료 네트워크를 사용한 레시피 추천", *WebSci* (일리노이주 에반스턴).

Tian, Y., Zhang, C., Metoyer, R. A., 및 Chawla, N. (2021). "네트워크를 이용한 레시피 표현 학습", *제30회 ACM 정보 및 지식 관리 국제 컨퍼런스 논문집* 1824-1833. doi: 10.1145/3459637.3482468

Trattner, C., and Elsweiler, D. (2017a). 음식 추천 시스템 : 중요한 기여, 과제 및 향후 연구 방향. *arXiv 사전 인쇄물* arXiv: 1711.02760.

Trattner, C., 및 Elsweiler, D. (2017b). "식사 계획 및 추천 시스템에 대한 인터넷 소스 레시피의 건강성 조사", *WWW* (퍼스, WA).

Trattner, C., & Elsweiler, D. (2019). "온라인 레시피 포털을 위한 추천 알고리즘 평가", *RecSys* (코펜하겐).

미국 농무부, A. R. S. (2019). *표준 참조를 위한 미국 농무부 국가 영양소 데이터 베이스, 릴리스 27. 식품 조성 실험실 홈페이지의 방법 및 적용*, 온라인에서 사용 가능: <http://www.ars.usda.gov/nea/bhnrc/mafcl>.

우에다, M., 타카하타, M., 나카지마, S. (2011). "개인화된 요리 레시피 추천을 위한 사용자의 음식 선호도 추출", *ISWC* (본).

van der Maaten, L., & Hinton, G. (2008). t-sne를 사용한 데이터 시각화. *J. Mach. Learn. Res.* 9, 2579-2605. 온라인에서 사용 가능: <http://jmlr.org/papers/v9/vandermaaten08a.html>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., 외.

(2017). "주의만 있으면 됩니다.", *NeurIPS* (캘리포니아 롱비치). ^{HGAT를 통한 레시피 추천}

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y.

(2018). "그래프 주의 네트워크", *ICLR* (BC 주 밴쿠버).

비백, M., 만주, N., 비제이, M. (2018). "머신러닝 기반 음식 레시피 추천 시스템", *ICCR*.

우, Z., 판, S., 첸, F., 롱, G., 장, C., 유, P. S. (2021). 그래프 신경망에 대한 포괄적 인 설문 조사. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 4-24. doi: 10.1109/TNNLS.2020.2978386

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-I., Jegelka, S. (2018). "점핑 지식 네트워크가 있는 그래프에 대한 표현 학습", *ICML* (스톡홀름).

양, L., 시에, A., 양, H., 폴락, J., 델, N., 벨롱기, S., 쿨, C., 에스트린, D. (2017). Yum-me: 개인화된 영양소 기반 식사 추천 시스템.

ACM Trans. Inf. Syst. 36, 1-31. doi: 10.1145/3072614

Zhang, F., Yuan, N. J., Zheng, K., Lian, D., Xie, X., Rui, Y. (2016). "레스토랑 추천을 위한 식사 선호도 활용", *WWW* (몬트리올, QC).

Zhou, C., Bai, J., Song, J., Liu, X., Zhao, Z., Chen, X., et al. (2018). "Atrank: 추천을 위한 주의 기반 사용자 행동 모델링 프레임워크", *AAAI*(뉴올리언스, LA).

이해 상충: 저자는 이 연구가 잠재적 이해 상충으로 해석될 수 있는 상업적 또는 재정적 관계가 없는 상태에서 수행되었음을 선언합니다.

발행인 참고: 이 글에 표현된 모든 주장은 전적으로 저자의 주장일 뿐이며, 저자의 소속 기관이나 발행인, 편집자 및 검토자의 주장을 반드시 대변하는 것은 아닙니다. 이 문서에서 평가할 수 있는 제품 또는 제조업체가 주장할 수 있는 모든 제품은 게시자가 보증하거나 보증하지 않습니다.

저작권 © 2022 Tian, Zhang, Metoyer 및 Chawla. 이 문서는 크리에이티브 커먼즈 저작자표시 라이선스(CC BY)의 조건에 따라 배포되는 오픈 액세스 문서입니다. 다른 포럼에서의 사용, 배포 또는 복제는 허용되는 학술 관행에 따라 원저자와 저작권 소유자의 출처를 밝히고 본 저널의 원본 출판물을 인용하는 경우에 한하여 허용됩니다. 본 약관을 준수하지 않는 사용, 배포 또는 복제는 허용되지 않습니다.