Journal of the Korea Convergence Society
Vol. 12. No. 7, pp. 53-60, 2021

ISSN 2233-4890 / eISSN 2713-6353
https://doi.org/10.15207/JKCS.2021.12.7.053

LOD-기반 추천 시스템에서 LOD 그래프에 가중치를 사용한 의미 거리 측정 모델

허원회
성결대학교 미디어소프트웨어학부 교수

A Semantic Distance Measurement Model using Weights on the LOD Graph in an LOD-based Recommender System

Huh, Wonwhoi
Professor, Division of Media Software, Sungkyul University

요 약 LOD-기반 추천 시스템은 보통 DBpedia와 같은 LOD 데이터세트 내에서 사용가능한 데이터를 활용하여 최종 사용자에게 영화, 책, 음악과 같은 아이템을 추천한다. 이러한 시스템은 링크드 데이터 리소스 쌍 간의 일치 정도를 측정하는 의미 유사도 알고리즘을 사용한다. 이 논문에서는 LOD 그래프의 링크에 사용자 평가 등급을 변환한 가중치를 할당하여 LOD-기반 추천 시스템에서 의미 거리를 측정하는 새로운 접근방식을 제안했다. 이 논문에서 제안된 의미 거리 측정 모델은 가중치 계산을 통해 그래프가 사용자에게 개인화되는 처리 단계와 이러한 가중치를 LDSD에 적용하는 방법을 기반으로 한다. 실험 결과는 다른 유사한 방법들과 비교하여 제안된 방법이 더 높은 정확도를 보였으며, 추천 시스템의 의미 거리 측정의 범위를 넓혀서 유사도 향상에 기여하였다. 향후 연구로는 다른 방법의 LOD-기반 유사도 측정을 사용하여 모델에 미치는 영향을 분석하는 것을 목표로 한다.

주제어 : 의미 거리, 링크드 오픈 데이터, 추천, RDF, 유사도

Abstract LOD-based recommender systems usually leverage the data available within LOD datasets, such as DBpedia, in order to recommend items(movies, books, music) to the end users. These systems use a semantic similarity algorithm that calculates the degree of matching between pairs of Linked Data resources. In this paper, we proposed a new approach to measuring semantic distance in an LOD-based recommender system by assigning weights converted from user ratings to links in the LOD graph. The semantic distance measurement model proposed in this paper is based on a processing step in which a graph is personalized to a user through weight calculation and a method of applying these weights to LDSD. The Experimental results showed that the proposed method showed higher accuracy compared to other similar methods, and it contributed to the improvement of similarity by expanding the range of semantic distance measurement of the recommender system. As future work, we aim to analyze the impact on the model using different methods of LOD-based similarity measurement.

Key Words : Semantic, Linked Open Data, Recommender, RDF, Similarity

*Corresponding Author : Huh, Wonwhoi(wonwhoi@sungkyul.ac.kr)
Received April 6, 2021 Revised April 26, 2021
Accepted July 20, 2021 Published July 28, 2021

1. 서론

추천 시스템(Recommender System)은 사용자가 사용할 아이템에 대한 만족할만한 제안을 제공하는 소프트웨어 도구 및 기술이다[1]. 이러한 제안은 사용자가 소셜 네트워크에서 연결할 대상과 구매할 제품들을 음악 또는 볼 영화와 같은 다양한 의사 결정 프로세스와 관련된다. 제품, 음악, 영화는 모두 특정 추천 시나리오에 있는 아이템의 예이다. 오늘날 거의 모든 온라인 서비스에는 추천 기능이 있으며, Ringo, GroupLens, Youtube, Amazon, Netflix 등은 시스템에서 추천 기능을 사용하여 사용자를 참여시키고 더 나은 서비스를 제공한다.

LOD(Linked Open Data)는 구조화된 데이터를 사용자들에게 공개적으로 소비하고 배포할 수 있는 새로운 표준과 형식인 구조화된 공유 가능 데이터이다. LOD중 일부는 음악과 같은 특정 지식 영역에 특화된 반면 대부분은 DBpedia와 같은 여러 영역 간에 개념을 포함하는 일반적인것이다[2]. LOD는 다양한 영역에서 구조화된 데이터를 광범위하게 제공하기 때문에 추천 시스템 분야에서 많이 연구되었다. 특히 LOD는 다중-도메인 개념을 포함하는 광범위한 개방형 데이터세트를 서로의 관계와 함께 제공하며, 이러한 관계를 통해 추천 시스템은 컬렉션 전체에서 관련 개념을 식별 할 수 있다[3]. 또한 LOD 표준 및 기술은 필요한 데이터를 검색하기 위한 표준 인터페이스를 제공하여 추천 시스템의 작업을 용이하게 하므로 데이터에 대한 온톨로지 지식을 제공 할뿐만 아니라 원시 데이터의 추가 컴퓨팅 처리가 필요하지 않다.

LOD-기반 추천 시스템은 일반적으로 DBpedia와 같은 LOD 데이터세트내에서 사용가능한 데이터를 활용하여 최종 사용자에게 아이템(예: 영화, 책, 음악)을 추천한다[4,5]. 이러한 시스템은 링크드 데이터 리소스 쌍 간의 일치 정도를 계산하는 의미 유사도 알고리즘을 사용한다. RDF는 데이터를 그래프로 나타내기 때문에 일반적으로 이러한 알고리즘은 직접 및 간접 링크의 수, 두 리소스 간의 경로 길이 또는 클래스 계층 구조에서의 위치를 계산한다[4].

LOD의 사용은 주로 그래프 특성 표현을 활용하거나 통계적 접근을 통해 다양한 방법으로 추천 시스템에서 탐색되었다[6]. 추천 시스템은 LOD의 내용을 활용하여 사용자 데이터가 거의 없는 상황에서도 추천할 관련 자원을 식별한다. LOD-기반 그래프 구조에서의 접근방식은 그래프에서 의미 거리로 리소스 관련성을 측정한다. 이 접근방식의 직관은 LOD 그래프에서 서로 연결된 리소스가 많을수록 관련성이 높다는 것이다. 이 개념은 리소스 관련성 측정의 핵심인 링크드 데이터 의미 거리(LDSD)[4]와 이를 기반으로 한 방식인 리소스 유사도(Resim)[7]와 PLDSD[3]이다. LDSD 방식은 한개의 중간 리소스를 통해 직접적이거나 간접적으로 연결되는 리소스 간의 의미 거리만 계산하며, Resim 방식은 링크가 두 개 이상 떨어져있는 리소스에 대해서도 유사도 측정을 한다. 그러나 PLDSD방식은 LDSD 의미 거리를 넓히는 전파된 방식으로 Resim 보다 더 많은 거리의 리소스에 대한 그래프 구조도 고려하여 측정한다. 또한 이 방식들은 모든 링크를 동일한 가중치를 부여하여 의미 거리를 측정하였기 때문에 링크별 중요도를 간과하였다.

이 연구는 LOD 그래프의 링크에 사용자가 부여한 평가 등급을 변환한 가중치를 할당하여 LOD-기반 추천 시스템에서 의미 거리를 측정하는 새로운 접근방식을 제안했다. 이 접근법은 사용자가 이전에 선택한 선호도(평가 등급)를 고려하여 최상의 특징에 순위를 매김으로써 희소성 문제를 최소화하는 것을 목표로 한다. 또한 특징 순위 작업은 이전 아이템과 새 아이템 간에 공유되는 특징이 사용자의 이전 선호도를 기반으로 하기 때문에 새 아이템이 시스템에 추가될 때 콜드-스타트 문제를 해결한다. 그리고 이 논문에서는 가중치 계산을 통해 그래프가 사용자에게 개인화되는 전처리 단계와 이러한 가중치를 추천 시스템 알고리즘의 핵심인 LDSD에 적용하는 방법을 제안한다.

2. 관련 연구

2.1 RDF 구조

RDF 문은 트리플이라고도 하며 〈subject, predicate, object〉 형식으로 나타낸다. Fig. 1에서와 같이 subject, predicate, object는 URI를 사용하여 고유하게 식별한다. subject는 문이 참조하는 리소스이고, predicate는 리소스가 가지는 특성을 표시하며, object는 속성 값에 해당되는 리소스 또는 리터럴이다. predicate는 두 유형(object 속성과 데이터타입 속성) 중에 하나인데, object 속성은 속성이 하나의 리소스를 다른 리소스에 연결하는 경우이고, 데이터타입 속성은 속성이 하나의 리소스를 리터럴(문자열이나 숫자)에 연결하는 경우이다. 또한 subject가 여러 object에 연결되어 다양한 문을 표현할 수 있고 각각의 object가 다른 문의 subject가 될 수가 있기 때문에 RDF 그래프 표현은 상호 연결된 노드의 방

대한 데이터세트를 생성한다.

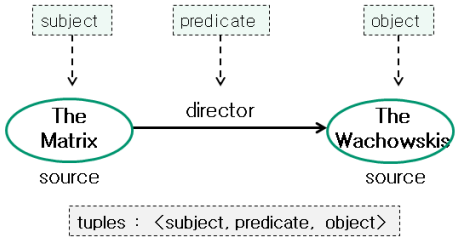


Fig. 1. RDF Structure

2.2 의미 거리 접근법 비교 분석

LOD 리소스간의 의미 거리를 측정하기 위하여 다양한 방법들이 제안되었다[2]. Musto(2016)의 연구에서는 LOD에서 얻은 지식이 그래프-기반 추천 알고리즘의 전체 성능에 미치는 영향을 연구했다[8]. Noia(2018)의 연구에서는 LOD-기반 요약이 온톨로지에서 제공하는 의미를 기반으로 완전히 자동화된 방법을 제안하여 어떻게 특징 선택 작업을 하는지를 보여주었다[9]. Passant(2010)의 연구(LDSD)에서는 LOD에 있는 리소스 사이의 의미 거리를 추정하여 추천 시스템에서 LOD를 이용하는 방법을 제안했다[4]. 이 접근 방식은 중간 리소스를 통한 간접 리소스뿐만 아니라 리소스 간의 직접 링크를 사용하여 이들 리소스 사이의 의미 거리를 계산한다. Passant(2010)의 연구에서는 LDSD 접근 방식을 사용하여 음악가와 밴드를 추천하기 위해 DBpedia 위에 구축된 dbrec라는 음악 추천 시스템을 만들었다[10].

Piao(2015)의 연구(Resim)에서는 최소값과 대칭성의 약점을 극복하면서 원래 LDSD 접근 방식을 개선한 리소스 유사도라는 개선된 링크드 데이터 의미 거리 접근방식을 제안했다[7]. 또한 두개 이상의 링크로 떨어진 노드 간에 대하여 속성-기반 유사도 측정을 사용하여 의미 거리에 참여하는 노드 수를 확장하였다. Alfarhood(2017)의 연구에서는 LDSD의 의미 거리 접근법의 범위를 넓히는 전파된 링크 데이터 의미 거리(PLDSD)라 불리는 접근법을 제안하였다[3]. PLDSD에서는 잘 알려진 모든 쌍 최단 경로 알고리즘을 사용하여 하나 또는 두 개의 링크 거리에 있는 리소스를 확장하여 의미 거리를 계산하였다.

앞에서 논의된 여러 연구들의 추천 알고리즘에서의 의미 거리 접근법에서는 각각의 문제점을 가지고 있다. 여러 연구들 중에서 이 논문의 방법과 비교 평가하는 방법들 중에 LDSD에서는 링크가 두 개 이상 떨어져 있는 리소스는 서로 관련 없는 것으로 간주되어 계산되었다. 또한 Resim에서는 기본적으로 해당 노드의 특성만 반영하였으며, 링크가 더 멀리 떨어진 리소스에 대한 그래프 구조를 고려하지 않고 리소스간의 유사도를 계산하였다. 그리고 PLDSD에서는 LOD-기반 그래프 축소와 의미 거리 전파를 통해 유사도를 측정하였으나 리소스에 연결된 링크들에 대해서는 고정된 값으로 리소스 간의 유사도를 계산하였다. 또한 이 방식들은 링크별 중요도를 간과함으로써 모든 링크를 동일한 가중치를 부여하여 의미 거리를 측정하였다. 따라서 이 논문에서는 LOD-기반 그래프에서 링크에 사용자가 부여한 평가 등급을 변환한 가중치를 할당하여 의미 거리를 측정하는 새로운 접근방식을 제안한다.

3. 제안된 LOD-기반 추천 시스템

3.1 개념 정의

3.1.1 링크드 데이터 그래프 정의

링크드 데이터는 정보 리소스 또는 리소스로 알려진 다양한 요소들에 관련된 RDF 문장들의 방대한 집합이다. 각 RDF 문(트리플이라고 함)은 subject와 object가 노드인 노드 및 링크로 표현되며, 이들 간의 관계(predicate)는 노드를 연결하는 링크이다. 링크는 연결되어 있으며 이는 링크의 방향이 관계 정의의 일부임을 의미한다. 따라서 링크드 데이터를 설명하는 RDF 명령문의 그래프 표시는 상호 연결된 노드의 대규모 그래프를 형성한다. 이를 통해 링크드 데이터를 리소스 및 이들 간의 관계에 대한 그래프로 정의한다.

LOD 그래프는 $G=\{R, P, T\}$로 정의된 방향 그래프이다. $R=\{r_1, r_2, ..., r_n\}$은 리소스 집합이고, $P=\{p_1, p_2, ..., p_n\}$는 속성(링크) 집합이며, $T=\{t_1, t_2, ..., t_n\}$는 리소스 쌍을 연결하는 $<r_1, p_1, r_2>$ 와 같은 트리플 집합이다. 따라서 형식 〈subject, predicate, object〉의 명령문은 RDF 트리플로 표현됨으로, $t_i = <r_a, p_j, r_b> \in T$ 는 subject $r_a \in R$에서 object $r_b \in R$로 연결하는 predicate $p_j \in P$ 의 인스턴스가 있음을 의미하며, Fig.

$$LDSD(r_a, r_b) = 1/(1+\sum_i \frac{C_d(l_i, r_a, r_b)}{1+\log(C_d(l_i, r_a))} + \sum_i \frac{C_d(l_i, r_b, r_a)}{1+\log(C_d(l_i, r_b))} + \sum_i \frac{C_{ii}(l_i, r_a, r_b)}{1+\log(C_{ii}(l_i, r_a))} + \sum_i \frac{C_{io}(l_i, r_a, r_b)}{1+\log(C_{io}(l_i, r_a))}) \quad (1)$$

1을 표현하면 트리플은 〈The Matrix, director, The Wachowskis〉 이다.

3.1.2 사용자 모델

사용자 모델은 추천 시스템에서 아이템에 대한 사용자의 선호도를 나타내는 방식이다. 이 연구에서 사용자 모델은 링크드 데이터 그래프의 일부로 모델링한다. 사용자 집합은 $U=\{u_1,u_2,...,u_n\}$로 가정하는데, 각각의 u_k는 LOD 그래프의 리소스이며, $U\subset R$ 이다.

Fig. 3을 참조하면 $P'\subset P$로 집합 $P'=\{p_1',p_2',p_3',p_4',p_5'\}=\{rating1,rating2,rating3,rating4,rating5\}$로 나타낼 수 있는데, 이는 사용자가 1에서 5까지의 등급으로 아이템을 얼마나 좋아하는지를 나타내는 속성이라고 가정한다. 또한 집합 T'는 형식 $t_i'=<u_k,p_l',r_a>\in T'$의 트리플로 구성된다. 예를 들어, 트리플 $t_1'=<u_1,rating5,r_1>$은 사용자 $u_1\in U$가 $r_1\in R$ 아이템을 5로 평가했음을 의미한다. 따라서 모델링의 총 도달 범위를 $G\cup G'$, 즉 그래프 $G=(R,P,T)$와 사용자 모델 그래프 $G'=(R,P',T')$ 사이의 합집합으로 정의한다.

3.1.3 LDSD

링크드 데이터 의미 거리(LDSD)는 LOD에서 두 리소스 사이의 관련성을 측정하는 접근방법이다[4]. 이 방식은 식 1과 같이 계산된다.

의미 거리 측정은 리소스 r_a에서 리소스 r_b로의 직접 링크와 그것의 반대 링크를 고려한다. 또한 리소스 r_a 및 r_b의 동일한 속성을 통해 동일하게 인커밍 노드 및 아웃고잉 노드를 고려한다. 의미 거리 측정의 범위는 0에서 1까지이며, 값이 클수록 두 리소스 간의 유사도가 적다. 따라서 리소스가 두 개 이상 떨어져있는 리소스는 서로 관련되지 않은 것으로 간주된다.

3.2 개인화된 그래프를 반영한 의미 거리

이 절은 DBpedia에서 개인화된 추천 모델을 도출하는 세부 사항을 기술한다. 이 논문에서 제안된 방법에 대한 아키텍처는 Fig. 2와 같다. 제안된 접근방법은 개인화된 그래프 생성, 평가 등급 변환, 빈도수 계산과 정규화된 평균 가중치 값 산출, 개인화된 추천 모델 생성 단계를 거친다.

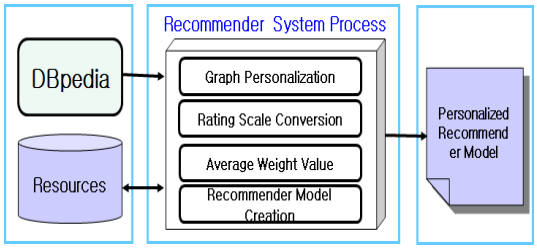


Fig. 2. The system architecture of the proposed method

3.2.1 개인화된 그래프 생성

이 논문에서는 사용자 모델을 입력으로 사용하여 평가 등급 특징을 동적으로 변환하는 RDF 그래프를 개인화한다. 이러한 방법은 지정된 추천 시스템에서 사용자가 평가한 이전의 선호도(1~5까지의 스타 리커드 척도)를 분석하고 변환하여 그래프의 링크에 할당한다. Fig. 3은 $G'\subset G$ 데이터세트인 영화 도메인을 설명하는 DBpedia 데이터세트의 일부이며, 그래프에서 RDF 타입인 rdf:type 형식은 접두어를 생략하였다. 사용자는 user를 의미하고, 영화 콘텐츠 리소스는 Toy Story, Avengers, Spider Man 등으로 표시되고, 사용자에 의해 이전에 평가된 영화만이 데이터세트 G'의 요소이다. 또한 영화 콘텐츠 리소스에 링크된 다른 콘텐츠 리소스인 director, product 등을 도시한다. 모델의 각 화살표는 트리플인 〈subject, predicate, object〉 형식으로 구성되고, 두 개의 리소스를 연결하는 속성 또는 predicate를 의미한다.

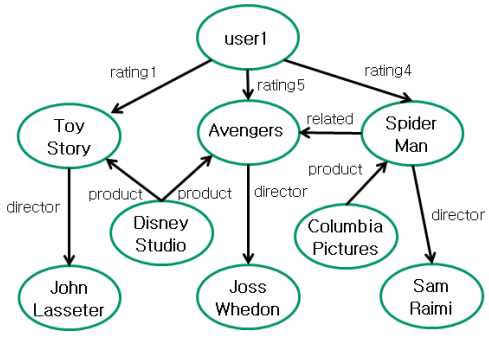


Fig. 3. A portion of dataset describing a movie domain

3.2.2 평가 등급 변환

추천 시스템에서의 평가는 사용자가 아이템을 좋아하거나 싫어하는 정도에 따라 제공하는 긍정적이거나 부정적인 명시적 피드백이다[11]. 평가 등급은 1~5까지의 스

타 리커드 척도를 사용한다. 그러나 스타 리커드 척도는 낮은 피드백을 적용하기가 어려워 가중치 계산 단계에서 음의 값을 할당하여 평가 등급을 정수 평가로 변환한다. 평가 5는 사용자가 아이템을 정말 좋아한다는 것을 의미하므로 등급을 3으로 변환한다. 평가 4는 사용자가 아이템을 좋아한다는 것을 의미하므로 등급을 2로 변환하며, 평가 3은 사용자가 아이템을 좋아하지도 싫어하지도 않는다는 것을 의미하므로 등급을 1로 변환한다. 마찬가지로 평가 2는 사용자가 아이템을 싫어한다는 것을 의미하므로 등급을 -1로 변환하며, 평가 1은 사용자가 아이템을 정말로 싫어한다는 것을 의미하므로 등급을 -2로 변환한다. 따라서 최저 등급(평가 1, 2)은 부정적인 사용자 피드백으로 간주한다. Table 1은 앞에서 설명한 평가 등급 변환을 요약하였다.

Table 1. The Conversion of Rating Scale

Rating scale	Property name	Conversion value
5	Rating5	3
4	Rating4	2
3	Rating3	1
2	Rating2	-1
1	Rating1	-2

Table 2는 표기법 $t_i^{'} =< u_k, p_l^{'}, r_a > \in T^{'}$ 를 사용하여 $G^{'}$의 모든 사용자 트리플을 표로 만들었다. 따라서 Table 2에는 사용자 u_1의 사용자 모델이 요약되어 있다. 또한 사용자가 평가한 평가 등급을 변환한 등급($R_{u_k r_a}$)를 마지막 열에 추가하였다. 이 등급(평가 변환 등급) $R_{u_k r_a}$는 u_k에 의해 주어진 r_a에 대한 평가 등급 값을 Table 1을 사용하여 정수 값인 등급으로 변환된 속성 $p_j^{'}$와 같다. 예를 들어, 트리플 $t_1^{'} =< u_1, p_1^{'}, r_1 >=$ 〈user1, rating1, Toy Story〉의 등급인 $R_{u_k r_a}$ 값은 -2이다. 나머지 트리플 $t_2^{'}$, $t_3^{'}$의 등급 변환도 트리플 $t_1^{'}$와 동일하다.

Table 2. Collection of user triples from graph $G^{'}$

triple ($t_i^{'}$)	subject (u_k)	predicate ($p_l^{'}$)	object (r_a)	$R_{u_k r_a}$
$t_1^{'}$	u1-user1	$p_1^{'}$-rating1	r1-Toy Story	-2
$t_2^{'}$	u1-user1	$p_5^{'}$-rating5	r2-Avengers	3
$t_3^{'}$	u1-user1	$p_4^{'}$-rating4	r3-Spider Man	2

3.2.3 정규화된 평균 가중치 도출

그래프 $G^{'}$의 모든 트리플은 표기법 $t_i =< r_a, p_j, r_b > \in T$를 사용하여 Table 3에 표시하였다. 따라서 Table 3은 그래프 $G^{'}$에 있는 모든 트리플 t_i의 집합을 나타내고, 이는 트리플 $t_1 =< r_1, p_1, r_4 >=$〈Toy Story, director, John Lasseter〉과 같이 u_1에 대한 사용자 모델 집합으로부터 영화(Toy Story, Avengers, Spider Man)에 링크된 모든 트리플이다. 이 논문에서는 테이블을 단순화하기 위하여 영화가 object인 트리플을 반전시키고 항상 문장의 subject로 만든다. Table 3에서는 t_4,t_5,t_6가 역 트리플로 표현되었다. 예를 들어 t_4는 $< r_1, p_2, r_5 >$로 표시된다.

Fig. 3과 Table 2, Table 3을 사용하여 한사람의 사용자 $u_k \in U$ 에 따라 그래프 $G^{'}$를 변환된 평가 등급을 반영하고 빈도수를 계산하여 평균 가중치를 다음과 같이 도출한다.

빈도수 계산($F(r_b)$): 그래프 $G^{'}$에서 트리플 $t_i =< r_a, p_j, r_b > \in T$의 빈도수 $F(r_b)$를 계산한다. i는 1에서 n까지 연속적으로 변하며, 반복하는 동안에 Table 3의 object(r_b) 열에서 각 리소스가 몇 번 나타나는지를 계산한다. 즉 t_i 트리플에서 object r_b의 리소스 중에서 같은 리소스가 몇 번이나 나타나는지를 계산한다.

평균 가중치 계산($Wt(p_j,u_k)$): i가 1에서 n까지 변하는 트리플 $t_i =< r_a, p_j, r_b > \in T$ 에 대해 반복된 합산을 수행한다. 식 2와 같이 각각의 반복에서 r_a와 관련된 변환된 등급 $R_{u_k r_a}$ 값을 찾아서 object 빈도수 $F(r_b)$에 곱한다. $t_i =< r_a, p_j, r_b > \in T$마다 이 프로세스가 반복된 후 각 속성 p_j의 평균값을 계산한다. 동일한 속성 p_j가 그래프에서 둘 이상의 객체 r_b에 연결된 것으로 나타나면 값을 합산하여 평균을 계산한다. 또한 그래프에서 둘 이상의 영화 r_a에 대해 동일한 속성 p_j가 나타나면 값이 합산된다. 그리고 사용자 u_k가 주어지면 각 속성 p_j에 대한 정규화된 평균 가중치인 $Wt(p_j,u_k)$를 찾아 값을 정규화한다.

사용자 u_1에 대한 Fig. 3의 $Wt(p_j,u_k)$ 값은 Table 3과 같다. 식 2에서 $t_i =< r_a, p_j, r_b > \in T$ 이다.

$$Wt(p_j,u_k) = 1/(1+\frac{\sum_{t_i} F(r_b).Rat_{u_k r_a}}{n}) \qquad (2)$$

$$WLDSD(r_a, r_b, u_k) = 1/(1+\sum_i \frac{C_d(p_i, r_a, r_b)\, Wt(p_j, u_k)}{1+\log(C_d(p_i, r_a, n))} + \sum_i \frac{C_d(p_i, r_b, r_a)\, Wt(p_j, u_k)}{1+\log(C_d(p_i, r_b, n))} + \sum_i \frac{C_{ii}(p_i, r_a, r_b)\, Wt(p_j, u_k)}{1+\log(C_{ii}(p_i, r_a, n))} + \sum_i \frac{C_{io}(p_i, r_a, r_b)\, Wt(p_j, u_k)}{1+\log(C_{io}(p_i, r_a, n))}) \quad (3)$$

3.2.4 개인화된 추천 모델 생성

이 논문에서의 제안된 추천 모델은 앞서 설명한 개인화 방법의 평균 가중치($Wt(p_j,u_k)$)와 LOD-기반 의미 거리(LDSD)를 결합한 유사성 측정을 정의한다. RDF 그래프에 가중치를 할당하면 지정된 콘텐츠-기반 추천 시스템에서 사용자의 선호도를 더 정확하게 나타낼 수가 있다. 따라서 이 논문의 접근방식이 도메인 지식과 사용자 개인화가 모두 중요한 역할을 하는 LOD-지원 콘텐츠-기반 추천 시스템에 적용될 수 있음을 의미한다.

기존의 LDSD[4] 유사성 측정에서는 모든 링크(p_j)가 같은 링크 가중치를 갖는 것으로 하여 계산된다. 그러나 이 논문에서는 추천 시스템을 한사람의 사용자 u_k로 개인화하기 위하여 평균 가중치 $Wt(p_j, u_k)$를 식 1에 추가하여 식 3을 얻는다. $Wt(p_j, u_k)$는 한사람의 사용자 u_k가 주어진 하나의 속성 p_j에 대한 가중치를 나타내므로, 그 값은 식 3에 정의되어있는 것과 같이 식 1에 있는 모든 $C(p_j, r_a, r_b)$ 함수에 곱해진다.

4. 실험 및 평가

4.1 데이터세트 설정

이 논문에서의 실험 평가 목표는 제안된 추천 모델이 LOD-기반 추천 시스템으로 각 사용자에 대하여 개인화할 때 성능 측면에서 평가한다.

실험 평가에 사용된 파일은 Movielens 1M 데이터세트이다[12,13]. 이 파일에는 2000년에 MovieLens에 가입한 6,040명의 MovieLens 사용자가 만든 약 3,900개의 영화에 대한 1,000,209개의 익명 평가 등급이 포함되어 있다. Movielens 1M 데이터세트에는 users, movies, ratings 파일이 있다. users 파일에는 userid, gender, age, occupation, zip-code 필드가 있으며, movies 파일에는 movieid, title, genres 필드가 있으며, ratings 파일에는 userid, movieid, rating, timestamp 필드가 있다. 이 파일을 바탕으로 사용자 모델(영화)에 대한 사용자와 영화 및 사용자 평가 등급을 저장할 수 있는 데이터베이스를 모델링하고 데이터를 저장했다. 사용자별로 속성의 평균 가중치 및 유사도 측정 결과는 파일에 동적으로 저장된다. 각 사용자 u_i와 리소스간의 유사도 점수는 식 4와 같이 계산되며, $SimDist(r_a, r_b)$는 이 논문을 포함하여 추천 시스템 평가에 사용된 의미 거리 측정치(LDSD, Resim, PLDSD, 이 논문)를 나타낸다. 또한 $Profile(u_i)$는 사용자 u_i가 이전에 즐겨 찾았던 모든 리소스를 포함하는 사용자 u_i의 사용자 프로필(사용자 모델)이다[2].

$$similarity(u_i, r_a) = \frac{\sum_{r_b \in Profile(u_i)} (1 - SimDist(r_a, r_b))}{|Profile(u_i)|} \quad (4)$$

Table 3. Collection of source triples from graph G'

triple (t_i)	subject (r_a)	predicate (p_j)	object (r_b)	$R_{u_k r_a}$	$F(r_b)$	$Wt(p_j,u_k)$
t_1	r1-Toy Story	p_1-director	r4-John Lasseter	-2	1	0.2
t_2	r2-Avengers	p_1-director	r6-Joss Whedon	3	1	0.2
t_3	r3-Spider Man	p_1-director	r8-Sam Raimi	2	1	0.2
t_4	r1-Toy Story	p_2-product	r5-Disney Studio	-2	2	0.26
t_5	r2-Avengers	p_2-product	r5-Disney Studio	3	2	0.26
t_6	r3-Spider Man	p_2-product	r7-Columbia Pictures	2	1	0.26
t_7	r3-Spider Man	p_3-related	r2-Avengers	2	1	0.4

4.2 실험 결과 및 평가

영화의 순위 목록과 미리 정의된 좋아하는 영화의 테스트 데이터세트를 가지고 생성된 순위의 품질을 평가하기 위해 두 가지 평가 지표가 적용되었다. 이러한 측정 척도는 유사도 알고리즘이 원하는 유사 아이템 집합을 얼마나 잘 예측할 수 있는지 고려한다. 성능의 비교 평가는 LDSD, Resim, PLDSD와 이 논문의 방법을 평가하였다. 테스트 데이터세트로 의미 거리 방법의 효과를 측정하기 위한 추천 시스템의 표준 평가 척도는 F_1-score와 MRR(Mean Reciprocal Rank)를 사용하였다. 정밀도와 재현율의 조화 평균인 F_1-score는 식 5와 같다.

$$F_1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (5)$$

MRR은 상호 순위의 평균값이며, 주어지는 질의에 대하여 그 질의와 정확히 일치하는 것이 몇 번째에 있는가는 측정한다. 즉 사용자와 관련된 첫 번째 항목의 평균 순위를 나타내며 식 6과 같이 계산된다. 여기서 $rank_i$는 질의 Q_i에서 관련 결과의 최고 순위이다[14,15].

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \quad (6)$$

Table 4. Results of the recommender system using different distance measures

	LDSD	Resim	PLDSD	this paper
MRR	0.029	0.039	0.045	0.051
F1@5	0.036	0.046	0.051	0.055
F1@10	0.045	0.056	0.061	0.064
F1@20	0.047	0.055	0.061	0.067

Table 4는 테스트 데이터세트를 F_1-score 및 MRR 척도를 사용하여 평가한 실험 결과이다. F_1-score 값은 상위 5개 결과(F_1@5), 상위 10개 결과(F_1@10), 상위 20개 결과(F_1@20)로 제시했으며, MRR(평균 역수 순위) 값은 사용자와 관련된 첫 번째 아이템의 평균 순위를 나타냈다. 이 논문의 평가 결과는 MRR, F_1@5, F_1@10, F_1@20 값이 각각 0.051, 0.055, 0.064, 0.067의 성능을 도출하여 실험한 다른 방법(LDSD, Resim, PLDSD)에 비하여 사용자 모델 성능이 우수함을 입증하였다.

MRR 값은 LDSD, Resim, PLDSD, 이 논문이 각각 0.029, 0.039, 0.045, 0.051의 평가 값이 도출되었으며, 이 논문이 LDSD에 비해 75.8%, Resim에 비해 30.7%, PLDSD에 비해 13.3%가 개선되었다. 또한 상위 5개 결과인 F_1@5에서의 F_1-score 값은 LDSD, Resim, PLDSD, 이 논문이 각각 0.036, 0.046, 0.051, 0.055의 결과가 도출되었으며, 이 논문이 LDSD에 비해 52.7%, Resim에 비해 19.5%, PLDSD에 비해 7.8%가 개선되었다. 그리고 상위 10개 결과인 F_1@10에서의 F_1-score 값은 LDSD, Resim, PLDSD, 이 논문이 각각 0.045, 0.056, 0.061, 0.064인 결과를 나타내었으며, 상위 20개 결과인 F_1@20에서의 F_1-score 값은 LDSD, Resim, PLDSD, 이 논문이 각각 0.047, 0.055, 0.061, 0.067의 평가 점수를 보여주었다. 이 논문에서의 개선된 결과는 실험한 결과 컷오프 지점(F_1@5, F_1@10, F_1@20)에서도 유지되었다.

이 논문의 실험에서는 LOD-기반 추천 시스템이 사용자가 과거에 평가한 아이템(예, 영화)의 특징을 고려하는 개인화된 특징 순위 지정방법을 활용할 수 있다는 제안을 검증했다. 이 논문에서는 사용자 모델 전략에서 통계적으로 의미심장한 개선을 얻었다. 즉 사용자에게 아이템을 추천하기 위해 더 중요한 특징의 순위를 매기는 기준으로 사용자의 과거 평가를 활용하는 것이 기존 방법(LDSD, Resim, PLDSD)과 비교하여 통계적으로 유의미한 것으로 나타났다.

5. 결론

이 논문에서는 링크드 데이터의 리소스에 대한 의미 유사성을 계산하기 위해 새로운 방법을 소개했다. 이 논문의 접근 방식에서는 과거의 사용자 선택에 따라 그래프에서 링크의 가중치를 측정할 때 그래프 개인화 단계를 제시하고, 개인화된 그래프를 사용하여 LOD-기반 추천 시스템에서 의미 유사성을 계산하는 방법을 제안하였다. 평가 결과는 다른 거리 측정 방법을 능가하는 성능을 보여주었다. 이 결과는 그래프에서 단지 리소스간의 거리만 고려하여 의미 거리 계산을 하는 것 보다는 각각의 링크를 고려하여 가중치를 부여하는 것이 LOD-기반 추천 시스템의 정확도가 향상되었음을 보여주었다. 향후 연구로는 다른 방법의 LOD-기반 유사도 측정을 사용하여 모델에 미치는 영향을 분석하는것을 목표로 한다.

REFERENCES

[1] T. D. Noia & V. C. Ostuni. (2015). Recommender System and Linked Open Data. Reasoning *Web 2015: Reasoning Web. Web Logic Rules,* 88-113.
DOI: 10.1007/978-3-319-21768-0_4

[2] J. G. Cho. (2020). A New Semantic Distance Measurement Method using TF-IDF in Linked Open Data. *Journal of the Korea Convergence Society, 11(10),* 89-96.
DOI : 10.15207/JKCS.2020.11.10.089

[3] S. Alfarhood, K. Labille & S. Gauch. (2017) PLDSD: Propagated Linked Data Semantic Distance. *IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises(WETICE),* 278-283.
DOI: 10.1109/WETICE.2017.16

[4] A. Passant. (2010, March). Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations. *In AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, 77,* 123-129.

[5] G. Piao & J. G. Breslin. (2016). Measuring Semantic Distance for Linked Open Data-enabled Recommander Systems. *SAC '16: Proceedings of the 31st Annual ACM Symposium on Applied Computing,* 315-320.
DOI: 10.1145/2851613.2851839

[6] C. Figueroa, I. Vagliano, O. R. Rocha & M. Morisio. (2015, December). A systematic literature review of Linked Data-based recommender systems. *Concurrency and Computation: Practice and Experience, 27(17),* 4659-4684.
DOI: 10.1002/cpe.3449

[7] G. Piao, S. S. Ara & J. G. Breslin. (2015). Computing the Semantic Similarity of Resources in DBpedia for Recommendation Purposes. *In 5th Joint International Semantic Technology Conference,* 185-200, Springer, Cham.
DOI: 10.1007/978-3-319-31676-5

[8] C. Musto, P. Lops, P. Basile, M. D. Gemmis & G. Semeraro. (2016). Semantics-aware Graph-based Recommender Systems Exploiting Linked Open Data. *In UMAP'16,* 229-237.
DOI: 10.1145/2930238.2930249

[9] T. D. Noia, C. Magarelli, A. Maurino, M. Palmonari & A. Rula. (2018). Using Ontology-Based Data Summarization to Develop Semantics-Aware Recommender Systems. *In ESWC 2018,* 128-144.

[10] A. Passant. (2010). Dbrec: Music Recommendations Using DBpedia. *In ISWC 2010- Volume Part II, Springer-Verlag,* 209-224.

[11] F. Ricci, L. Rokach & B. Shapira. (2015). Recommender systems: introduction and challenges. *In Recommender systems handbook. Springer International Publishing,* 1-34.

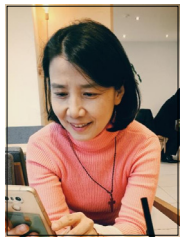
[12] Google. (2021). *Movielens 1M Dataset.* grouplens [Online].
https://grouplens.org/datasets/movielens/1m/

[13] Google. (2021). *MappingMovielens2DBpedia.* researchGate [Online].
https://www.researchgate.net/publication/297369577_mapping-movielens-dbpedia

[14] D. Khongorzul, S. M. Lee & M. H. Kim. (2019). OrdinalEncoder based DNN for Natural Gas Leak Prediction. *Journal of the Korea Convergence Society, 10(10),* 7-13.
DOI : 10.15207/JKCS.2019.10.10.007

[15] J. G. Cho. (2020). A User's location localization method using Smartphone sensor on a subway. *Journal of the Korea Convergence Society, 11(3),* 37-43.
DOI : 10.15207/JKCS.2020.11.3.037

허 원 회(Huh, Wonwhoi) **[정회원]**



- 1993년 2월 : 국민대학교 전자공학과 (공학사)
- 1997년 5월 : Pratt Institute Computer Graphics (MFA)
- 2012년 8월 : 서울과학기술대학교 IT 디자인 융합대학 디지털콘텐츠디자인 전공(디자인학박사)
- 2004년 3월 ~ 현재 : 성결대학교 미디어 소프트웨어학과 교수
- 관심분야 : IT, 콘텐츠, 디자인, 진로
- E-Mail : wonwhoi@sungkyul.ac.kr