https://doi.org/10.5392/JKCA.2020.20.03.018

콘텐트 노드의 유사성 제어를 통한 그래프 구조 데이터 검색의 다양성 향상

Improving Diversity of Keyword Search on Graph-structured Data by Controlling Similarity of Content Nodes

박창섭
동덕여자대학교 컴퓨터학과

Chang-Sup Park(cspark@dongduk.ac.kr)

요약

최근 소셜 네트워크, 시맨틱 웹 등 여러 분야에서 그래프 구조 데이터가 널리 사용됨에 따라 대량의 그래프 데이터에 대한 효과적이고 효율적인 검색 방법의 필요성이 커지고 있다. 기존 키워드 기반 검색 방법들은 대부분 주어진 질의에 대한 연관도만을 고려하여 결과를 구한다. 그러나 이런 방법은 질의 연관도는 높지만 콘텐트 노드들을 공유하는 유사한 결과들이 함께 선택될 가능성이 높다. 이런 문제점을 개선하기 위해 본 논문에서는 키워드 질의에 대한 답 트리에 포함된 콘텐트 노드들의 유사성을 제어하여 콘텐트 노드가 다양한 답 트리들을 구하는 top-k 검색 방법을 제안한다. 다양한 답 트리 집합의 기준을 정의하고, 다양한 top-k 결과 집합을 구하기 위한 두 가지 방법으로 점진적 나열 알고리즘과 A* 탐색 기법을 이용한 휴리스틱 탐색 알고리즘을 설계한다. 또 휴리스틱 탐색의 성능을 높이기 위한 개선 방법을 제시한다. 실 데이터를 이용한 성능 실험 결과를 통해, 본 논문에서 제안한 휴리스틱 탐색 방법이 질의 연관성뿐만 아니라 콘텐트 노드들의 상이도가 높은 다양한 답 트리들을 효율적으로 구할 수 있음을 보인다.

■ 중심어 : | 그래프 구조 데이터 | 키워드 검색 | top-k 질의 | 다양화 | A* 탐색 |

Abstract

Recently, as graph-structured data is widely used in various fields such as social networks and semantic Webs, needs for an effective and efficient search on a large amount of graph data have been increasing. Previous keyword-based search methods often find results by considering only the relevance to a given query. However, they are likely to produce semantically similar results by selecting answers which have high query relevance but share the same content nodes. To improve the diversity of search results, we propose a top-k search method that finds a set of subtrees which are not only relevant but also diverse in terms of the content nodes by controlling their similarity. We define a criterion for a set of diverse answer trees and design two kinds of diversified top-k search algorithms which are based on incremental enumeration and A* heuristic search, respectively. We also suggest an improvement on the A* search algorithm to enhance its performance. We show by experiments using real data sets that the proposed heuristic search method can find relevant answers with diverse content nodes efficiently.

■ keyword : | Graph-structured Data | Keyword Search | Top-k Query | Diversification | A* Search |

* 본 연구는 2018년도 동덕여자대학교 학술연구비 지원에 의해 수행되었습니다.
접수일자 : 2019년 12월 27일
심사완료일 : 2020년 01월 28일
수정일자 : 2020년 01월 28일
교신저자 : 박창섭, e-mail : cspark@dongduk.ac.kr

I. 서 론

최근 소셜 네트워크(social network), 시맨틱 웹(semantic Web), 바이오-인포매틱스(bio-informatics), 빅 데이터(big data) 등 여러 분야에서 객체들의 정보 및 객체들 간의 관계를 그래프 구조로 표현한 데이터와 이를 이용한 응용들이 증가하고 있다. 이에 따라 대량의 그래프 구조 데이터에서 원하는 정보를 효율적으로 찾기 위한 검색 기술에 관한 연구가 활발히 이루어지고 있다. SPARQL[1], Cypher[2] 등과 같은 패턴 기반 질의 언어는 사용자가 질의어의 문법과 사용법을 알아야 할뿐만 아니라 주어진 그래프 데이터의 스키마를 이해하고 올바른 패턴 질의를 작성해야 하는 어려움이 있다. 이에 대한 대안으로 기존 정보 검색에서 널리 사용되는 키워드 검색 방법을 적용하는 연구들이 수행되고 있다. 이 방법은 사용자가 입력한 키워드들이 모두 포함된 서브트리(subtree)나 서브그래프(subgraph)들을 찾는다. 입력된 키워드가 하나 이상 포함된 콘텐트 노드(content node)들의 집합을 포함하는 서브트리나 서브그래프는 그 노드들이 그래프에서 서로 어떤 관계로 연결되어 있는지를 보여준다. [그림 1]은 그래프 구조를 갖는 영화 시맨틱 웹 데이터의 예로, 사각형으로 표현된 노드들이 검색 가능한 키워드들을 포함하는 콘텐트 노드를 나타낸다.

대규모 그래프 데이터에는 주어진 질의에 대한 검색 결과들이 매우 많이 존재할 수 있으므로, 질의에 대한 각 결과의 연관도(relevance)를 측정하여 연관도가 가장 높은 k개의 결과들을 구하는 top-k 검색 방법이 널리 사용된다. 그러나 검색된 결과들이 같은 콘텐트 노드들을 서로 공유할 경우, top-k 질의 결과가 구조적, 의미적으로 유사한 결과 구조들로 이루어지게 된다. 주어진 그래프 데이터가 연결성(connectivity)이 높고 질의와 연관된 콘텐트 노드들이 다양한 경로로 연결되어 있을수록 같은 콘텐트 노드들을 포함하는 결과들이 함께 검색될 가능성이 높아진다. 일반적으로 키워드 검색 방식은 사용자가 자신의 검색 의도를 정확하게 표현하는데 한계가 있으므로, 사용자에게 다양한 결과들을 제공하여 선택하도록 하는 것이 사용자의 검색 만족도를 높일 수 있다. 따라서 질의 연관도뿐만 아니라 콘텐트

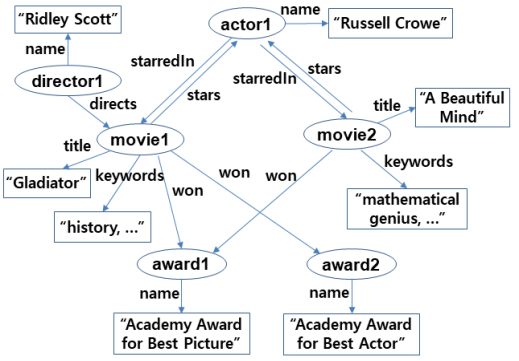

그림 1. 그래프 구조 데이터의 예

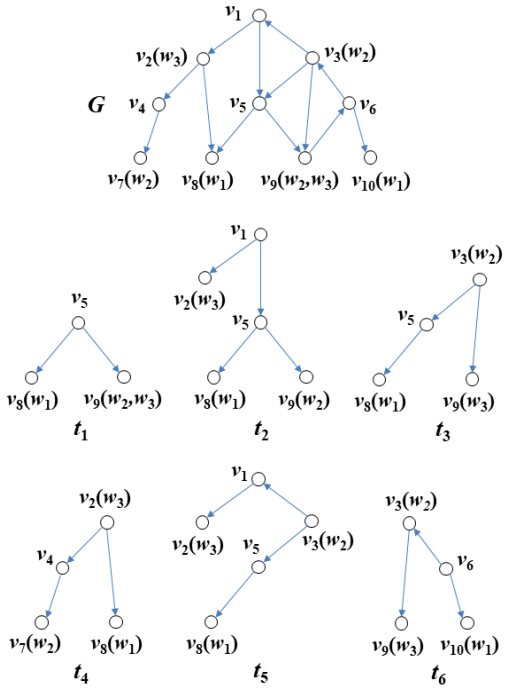

그림 2. 그래프 데이터에 대한 키워드 검색 결과의 예

노드 집합 및 구조가 상이한 다양한 결과들을 찾을 수 있는 검색 방법이 필요하다.

예를 들어, [그림 2]의 G와 같은 그래프 데이터에서 키워드 w_1, w_2, w_3가 노드 v_i에 그림과 같이 포함되어 있다고 가정하자($1 \le i \le 10$). 이 그래프 데이터에 대해 키워드 질의 $q = \{w_1, w_2, w_3\}$가 주어졌을 때, 서브트리 t_1, t_2, ..., t_6는 이 질의의 모든 키워드들을 포함하므로

이 질의의 결과들이 될 수 있다. 이 결과 트리들 중 일부는 같은 콘텐트 노드들을 공유한다. 예를 들어 t_1, t_2, t_3는 노드 v_8, v_9을 공유하고, t_4와 t_5는 노드 v_2, v_8을 포함한다. 만일 top-3 검색 결과로 t_1, t_2, t_3가 선택된다면 이 서브트리들은 질의와의 연관성은 높지만 콘텐트 노드들의 중복이 많아 의미적으로 유사한 결과를 나타낼 가능성이 크다. 한편, t_1, t_4, t_6를 top-3 검색 결과로 선택한다면 이들 간에는 중복된 콘텐트 노드가 적으므로 앞의 검색 결과보다 더 다양한 결과를 사용자에게 제공할 수 있다.

본 논문에서는 이와 같이 그래프 데이터에서 다양한 검색 결과들의 구하기 위한 다양화된 top-k 키워드 검색 방법을 제안한다. 본 논문의 연구 방향 및 구성은 다음과 같다. 2장에서 본 연구와 관련된 기존 연구들에 대해 분석하고, 3장에서는 검색 결과 트리에 포함된 콘텐트 노드들의 중복성을 바탕으로 두 검색 결과의 상이도(dissimilarity)를 정의하고, 검색 결과들의 평균 상이도가 주어진 기준 값보다 크면서 질의 연관도가 가장 높은 다양한 top-k 답 트리 집합을 구하는 문제를 정의한다. 4장에서는 이 문제의 답을 효율적으로 찾을 수 있는 질의 처리 알고리즘으로 답 트리 집합들을 점진적으로 나열하는 알고리즘과 A* 탐색 기법을 이용한 휴리스틱 탐색 알고리즘을 설계한다. 5장에서는 실제적인 그래프 데이터를 이용한 실험을 통해 제안한 알고리즘들의 성능을 평가하고, 6장에서 결론을 맺는다.

II. 관련 연구

그래프 데이터 상의 키워드 질의에 대해 질의 연관도가 가장 높은 top-k 서브트리들의 집합을 효율적으로 구하기 위한 다양한 방법들이 제안되었다. 이 방법들은 질의 연관도의 정의와 결과 집합의 제약조건에 따라 스테이너 트리 시맨틱(Steiner tree semantic)을 따르는 방법들[3-5]과 개별 루트 시맨틱(distinct root semantic)에 기반한 방법들[6-10]로 구분될 수 있다. 또 질의 결과로 질의 키워드를 포함하면서 연관도가 높은 서브그래프들의 집합을 효율적으로 구하기 위한 연구들도 수행되었다[11-13].

이런 방법들은 일반적으로 질의 결과들의 유사성을 고려하지 않고 질의 연관성만을 기준으로 top-k 결과들을 탐색하므로, 유사한 결과들이 함께 검색될 가능성이 있다. 개별 루트 시맨틱을 이용한 방법에서는 루트 노드는 다르지만 콘텐트 노드들이 동일하거나 중복된 서브트리들이 함께 검색되어 질의 결과의 다양성을 저하시킬 수 있다. 그래프 데이터에 대한 키워드 검색에서 결과 트리들 간의 구조적인 유사성과 중복성의 문제는 [4]에서 처음 지적되었다. [14]에서는 주어진 질의 키워드들을 모두 포함하는 콘텐트 노드 집합들 중 중복된 노드가 전혀 없으면서 크기가 가장 작은 것들을 찾기 위한 방법을 제안하였다. [8]에서는 개별 루트 시맨틱을 따르면서 불필요한 노드들과 경로들을 제외시킨 비-중복적인 서브트리들을 구하는 방법을 제안하였다. 또 [9]에서는 콘텐트 노드 집합들이 노드를 공유하지 않으면서 질의 연관도가 가장 높은 top-k 서브트리들을 효율적으로 찾는 알고리즘이 제안되었다. [5]에서는 콘텐트 노드들뿐만 아니라 내부 노드들과 간선들 간에도 중복이 전혀 없으면서 크기가 가장 작은 서브트리들의 집합을 찾는 방법을 제안하였다. 최근 [10]에서는 질의 결과로 검색되는 서브트리들이 같은 루트 노드를 공유할 수 있되, 루트 노드들의 중복도를 제어할 수 있는 top-k 검색 알고리즘을 제시하였다. 또 [15]에서는 그래프에 속한 노드들의 타입 정보를 고려하여 주어진 질의 키워드들과 연관성이 높은 다양한 타입의 노드들을 포함하는 결과 집합을 구하는 키워드 검색 방법을 제안하였다. 한편, [16]에서는 RDF 데이터에 대한 키워드 검색에서 주어진 질의에 대한 다양한 해석을 나타내는 다양한 패턴 그래프들을 생성하는 방법을 연구하였다.

상기 기존 연구들과 본 연구의 차이점은 다음과 같다. [8][10]에서 제안된 방법들은 본 연구와 달리 콘텐트 노드 집합들의 중복성을 고려하지 않는다. [9][14]의 방법들은 top-k 결과들에 포함된 콘텐트 노드 집합들 간에 공유되는 노드가 전혀 없는 결과들을 찾는 반면, 본 연구에서 제안하는 방법은 콘텐트 노드 집합들 간에 일부 노드들을 공유하는 결과들을 구할 수 있다. 또 [5]에서 제안된 방법은 검색 결과들의 내부 노드들과 간선들 간에도 중복을 제거함으로써 지나치게 제한적인 검색 결과들만을 생성한다. [15]의 방법은 그래프 데이터

의 노드들에 대한 타입 정보가 필요하고, 검색 결과의 크기 대신 노드 타입에 기반한 다양성 조건을 이용하므로 원하는 결과의 크기를 지정하거나 예측할 수 없다. 또한 본 연구에서 제안하는 방법은 사용자가 검색 결과들의 유사도를 미리 설정하여 제어할 수 있다.

한편, 기존의 정보 검색, 웹 검색, 추천 시스템 등에서 키워드 검색 결과의 다양성을 높이기 위한 방법들이 연구되었다[17-20]. 그러나 이 방법들은 검색의 대상이 되는 데이터의 구조나 질의 결과 구조, 그리고 질의 연관도 척도 등이 매우 다르기 때문에 그래프 데이터에 대한 키워드 검색에 적용하기에 적합하지 않다.

III. 다양화된 Top-k 검색

본 논문에서 검색 대상이 되는 그래프 데이터는 하나의 유향 그래프 $G(V, E)$로 표현된다. V는 노드들의 집합으로 각 노드는 검색가능한 키워드들을 포함하는 텍스트 데이터를 갖는다. 키워드 w를 포함하는 노드를 w에 대한 콘텐트 노드라 부르고 그런 노드들의 집합을 $V(w)$라 표기한다. 간선은 인접한 두 노드 사이의 관계를 나타내고 그들 간의 의미적 거리를 뜻하는 가중값(weight)을 갖는다. 그래프 데이터에 대한 키워드 질의의 답 트리(answer tree)는 다음과 같이 정의된다.

정의 1. (답 트리) 그래프 데이터 $G(V, E)$에 대해 l개의 키워드로 구성된 질의 $q = \{w_1, w_2, \cdots, w_l\}$가 주어질 때, 각 키워드 w_i를 포함하는 콘텐트 노드들의 집합 $C = \{v_i \mid v_i \in V(w_i), 1 \le i \le l\}$가 존재하고 V에 속한 노드 r에서부터 C에 속한 모든 노드까지 경로들이 존재한다고 가정하자. G의 서브트리 중 r을 루트 노드로 갖고 C에 속한 모든 노드들과 r에서부터 C에 속한 각 노드까지의 최단 경로들을 포함하며, 단말 노드들은 모두 C에 속하는 서브트리 t를 질의 q에 대한 *답 트리*라 하고, $t(r, C)$로 나타낸다. □

주어진 키워드 질의 q에 대해 위 정의 1을 따르는 답 트리는 일반적으로 여러 개가 존재할 수 있다. q에 대한 모든 답 트리들의 집합을 $T(q, G)$라 표기한다. top-k 검색에서는 이들 중 질의와 연관성이 가장 높은 k개의 답 트리들을 구하기 위해 질의에 대한 각 답 트리의 연관도를 계산한다. 검색 결과로 서브트리를 이용하는 방법들은 일반적으로 각 질의 키워드를 포함하는 콘텐트 노드들의 키워드 연관도와, 서브트리에서 루트 노드와 각 콘텐트 노드 간의 근접도를 기반으로 서브트리의 질의 연관도를 계산한다[6-10]. 본 논문에서는 기존 연구[9][10]에서 정의된 연관도 척도를 이용하며, 질의 q에 대한 답 트리 t의 연관도를 $rel(t, q)$로 나타낸다.

주어진 질의에 대한 다양한 결과들을 구하기 위해 두 답 트리 사이의 상이도를 정의한다. 두 답 트리 $t_1(r_1, C_1)$, $t_2(r_2, C_2)$의 상이도는 두 답 트리의 콘텐트 노드 집합 C_1, C_2 사이의 자카드 거리(Jaccard distance)를 기반으로 다음과 같이 계산된다. 즉,

$$dissim(t_1(r_1, C_1), t_2(r_2, C_2)) = 1 - \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$$

위 값은 [0, 1] 범위에 속한다. 두 답 트리의 콘텐트 노드들이 정확히 일치할 경우 두 답 트리의 상이도는 0이고, 콘텐트 노드들이 모두 다를 경우 상이도는 1이 된다.

예를 들어, [그림 2]의 답 트리들 중 $t_2(v_1, \{v_2, v_8, v_9\})$와 $t_3(v_3, \{v_3, v_8, v_9\})$의 상이도는

$$dissim(t_2, t_3) = 1 - \frac{|\{v_8, v_9\}|}{|\{v_2, v_3, v_8, v_9\}|} = 0.5$$

이고, $t_5(v_3, \{v_2, v_3, v_8\})$와 $t_6(v_6, \{v_3, v_9, v_{10}\})$의 상이도는

$$dissim(t_5, t_6) = 1 - \frac{|\{v_3\}|}{|\{v_2, v_3, v_8, v_9, v_{10}\}|} = 0.8$$

이다. 두 개 이상의 다양한 답 트리들의 집합은 위의 상이도 척도를 기반으로 다음과 같이 정의된다.

정의 2. (다양한 답 트리 집합) [0, 1] 범위에 속하는 실수 τ와 두 개 이상의 답 트리들의 집합 S가 주어질 때, S에 속한 답 트리들 사이의 평균 상이도가 τ 보다 크면, 즉, 다음 조건을 만족하면, S를 *다양한 답 트리 집합*이라 한다.

$$avg\_dissim(S) = \frac{2}{|S|(|S|-1)} \sum_{t_1, t_2 \in S, t_1 \neq t_2} dissim(t_1, t_2) \ge \tau$$

□

τ의 값은 검색 시스템에서 미리 설정되거나 또는 검색을 실행하는 사용자에 의해 주어질 수 있다.

한편, 답 트리 집합 S의 질의 연관도는 S에 속한 모든 답 트리들의 질의 연관도의 합으로 정의된다. 즉,

$$rel(S,q)=\sum_{t\in S} rel(t,q)$$

그래프 데이터에 대한 다양화된 top-k 검색은 다음과 같은 답 트리 집합을 찾는다.

정의 3. (다양화된 top-k 검색) 키워드 질의 q와 검색 결과의 크기를 나타내는 자연수 k, 그리고 [0, 1] 범위에 속하는 실수 τ가 주어질 때, q에 대한 답 트리 집합들 중 크기가 k이면서 정의 2를 만족하는 다양한 답 트리 집합들의 집합을 D이라 하자. 즉,

$$D=\{S|S\subseteq T(q,G),|S|=k,avg\_dissim(S)\geq \tau\}$$

다양화된 top-k 검색은 D에서 질의 연관도가 가장 큰 집합 S^*를 찾는다. 즉,

$$S^*=\underset{S\in D}{\arg max}\, rel(S,q)=\underset{S\in D}{\arg max}\sum_{t\in S} rel(t,q)$$

위와 같이 평균 상이도가 τ 보다 크고 질의 연관도의 합이 가장 큰 k개의 답 트리들의 집합을 *다양한 top-k 답 트리 집합*이라 한다. □

예 1. 키워드 질의 q에 대해 평균 상이도가 0.8 이상인 다양한 top-3 답 트리 집합을 구한다고 하자 (즉, k = 3, τ = 0.8). 그래프 데이터에서 q와 연관도가 가장 높은 5개의 답 트리를 순서대로 t_1, t_2, t_3, t_4, t_5 라 하고, 이들의 질의 연관도와 두 답 트리 사이의 상이도가 [그림 3]의 노드와 간선에 표시된 것과 같다고 가정하자.

이 그래프 데이터에서 크기가 3인 답 트리 집합들 중 {t_1, t_2, t_3}와 {t_1, t_2, t_4}는 평균 상이도가 각각 약 0.67, 0.57 로 모두 0.8 보다 작으므로 다양한 답 트리 집합이 아니다. 반면 {t_2, t_3, t_4}, {t_1, t_3, t_5}는 평균 상이도가 0.8 이므로 다양한 답 트리 집합에 해당된다. 이 두 집합의 질의 연관도는 각각 2.0, 2.1이므로, 만일 q에 대한 다른 답 트리가 존재하지 않는다고 하면 {t_1, t_3, t_5}가 다양한 top-3 답 트리 집합이 된다. □

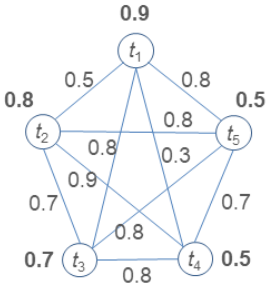

그림 3. 답 트리들의 질의 연관도 및 상이도의 예

본 논문에서는 이와 같은 다양화된 top-k 검색을 위한 효율적인 질의 처리 방법을 제안한다.

IV. 질의 처리 방법

제안하는 질의 처리 방법은 그래프 데이터에서 주어진 질의에 대한 답 트리들을 질의 연관도가 가장 큰 것부터 차례대로 하나씩 구하면서, 그것들 중 정의 2의 다양성 조건을 만족하면서 연관도가 가장 높은 답 트리들의 집합을 찾는 점진적 탐색(incremental search) 방식을 이용한다.

그래프 데이터에서 질의 연관도가 가장 큰 서브트리 또는 서브그래프들을 차례대로 구하기 위한 다양한 방법들이 제안되었다[4][7-9][14]. 예를 들어 [7]에서는 그래프 내의 키워드-노드 경로들에 대한 연관도 기반 역 리스트와 해시 맵 인덱스를 구축 및 활용한다. 본 논문에서는 이런 기존 방법들을 기반으로 주어진 질의에 대해 연관도가 가장 높은 결과, 즉, 답 트리들을 순서대로 하나씩 구해 반환하는 *findNextAnswer* 함수를 이용한다.

1. 점진적 나열

최적의 다양한 top-k 답 트리 집합을 구하기 위한 점진적 나열(incremental enumeration) 알고리즘은 그래프 데이터에서 주어진 질의에 대한 답 트리 집합들을 점진적으로 찾아 나가면서 다음과 같은 성질을 이용하여 불필요한 답 트리 집합들을 효과적으로 제외시킨다. 먼저, 답 트리 집합 S에 속한 모든 답 트리 쌍들의 상이

도의 합을 $sum_dissim(S)$라고 표기한다. 질의 q에 대한 i개의 답 트리들을 포함하는 집합 S_i가 있을 때, S_i에 $(k-i)$개의 새로운 답 트리를 추가하여 총 k개의 답 트리들로 구성되는 집합 S_k를 생성하면 다음이 성립한다 (단, $i \le k$).

$$sum\_dissim(S_k) \le \left(sum\_dissim(S_i) + \left(\frac{k(k-1)}{2} - \frac{i(i-1)}{2}\right)\right)$$

$sum_dissim(S_k)$가 최대인 경우는 S_k에 속한 답 트리 쌍들 중 최소한 어느 하나가 S_i에 속하지 않는 쌍들의 상이도가 모두 1인 경우이다.

만일 S_k가 정의 2를 만족하는 다양한 답 트리 집합이라고 가정하면, 정의 2와 위 식에 의해 다음 식이 성립한다.

$$\tau \le \frac{2}{k(k-1)} sum\_dissim(S_k) \le \frac{2}{k(k-1)}\left(sum\_dissim(S_i) + \left(\frac{k(k-1)}{2} - \frac{i(i-1)}{2}\right)\right)$$

즉,

$$sum\_dissim(S_i) \ge \frac{1}{2}(i(i-1) - k(k-1)(1-\tau)) \qquad (1)$$

따라서 만일 주어진 집합 S_i가 위 조건식 (1)을 만족하지 않으면, S_i의 답 트리들을 모두 포함하고 총 k개의 답 트리를 갖는 다양한 답 트리 집합 S_k는 존재할 수 없다. 이 성질을 이용하면 질의 처리 시 불필요한 답 트리 집합들을 고려 대상에서 제외할 수 있으므로 다양한 top-k 결과 집합을 효율적으로 탐색할 수 있다. 본 논문에서는 위 식 (1)을 답 트리 집합의 다양성 조건(*div-constraint*)이라 부른다. 식 (1)은 $i = k$인 경우 정의 2의 조건과 같다.

점진적 나열 알고리즘을 의사코드로 나타내면 Algorithm 1과 같다. 이 알고리즘에서는 주어진 질의에 대한 다양한 top-k 답 트리 집합을 찾기 위해 k개의 우선순위 큐 Q_1, Q_2, ..., Q_k를 이용해서 답 트리 집합들을 체계적으로 생성한다. Q_i는 i개의 답 트리들로 구성된 답 트리 집합 S_i들을 질의 연관도 $rel(S_i, q)$를 우선순위로 하여 저장한다($1 \le i \le k$). 또 실행 중에 발견되는, 크기가 k인 다양한 답 트리 집합들 중 질의 연관도가 가장 큰 것을 S^*로 나타낸다. *findNextAnswer* 함수를 통해 발견된 새로운 답 트리를 t라 할 때, 각 우선순위 큐 Q_i ($1 \le i \le k-1$)에 속한 각 답 트리 집합 S에 대해 t를 추가한 새로운 답 트리 집합 T를 생성하고 그것을 Q_{i+1}에 저장한다(3~6행, 9행). 이 때 다음과 같은 방법을 통해 필요 없는 답 트리 집합들을 제외시키거나 기존의 우선순위 큐에서 제거한다.

(a) T의 답 트리들을 모두 포함하는, 크기가 k인 답 트리 집합이 가질 수 있는 질의 연관도의 최대값이 현재의 최적 해 S^*의 질의 연관도보다 크지 않으면 T는 S^*보다 더 좋은 결과를 유도할 수 없으므로 더 이상 고려할 필요가 없다(7행). T를 포함하는 k 크기 답 트리 집합의 질의 연관도의 최대값은 T에 추가될 답 트리들의 연관도가 모두 t의 연관도와 같다고 가정하고 계산한다. 또 향후에 발견될 새로운 답 트리들의 연관도는 t보다 크지 않으므로, Q_i에서 S뿐만 아니라 S보다 질의 연관도가 크지 않은 답 트리 집합들은 모두 현재의 최적해 S^*보다 더 좋은 결과를 생성할 수 없다. 따라서 이들을 모두 Q_i에서 제거한다(11행).

(b) T는 식 (1)의 다양성 조건을 만족해야 한다. 그렇지 않으면 T를 포함하는 크기가 k인 다양한 답 트리 집합은 존재할 수 없으므로 T를 우선순위 큐에 추가할 필요가 없다(8행).

Algorithm 1. Incremental Enumeration

```
input: 키워드 질의 q, 자연수 k, 실수 τ ∈[0, 1]
output: 다양한 top-k 답 트리 집합
1   Qi ← ∅ (1≤i≤k);   // k개의 우선순위 큐
2   S* ← ∅;           // 발견된 최적 해
3   while (t ← findNextAnswer(q)) ≠ null do
4     for i ← k-1 to 1 do
5       for-each S∈Qi in a decreasing order of
        rel(S, q) do
6         T ← S ∪ {t};
7         if rel(T, q) + (k-|T|)·rel(t, q) > rel(S*, q)
          then
8           if T satisfies div-constraint then
9             Qi+1.add(T);
10        else
11          Remove all sets S' from Qi such that
            rel(S',q)≤rel(S, q).
12    if S* = ∅ then Q1.add({t});
13    if Qk ≠ ∅ then
14      S* ← Qk.pop();
15      Remove all sets from Qk.
16      for i ← 1 to k-1 do
17        for-each S∈Qi do
18          if S ⊂ S* then Qi.remove(S);
19    if S* ≠ ∅ then
20      if rel(S*, q)≥ max1≤i≤k-1{rel(Qi.top(), q)
                          +(k-i)·rel(t, q)} then
21        break;
22  return S*;
end
```

(c) 크기가 k인 다양한 답 트리 집합 S^*가 이미 발견된 경우, 답 트리 t와 향후 발견될 k-1개의 답 트리로 구성되는 집합은 질의 연관도가 S^*보다 클 수 없으므로 답 트리 집합 $\{t\}$는 고려할 필요가 없다(12행).

(d) Q_k에 새로 추가된, k개의 답 트리를 갖는 다양한 답 트리 집합들은 행 7의 조건에 의해 기존의 S^*보다 질의 연관도가 크므로 그들 중 질의 연관도가 가장 큰 것이 새로운 최적 해 S^*로 선택된다(행 13~15). 이 때, Q_i ($1 \le i \le k$-1)에 포함된 답 트리 집합들 중 S^*의 부분 집합에 해당하는 것들은 향후 새로운 답 트리들을 추가해도 S^*보다 더 좋은 결과를 생성할 수 없으므로 고려 대상에서 제외한다(16~18행).

이 알고리즘의 실행 중 현재의 최적 해 S^*의 질의 연관도가 우선순위 큐들에 포함된 모든 답 트리 집합으로부터 파생될 수 있는 k개의 답 트리 집합들의 질의 연관도들보다 크거나 같을 경우, 전체 그래프 데이터에서 S^*보다 더 좋은 결과는 존재하지 않음이 보장되므로 알고리즘을 종료하고 S^*를 검색 결과로 반환한다(19~22행).

예 2. 예 1과 같이, 어떤 키워드 질의 q에 대해 평균 상이도가 0.8이상인 다양한 top-3 답 트리 집합을 구한다고 하고 (즉, k = 3, τ = 0.8), 그래프 데이터에서 q와 연관도가 가장 높은 5개의 답 트리 t_1, t_2, ..., t_5의 질의 연관도와 두 답 트리 사이의 상이도가 [그림 3]과 같다고 가정하자. 식 (1)의 다양성 조건에 의해, 크기가 2인 답 트리 집합 S가 $sum\_dissim(S) < ½(2 - 1.2) = 0.4$ 이면, S를 포함하는, 크기가 3인 답 트리 집합들은 모두 이 질의의 결과가 될 수 없다.

답 트리 t_1, t_2, t_3, t_4가 차례대로 발견됐을 때, 위 **Algorithm 1**에 의해 각 우선순위 큐에 저장되는 답 트리 집합들은 다음과 같다.

$Q_1 = [\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}]$,

$Q_2 = [\{t_1, t_2\}, \{t_1, t_3\}, \{t_2, t_3\}, \{t_2, t_4\}, \{t_3, t_4\}]$,

$Q_3 = [\{t_2, t_3, t_4\}]$.

$\{t_1, t_2, t_3\}$, $\{t_1, t_2, t_4\}$, $\{t_1, t_3, t_4\}$의 평균 상이도는 각각 약 0.67, 0.57, 0.63으로 모두 τ보다 작으므로 제외되고, $\{t_2, t_3, t_4\}$는 평균 상이도가 0.8이므로 선택된다(8행). $\{t_1, t_4\}$는 $sum\_dissim(\{t_1, t_4\}) = 0.3 < 0.4$ 이므로 다양성 조건을 만족하지 못해 제외된다. 따라서 $\{t_2, t_3, t_4\}$가 최적 해 S^*로 선택되고, 각 우선순위 큐에서 S^*의 부분집합에 해당되는 것들은 모두 제외되어(16~18행),

$Q_1 = [\{t_1\}]$, $Q_2 = [\{t_1, t_2\}, \{t_1, t_3\}]$, $Q_3 = \varnothing$

가 된다. 이 때, $rel(Q_2.top(), q) + 1 \cdot rel(t_4, q) = 1.7 + 0.5 = 2.2$ 로 S^*의 질의 연관도 2.0보다 크므로 알고리즘의 종료 조건(20행)은 만족되지 않는다.

이제 새로운 답 트리 t_5가 발견되면, Q_2에서부터 파생되는 두 답 트리 집합 중 평균 상이도가 τ 보다 큰 $\{t_1, t_3, t_5\}$만이 Q_3에 추가된다. Q_1으로부터 생성되는 $\{t_1, t_5\}$는 그것을 포함하는 크기 3인 답 트리 집합의 연관도의 최대값이 1.9로 S^*의 질의 연관도보다 작으므로 제외된다(7행). 또 S^*가 존재하므로 $\{t_5\}$는 생성되지 않는다(12행). 따라서 $Q_1 = [\{t_1\}]$, $Q_2 = [\{t_1, t_2\}, \{t_1, t_3\}]$, $Q_3 = [\{t_1, t_3, t_5\}]$가 된다. 이 때, $\{t_1, t_3, t_5\}$는 질의 연관도가 2.1로 기존 최적 해 $\{t_2, t_3, t_4\}$보다 크므로 새로운 최적 해 S^*로 선택되고, 우선순위 큐에서 이 집합의 부분집합에 해당되는 것들은 모두 제거되어, $Q_1 = Q_3 = \varnothing$, $Q_2 = [\{t_1, t_2\}]$가 된다. $rel(\{t_1, t_2\}, q) + 1 \cdot rel(t_5, q) = 1.7 + 0.5 = 2.2$로 S^*의 질의 연관도보다 크므로 알고리즘의 종료 조건은 만족되지 않는다. 만일 q에 대한 다른 답 트리가 존재하지 않거나, t_5 다음으로 연관도가 높은 답 트리 t_6의 질의 연관도가 0.3이고 t_1과의 상이도가 0.5라고 하면 $\{t_1, t_2, t_6\}$는 평균 상이도가 τ보다 작아 제외되고 $\{t_1, t_2\}$는 알고리즘의 종료 조건을 만족하여 $\{t_1, t_3, t_5\}$가 전체 그래프 데이터 상에서 다양한 top-k 답 트리 집합으로 결정된다. □

2. 휴리스틱 탐색

앞 절의 **Algorithm 1**에서 정의된 점진적 나열 알고리즘보다 더 적은 개수의 답 트리들을 생성하기 위해, 본 절에서는 A^* 탐색 방법에 기반한 휴리스틱 탐색 알고리즘을 제안한다. 전체적인 알고리즘은 다음과 같다.

그래프 데이터에 대한 키워드 질의 q가 주어질 때, **Algorithm 2**에서는 *findNextAnswer* 함수를 이용하여 q와 연관도가 가장 높은 답 트리들을 차례대로 하나씩 찾고 리스트 *Tops*에 순서대로 저장한다(2~3행). *Tops*가 k개 이상의 답 트리들을 포함할 경우, 그것들

```
Algorithm 2. Heuristic Search
input: 키워드 질의 q, 자연수 k, 실수 τ ∈[0, 1]
output: q에 대한 다양한 top-k 답 트리 집합
1   Tops ← ∅;   // q에 대한 답 트리 리스트
2   while (t ← findNextAnswer(q)) ≠ null do
3     Tops.append(t);
4     if |Tops| ≥ k then
5       answer ← A*_Search(q, k, τ, Tops);
6       if answer ≠ ∅ then
7         return answer;
8   return ∅;
end
```

을 대상으로 A^* 탐색을 실행한다(4~5행). 그 결과 최적의 다양한 top-k 답 트리 집합이 발견되면 그것은 *그래프 데이터 전체에 대한* 최적 해임이 보장되고 알고리즘은 종료한다(6~7행). 만일 *Tops* 내에 최적의 답 트리 집합이 존재하는지 알 수 없으면 A^* 탐색은 최적 해를 구하지 못한다. 이 경우 다시 *findNextAnswer* 함수를 통해 새로운 답 트리를 구하여 *Tops*에 추가한 후 A^* 탐색을 실행하는 과정을 반복한다.

2.1 A* 탐색 알고리즘

질의 q에 대해 연관도가 가장 높은 답 트리들의 정렬된 리스트 *Tops*가 주어질 때, 제안하는 A^* 알고리즘은 *Tops*에 속한 답 트리들의 부분 집합들을 체계적으로 생성하면서 그래프 데이터 전체에서 가장 연관도가 높은 다양한 top-k 답 트리 집합을 찾는다. 본 알고리즘의 탐색 공간은 상태(state)들로 구성되고, 각 상태는 *Tops*로부터 선택된 k개 이하의 다양한 답 트리 집합 *ans*와 그것에 관한 *score*, *pos*, *ub* 속성들을 포함한다. *score*는 *ans*의 질의 연관도 *rel*(*ans*, q)이고, *pos*는 *ans*에 속한 답 트리들 중 *Tops* 안에서 가장 뒤에 위치하는 것의 위치 값, 즉 연관도 순위를 나타낸다. 예를 들어, *Tops* = [t_1, t_2, ... t_{10}] 일 때, 상태 s의 답 트리 집합 *ans*가 {t_2, t_4, t_5} 이면 s의 *pos*는 5이다. *ub*는 현재 상태의 *ans*로부터 유도될 수 있는 다양한 k 크기 답 트리 집합의 질의 연관도의 상한 값(upper-bound)을 나타낸다.

본 A^* 탐색에서 *ub*는 존재하는 상태들 중 확장 대상이 될 상태를 선택하는 기준으로 사용된다. 즉, 아직 탐색되지 않은 상태들 중 *ub* 값이 가장 큰 상태 s를 선택하여 그것의 자식 상태들을 생성한다. 그 자식 상태들은 s의 답 트리 집합 *ans*에 새로운 답 트리 하나를 추가한 새로운 답 트리 집합을 갖는데, 추가될 답 트리는 *Tops*에서 s의 답 트리들보다 연관도 순위가 낮은, 즉, s의 *pos*보다 뒤에 위치한 답 트리들 중에서만 선택된다.

따라서 크기가 k보다 작은 답 트리 집합을 갖는 상태 s의 *ub* 값, 즉 s로부터 파생될 수 있는 최적의 k 크기 답 트리 집합의 연관도는 다음과 같이 계산될 수 있다. *Tops*에서 s의 *pos* 다음에 있는 답 트리부터 차례대로 k-|*ans*|개의 답 트리들을 선택하여 총 k개의 답 트리들로 이루어지는 집합을 고려하면 그 집합의 질의 연관도가 s의 *ub* 값이 된다. 만일 *Tops* 내의 답 트리들만으로는 k 크기 답 트리 집합을 구성할 수 없다면, 그래프에서 아직 발견되지 않은 답 트리들의 질의 연관도가 모두 *Tops*의 마지막 (즉, 연관도가 가장 작은) 답 트리와 같다고 가정하고 총 k개의 질의 연관도들의 합으로 *ub*를 계산한다. 이 알고리즘의 형식적인 정의는 생략하고, 이를 구현한 함수를 *computeUB*라 부른다. 예를 들어, k = 5이고 *Tops* = [t_1, t_2, ... t_{10}] 일 때, 상태 s의 답 트리 집합 *ans*가 {t_2, t_4, t_5} 이면 *pos* = 5이므로 *ub* = *rel*({t_2, t_4, t_5, t_6, t_7}, q)이다. 만일 s의 *ans*가 {t_2, t_4, t_9} 이면 *pos* = 9이고 *ub* = *rel*({t_2, t_4, t_9, t_{10}}, q) + *rel*({t_{10}}, q)가 된다. 만일 상태 s의 *ans*가 이미 k개의 답 트리를 포함하면 *ub*는 *ans*의 *score*, 즉 질의 연관도와 같은 값을 갖는다.

Algorithm 3은 제안하는 A^* 탐색 알고리즘의 의사코드이다. 이 알고리즘에서는 새로 발견되는 답 트리 집합에 대해 새로운 상태를 생성하고 그것의 *ub* 값을 우선순위로 사용하여 우선순위 큐 H에 저장한다. 3~18행의 중심 루프를 반복할 때마다 H에서 *ub*가 가장 큰 상태 s를 선택한다. s가 나타내는 답 트리 집합 *ans*가 k개 미만의 답 트리를 포함한 경우, *Tops*에서 아직 고려되지 않은 새로운 답 트리 하나를 선택해서 추가한 새로운 답 트리 집합 *ans'*을 고려한다. *ans'*이 식 (1)의 다양성 조건을 만족하면 그것으로부터 크기가 k인 다양한 답 트리 집합이 유도될 수 있으므로, *ans'*을 나타내는 새로운 자식 상태 e를 생성한다. 이 때 e의 *ub*는 앞에서 기술한 *computeUB* 함수를 이용하여 계산한다. 그리고 생성된 상태 e를 우선순위 큐 H에 저장한다(7~11행).

Algorithm 3. ***A* Search***

```
input: 키워드 질의 q, 자연수 k, 실수 τ ∈[0, 1],
       q에 대한 답 트리 리스트 Tops
output: q에 대한 다양한 top-k 답 트리 집합
1    H ← (∅, 0, 0, 0);    // 상태들의 우선순위 큐
2    ub_unseen ← 0;
3    while (s ← H.pop()) ≠ null do
4      if |s.ans| < k then
5        for i ← s.pos+1 to |Tops| do
6          t_i ← i-th answer tree in Tops;
7          ans' ← s.ans∪{t_i};
8          if ans' satisfies div-constraint then
9            ub ← computeUB(s, t_i, q, k, Tops);
10           e ← (ans', s.score+rel(t_i, q), i, ub);
11           H.add(e);
12       if no child state of s has been created then
13         ub_unseen ← max {ub_unseen,
                   s.score+(k-|s.ans|)·rel(t_|Tops|, q)};
14     else  // |s.ans| = k
15       if s.score ≥ ub_unseen then
16         return s.ans;
17       else
18         return ∅;
19   return ∅;
end
```

만일 s의 답 트리 집합 ans에 추가할 수 있는 새로운 답 트리가 $Tops$에 존재하지 않거나(즉, $s.pos = |Tops|$), ans에서 확장된 답 트리 집합이 다양성 조건을 만족하지 않을 경우, ans의 답 트리들과 $Tops$에 속하지 않은(즉, 그래프에서 아직 발견되지 않은) 답 트리들로 구성되는, 크기가 k인 다양한 답 트리 집합이 가질 수 있는 질의 연관도의 상한 값을 계산하여 ub_unseen의 값으로 유지한다(12~13행).

한편, s의 답 트리 집합 ans가 k개의 답 트리를 포함할 경우, 만일 그것의 질의 연관도가 현재의 ub_unseen 값보다 크거나 같으면 ans가 그래프 데이터 전체에 대한 다양한 top-k 답 트리 집합임이 보장되므로 A^* 탐색 알고리즘을 종료하고 ans를 질의 결과로 반환한다(14~16행). 만일 그렇지 않다면, ans는 그래프 전체의 최적 해임을 보장할 수 없다. 또 ans 뿐만 아니라 H에 포함된 다른 상태들로부터 유도될, 크기가 k인 다양한 답 트리 집합들은 모두 연관도가 ub_unseen 보다 작을 수밖에 없으므로, $Tops$에서는 최적의 다양한 top-k 결과 집합을 구하지 못하고 A^* 탐색을 종료한다(17~18행).

예 3. 예 2와 같이, 질의 q에 대해 평균 상이도가 0.8 이상인 다양한 top-3 답 트리 집합을 구한다고 가정하고, 그래프 데이터에서 질의 연관도가 가장 높은 5개의 답 트리 t_1, t_2, ..., t_5의 연관도와 상이도가 [그림 3]과 같다고 가정하자. [그림 4]는 $Tops$ = [t_1, t_2, ... t_5] 일 때 상기 A^* 탐색 알고리즘에 의한 상태들의 생성 과정을 나타낸다. 색칠된 상태 s_0, s_1, s_6, s_7, s_8은 우선순위 큐 H에서 가장 큰 ub를 갖고 있어 선택 및 확장된 상태들이다. 상태 s_6로부터 파생되는 세 개의 자식 상태들은 모두 답 트리 집합이 다양성 조건을 만족하지 않아 상태가 생성될 수 없다(8행). 이 때 ub_unseen 값이 2.2로 계산된다(12~13행). 상태 s_7의 두 번째 자식 상태 s_8의 답 트리 집합 {t_1, t_3, t_5}는 평균 상이도가 주어진 기준을 만족하므로 상태가 생성될 수 있고, 이 상태는 우선순위 큐에서 가장 큰 ub 값을 가지므로 다음 단계에서 선택된다. 이 답 트리 집합은 $Tops$에 존재하는 최적의 다양한 답 트리 집합이지만, 질의 연관도가 ub_unseen보다 작으므로 그래프 전역적인 최적 해임을 보장할 수 없다. 따라서 이 A^* 탐색은 최적 해를 구하지 못하고 종료한다(14~18행). □

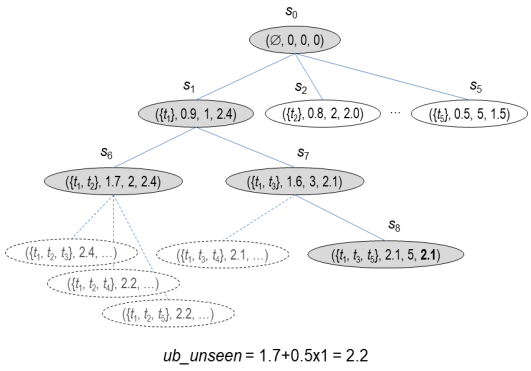

그림 4. ***A** 탐색 알고리즘의 실행 예**

2.2 탐색 결과를 활용한 알고리즘 개선

앞에서 기술한 휴리스틱 탐색 알고리즘은 주어진 $Tops$ 리스트를 대상으로 A^* 탐색을 실행했을 때 그래프 전역적인 최적 해를 구하지 못하면 새로운 답 트리가 추가된 $Tops$ 리스트를 대상으로 A^* 탐색을 처음부터 다시 시작해야 한다. 이로 인해 많은 상태들의 재생성이 필요한데, 이를 줄이기 위해 이전 A^* 탐색에서 발견된, 크기가 k인 답 트리 집합의 질의 연관도를 다음

A^* 탐색에서 *ub*의 하한(lower-bound)으로 활용하는 것을 고려한다.

앞 절의 **Algorithm 3**에서는 우선순위 큐 H에서 선택된 상태 s가 k개의 다양한 답 트리를 가진 상태인 경우(14행), 15행의 조건이 만족되지 않아서 그것을 그래프 전역적인 최적 해로 확정할 수 없을 경우에도 하나의 후보 해가 발견된 것이므로 그것의 *score*, 즉, 질의 연관도는 다음 A^* 탐색 시 생성이 필요한 상태들을 선별하는 기준이 될 수 있다. 즉, 그 후보 해의 질의 연관도보다 *ub*가 작은 상태들은 그 후보해보다 연관도가 더 큰 다양한 k 크기 답 트리 집합을 생성할 수 없으므로, 그 *score*를 하한 값으로 사용하여 불필요한 상태들의 생성을 피할 수 있고, 이를 통해 새로운 A^* 탐색의 성능을 향상시킬 수 있다.

이를 위해 **Algorithm 3**의 18행에서 탐색을 종료하기 전에 현재 발견된 후보 해의 *score*를 정적 변수를 통해 저장해 두거나 또는 결과로 반환 및 재입력 받아 다음 A^* 탐색 실행 시에 활용할 수 있다. 전자의 방법을 이용할 경우 **Algorithm 3**에서 다음과 같은 수정이 필요하다.

(a) 중심 루프(3~18행)가 시작되기 전에 정적 변수 *lowerbound*를 선언한다.

```
static lowerbound ← 0;
```

(b) 현재 상태 s의 자식 상태 e의 *ub*가 *lowerbound* 보다 크거나 같은 경우에만 e를 생성한다(10~11행). 그렇지 않을 경우, e 뿐만 아니라 그 후의 자식 상태들도 모두 *ub*가 *lowerbound* 보다 작으므로 더 이상 s의 자식 상태 생성을 진행할 필요 없이 s에 대한 확장을 끝낸다.

```
if ub ≥ lowerbound then
    e ← (ans', s.score+rel(t_i, q), i, ub);
    H.add(e);
else break;
```

(c) 탐색 중 크기가 k인 답 트리 집합을 찾았으나 그것이 그래프 전체의 최적 해임을 보장할 수 없는 경우 그것의 질의 연관도를 *lowerbound*에 저장하고 탐색을 종료한다(17~18행).

```
else
    lowerbound ← s.score;
    return ∅;
```

V. 성능 평가

본 논문에서 제안한 방법의 효과와 성능을 평가하기 위해 제안한 방법들을 Java 언어로 구현하고 실 데이터를 이용한 실험을 실시하였다. 실험에 사용된 데이터는 IMDB[1]에서 추출한 영화 관련 데이터로, 영화, 배우, 감독, 출연 등에 대해 검색 가능한 키워드들을 포함하는 콘텐트 노드들과, 노드들 사이의 관계를 나타내는 간선들로 이루어 진 그래프를 생성하였다. 그래프에 포함된 노드와 간선들의 개수는 각각 약 109만 개와 307만 개이며, 노드들에 포함된 키워드의 개수는 약 576만 개이다. 이 그래프 구조의 생성 및 노드들 간의 최단 경로 계산을 위해 JGraphT[2] 라이브러리를 이용하였다. 또 키워드에 대한 노드들의 연관도 계산을 위해 Apache Lucene[3]을 사용하였다. 실험은 Intel Xeon 2.1GHz CPU 4개와 120GB 메모리로 구성된 Linux 서버에서 실행하였다.

[표 1]은 본 실험에 사용된 테스트 질의들의 일부이다. 이 질의들에 대해, 4장에서 기술한 점진적 나열 알고리즘과 휴리스틱 탐색 알고리즘, 그리고 하한 값을 이용한 개선된 휴리스틱 탐색 알고리즘을 실행해서 최적의 다양한 top-k 답 트리 집합을 구하였다. 또, 비교를 위해 [7]에서 제안된 TA 기반 top-k 알고리즘을 이용해서 상이도의 제약이 없는 일반적인 top-k 답 트리들을 구하였다. 그 실행 결과는 순서대로 "Enum", "Heu", "Heu-lb", "Non-div"로 표기한다.

[그림 5]와 [그림 6]은 각 질의에 대해 평균 상이도가 0.7 이상인 다양한 top-5 답 트리 집합을 검색한 결과들의 질의 연관도와 상이도를 나타낸다(즉, τ = 0.7, k = 5). 제안한 세 방법의 검색 결과들은 모두 서로 일치하였고, 평균 상이도가 주어진 값 0.7 이상인 다양한 답 트리들이 검색되었다("Diversified"로 표기됨). 반면,

1 http://www.imdb.com/
2 http://jgrapht.org/
3 http://lucene.apache.org/

표 1. 테스트 질의

ID	키워드 리스트
Q1	Academy, award, drama
Q2	aircraft, ship, battle
Q3	crime, killer, thriller
Q4	disaster, Earth, escape
Q5	history, Rome, emperor
Q6	police, gangster, fight
Q7	SF, fantasy, future
Q8	SF, time, travel
Q9	spy, killer, action
Q10	vampire, zombie, horror

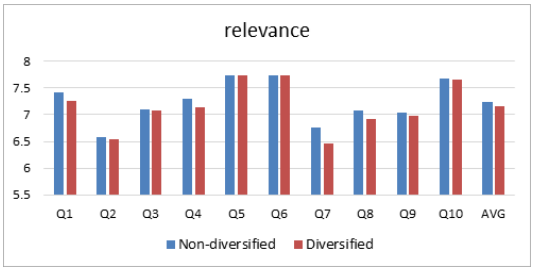

그림 5. top-k 검색 결과의 질의 연관도

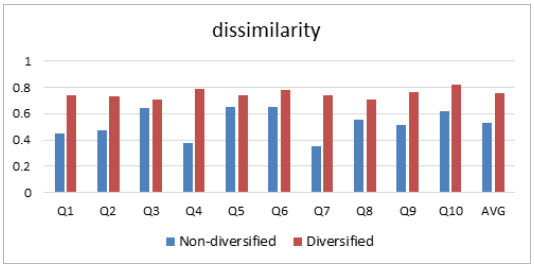

그림 6. top-k 검색 결과의 상이도

상이도의 제약이 없는 top-k 검색 결과들은 평균 상이도가 0.35 ~ 0.65로 나타나 답 트리들 사이에 콘텐트 노드의 중복이 많이 발생함을 알 수 있다(“Non-diversified”로 표기됨). 한편 질의 연관도는 다양화된 top-k 검색이 일반적인 top-k 검색보다 평균 약 1.23% 정도 낮은 결과를 생성하였다.

[그림 7]은 같은 실험에서 알고리즘들의 실행 시간을 측정하여 비교한 결과를 나타낸다. 대부분의 질의에서 다양화된 top-k 검색이 일반적인 top-k 검색보다 실행 시간이 오래 걸림을 알 수 있다. 점진적 나열 알고리즘과 기본적인 휴리스틱 탐색 알고리즘의 평균 실행 시간은 일반적인 top-k 검색 시간에 비해 각각 45.1%, 10.8% 정도 더 오래 걸렸다. 그러나 점진적 나열 방법

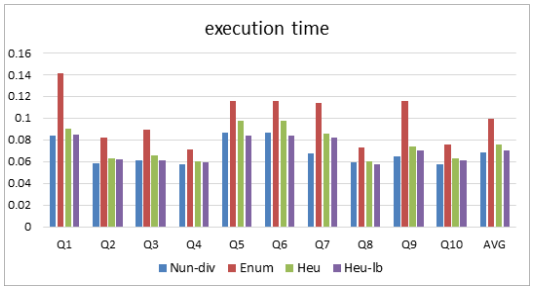

그림 7. 질의별 검색 실행 시간

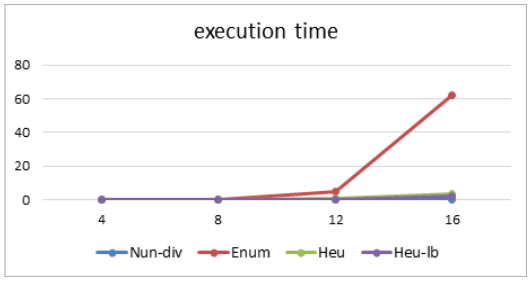

그림 8. 검색 결과의 크기에 따른 평균 실행 시간

에 비해서 휴리스틱 탐색 방법은 실행 시간이 평균 23.7% 정도 감소하였다. 또 하한 값을 이용하는 휴리스틱 탐색 방법은 기본적인 휴리스틱 탐색보다 평균 7.2% 더 적게 걸려 질의 처리 성능이 향상되었음을 알 수 있다.

[그림 8]은 구하는 검색 결과의 크기, 즉 답 트리의 개수(k)에 따른 각 알고리즘의 평균 질의 처리 시간을 측정한 결과를 나타낸다. 결과의 크기가 4에서 16으로 증가할 때 점진적 나열 알고리즘은 평균 실행 시간이 약 0.08초에서 62.3초로 급격히 증가하였으며, k = 16일 때는 질의에 따라 0.34~410초가 소요되어 편차가 매우 크게 나타났다. 휴리스틱 탐색 알고리즘은 결과의 크기가 증가함에 따라 평균 실행 시간이 약 0.07초에서 3.57초로 증가하였고, 하한 값을 이용하여 개선된 휴리스틱 탐색 방법은 평균 약 0.07초에서 2.05초로 그보다 적은 폭으로 증가하여, 검색 결과의 크기가 커질수록 성능 개선의 효과도 더 커짐을 알 수 있다.

VI. 결론

본 논문에서는 그래프 구조 데이터에 대한 키워드 검색 결과의 다양성을 높이기 위해 콘텐트 노드 집합의 유사도를 제한하면서 질의 연관도가 가장 높은 답 트리들을 구하는 다양화된 top-k 검색 방법을 제안하였다. 콘텐트 노드 집합이 상이한 다양한 답 트리 집합의 조건을 정의하고, 다양한 top-k 결과 집합을 구하기 위한 두 가지 방법으로 점진적 나열 알고리즘과 A^* 휴리스틱 탐색 알고리즘을 설계하였다. 또 반복적인 A^* 탐색을 효율적으로 실행할 수 있는 개선 방안을 제시하였다. 그래프 데이터를 이용한 성능 실험을 통해 제안한 방법들의 정확성을 확인하였고, 휴리스틱 탐색 알고리즘이 다양한 콘텐트 노드들을 포함하면서 질의 연관도가 높은 결과들을 효율적으로 구할 수 있음을 보였다. 본 논문에서 제안한 방법은 일반적인 top-k 검색 방법과는 달리 사용자가 검색 결과들의 유사도를 설정 및 제어할 수 있고, 이를 통해 사용자의 요구에 맞는 다양한 검색 결과를 생성할 수 있는 장점이 있다. 본 논문에서는 결과 구조로 서브트리만을 고려했으나, 제안한 top-k 검색 방법은 서브그래프 등 일반적인 결과 구조에 대해 확장 가능하며 결과 집합에 대한 부가적인 제약 조건이 있는 경우에도 적용 가능하다.

한편 본 연구의 한계점은 검색 결과들에 대한 유사도 계산 시 콘텐트 노드 집합의 중복성만 고려하며, 구하는 검색 결과의 수가 증가할수록 검색 알고리즘의 성능이 저하된다는 점 등이다. 이를 개선하기 위해 향후 콘텐트 노드뿐만 아니라 내부 노드나 경로들에 대한 유사성도 고려하여 검색 결과의 다양성을 높이는 연구와, 다수의 결과들을 효율적으로 구하기 위해 주어진 질의 연관도의 최소 한계를 보장하는 근사적인 top-k 질의 처리 알고리즘을 설계하기 위한 연구가 필요하다.

참 고 문 헌

[1] SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/

[2] Cypher Query Language. https://neo4j.com/developer/cypher-query-language/

[3] B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin, "Finding top-k min-cost connected trees in databases," Proc. of ICDE, pp.836-845, 2007.

[4] K. Golenberg, B. Kimelfeld, and Y. Sagiv, "Keyword proximity search in complex data graphs," Proc. of ACM SIGMOD Conference, pp.927-940, 2008.

[5] C. Liu, L. Yao, J. Li, R. Zhou, and Z. He, "Finding smallest k-Compact tree set for keyword queries on graphs using map-reduce," World Wide Web, Vol.19, No.3, pp.499-518, 2016.

[6] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar, "Bidirectional expansion for keyword search on graph databases," Proc. of the 31st Int'l Conference on VLDB, pp.505-516, 2005.

[7] H. He, H. Wang, J. Yang, and P. S. Yu, "BLINKS: ranked keyword searches on graphs," ACM SIGMOD Conference, pp.305-316, 2007.

[8] 박창섭, "그래프 데이터에 대한 비-중복적 키워드 검색 방법," 한국콘텐츠학회논문지, 제16권, 제6호, pp.205-214, 2016.

[9] C. S. Park, "Reducing redundancy in keyword query processing on graph databases," Journal of Information Science and Engineering, Vol.34, No.2, pp.551-574, 2018.

[10] C. S. Park, "Effective keyword search on graph data using limited root redundancy of answer trees," Int'l Journal of Web Information Systems, Vol.14, No.3, pp.299-316, 2018.

[11] L. Qin, J. X. Yu, L. Chang, and Y. Tao, "Querying communities in relational databases," Proc. of IEEE International Conference on Data Engineering, pp.724-735. 2009.

[12] M. Kargar and A. An, "Keyword search in graphs: finding r-cliques," Proc. of the VLDB Endowment, Vol.4, pp.681-692, 2011.

[13] W. Le, F. Li, A. Kementsietsidis, and S. Duan, "Scalable keyword search on large RDF data," IEEE Transaction on Knowledge and Data Engineering, Vol.26, No.11, pp.2774-2788, 2014.

[14] M. Kargar, A. An, and X. Yu, “Efficient duplication free and minimal keyword search in graphs,” IEEE Trans. on Knowledge and Data Engineering, Vol.26, No.7, pp.1657-1669, 2014.

[15] M. Zhong, Y. Wang, and Y. Zhu, “Coverage-oriented diversification of keyword search results on graphs,” Proc. of Int’l Conference on Database Systems for Advanced Applications, pp.166-183, 2018.

[16] A. Dass and D. Theodoratos, “Trading off popularity for diversity in the results sets of keyword queries on linked data,” Proc. of Int’l Conference on Web Engineering, pp.151-170, 2017.

[17] A. Angel and N. Koudas, “Efficient diversity-aware search,” Proc. of ACM SIGMOD Conference, pp.781-792, 2011.

[18] D. Rafiei, K. Bharat, and A. Shukla, “Diversifying web search results,” Proc. of the 19th Int’l Conference on WWW, pp.781-790, 2010.

[19] G. Capannini, F. M. Nardini, R. Perego, and F. Silvestri, “Efficient diversification of web search results,” Proc. of the VLDB Endowment, Vol.4, pp.451-459, 2011.

[20] C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification,” Proc. of the 14th Int’l Conference on WWW, pp.22-32, 2005.

저 자 소 개

박 창 섭(Chang-Sup Park) 정회원

- 1995년 2월 : KAIST 전산학과(공학사)
- 1997년 2월 : KAIST 전자전산학과(공학석사)
- 2002년 2월 : KAIST 전자전산학과(공학박사)
- 2002년 3월 ~ 2005년 2월 : KT 책임연구원
- 2005년 3월 ~ 2009년 2월 : 수원대학교 교수
- 2009년 3월 ~ 현재 : 동덕여자대학교 컴퓨터학과 교수

〈관심분야〉 : 데이터베이스, 정보 검색, 시맨틱 웹