이산 Cuckoo Search 기반 온톨로지 정렬 알고리즘

한군*, 정현준*, 백두권**†
*고려대학교 정보통신대학 컴퓨터 · 전파통신공학과
**고려대학교 융합소프트웨어전문대학원
email : {galmang2015, darkspen, baikdk}@korea.ac.kr

Discrete Cuckoo Search based Ontology Alignment Algorithm

Jun-Han*, Hyunjun Jung*, Doo-Kwon Baik**†
*Dept. of Computer and Radio Communications Engineering, Korea University
** Graduate School of Convergence IT, Korea University

요 약

기존 온톨로지들을 공유 및 재사용하기 위하여 온톨로지 정렬이 연구되고 있다. 기존 정렬 시스템은 온톨로지 데이터 양에 따라 매트릭스를 생성하고 과도한 계산을 통해 처리하여 대용량 데이터 집합에 대하여 공간적 및 계산적으로 부하를 발생하여 효율적이지 않다. 이를 해결하기 위하여 온톨로지 정렬을 휴리스틱 알고리즘을 적용하여 연구 진행하였다. 기존 휴리스틱 알고리즘은 계산이 간단하지만 조율해야 하는 파라미터가 많기에 특정 도메인에 최적 조합이 필요하며 만족한 성능을 얻지 못하였다. 이 논문에서는 Discrete Cuckoo Search(DCS) 기반 온톨로지 정렬 알고리즘을 제안한다. 제안한 알고리즘은 조율해야 하는 파라미터의 개수가 적고 Levy Flight 분포에 따라 탐색하여 계산이 간단하다. 제안된 알고리즘의 성능을 평가하기 위해 OAEI(Ontology Alignment Evaluation Initiative)에서 제공하는 벤치마크 데이터를 사용하여 정확률(Precision)과 재현율(Recall)을 구하고 기존 휴리스틱 정렬 알고리즘과 비교 평가하였다.

키워드: 온톨로지, 온톨로지 정렬, Cuckoo Search 알고리즘

1. 서론

인터넷과 웹 2.0 의 발전으로 인해 웹상의 데이터 양이 폭발적으로 증가 되면서 키워드에 기반을 둔 정보 검색의 정확성이 떨어지고 분석 시간이 늘어나고 있다[1]. 시맨틱 웹은 웹 상의 정보에 의미 부여 하여 컴퓨터가 정보 자원을 읽고 이해하고 해석하여 자동으로 정보를 사용자의 요구에 알맞게 처리해 준다. 온톨로지는 공유된 개념화에 대한 정형화되고 명시적인 명세로서 시맨틱 웹의 핵심적인 구성요소이다[2]. 특정 도메인에 관련된 자원, 속성 그리고 자원간의 관계를 이루는 프로퍼티(property) 요소들을 이용하여 그래프를 형성하고 추론을 하여 사용자가 요구하는 정보 검색의 만족도를 향상하는 장점이 있다. 온톨로지는 지식의 공유와 재사용을 목적으로 한 연구분야로 시맨틱 웹 이외에도 인공지능, 정보시스템, 데이터 통합, 정보 검색, 지식관리 등 분야에서 활발히 개발되면서 온톨로지 및 데이터 집합은 증가하였다[3].

대부분 온톨로지는 사용자의 주관성에 의해 같은 도메인을 표현함에도 서로 다른 형태로 구성되었다. 같은 개념이 다양한 의미로 표현되고 같은 의미를 다양한 개념으로 표현하였다. 기존 온톨로지 간의 상호 운용성을 보장하여 자원 공유 및 재사용하기 위하여 온톨로지 간에 서로 의미에 따라 대응관계를 이루는 연관된 개념들을 찾아 연결하는 작업이 필요하다. 개념들이 이루는 대응관계는 equivalence, subsumption, disjointness 로 분류되며 관계를 찾는 과정을 온톨로지 매칭이라 하고 대응관계들이 이루는 집합을 온톨로지 정렬이라 한다. 매칭은 유사도 측정 방법에 기반을 둔다. 유사도 측정 방법은 두 온톨로지간의 개념의 같은 의미를 표현하는지를 판단한다. 기존 유사도 측정 방법은 문자열 기반, 어휘 기반, 구조 기반, 인스턴스 기반 등이 있다.

기존 온톨로지 정보의 이질성과 모호성으로 인하여 단일 유사도 방법으로 만족한 결과를 제공하지 못한다. 이 단점을 극복하기 위해 기존 온톨로지 정렬 시스템은 여러 가지 유사도 방법을 결합하여 서로 다른 정보 타입 (label, text, description, structure, rules)을 처리하고 각 항목의 유사도 값을 최적의 가중치 조합에 기반을 두어 계산하고 연결하여 결과의 정확성을 높였다. 하지만 기존 정렬시스템은 온톨로지 데이터 양에 따라 매트릭스를 생성하고 과도한 계산을 통해 처

이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단 차세대정보· 컴퓨팅 기술개발사업의 지원을 받아 수행된 연구임(No.2012M3C4A7033346).
† 교신저자

리하여 대용량 데이터에 대하여 공간적 및 계산적으로 부하를 발생하여 효율적이지 않다. 이를 해결하기 위하여 온톨로지 정렬을 휴리스틱 알고리즘을 적용하여 연구하였다. 기존 휴리스틱 정렬 알고리즘은 좋은 성능을 보이지만 조율해야 하는 파라미터가 많고 계산이 복잡하며 특정 도메인의 다양한 데이터에 따라 최적의 파라미터 조합을 적용해야 한다.

이 논문에서는 DCS 기반을 둔 온톨로지 정렬 알고리즘을 제안한다. 제안한 알고리즘은 조율하는 파라미터의 개수가 적고 Levy Flight 분포에 따른 탐색을 하기에 효율적이며 계산이 간단하다. 제안한 알고리즘에서 군집에 속한 개체를 후보정렬로 표현하고 목적함수 측정에 기반을 두어 개체들 사이의 효율적인 협력과 경쟁을 이루는 탐색을 통해 최적의 정렬을 얻고자 한다. 실험은 OAEI 에서 제공한 벤치마크 데이터를 사용하여 정확률과 재현율을 구하고 기존 휴리스틱 정렬 알고리즘과 비교 평가한다.

이 논문은 다음과 같이 구성된다. 2 장에서는 기존 휴리스틱 정렬 알고리즘에 대해서 설명한다. 3 장에서는 DCS 기반 온톨로지 정렬 알고리즘을 제안한다. 4 장에서는 제안 알고리즘을 평가하고 관련 연구들과 비교 하며 5 장에서는 결론과 향후 연구를 기술한다.

2. 관련연구

최근에 온톨로지 정렬을 휴리스틱 알고리즘을 적용하여 연구 진행하였다. 휴리스틱 알고리즘을 적용 함으로써 온톨로지 정렬 처리에 아래와 같은 장점들이 있다. 임의적인 후보정렬로 시작하고 처리하여 대량의 데이터에 대하여 과도한 계산을 피할 수 있다. 도메인 데이터에 알맞은 목적함수 및 유사도 측정 함수를 쉽게 수정하고 대체 가능할 수 있다. 개체마다 독립적인 계산을 지원하여 병렬처리와 개체들 사이의 협력과 경쟁을 통해 전역적인 탐색이 가능하다. 종료조건을 지정하여 원하는 시각에 종료할 수 있다.

Jurgen Bock[4]은 DPSO 기반 온톨로지 정렬에 대하여 제안하였다. DPSO 알고리즘은 최초에 최적의 속성 부분 집합 선택을 위한 최적화 전략으로 PSO 를 변경하여 제안하였으며 온톨로지 정렬의 대응관계 선택과 비슷한 작업으로 온톨로지 정렬에 응용하였다. DPSO 알고리즘은 정렬에 좋은 성능을 보였지만 조율해야 하는 파라미터가 많고 계산이 복잡하여 특정 도메인 다양한 데이터에 따라 최적의 파라미터 조합을 적용해야 한다.

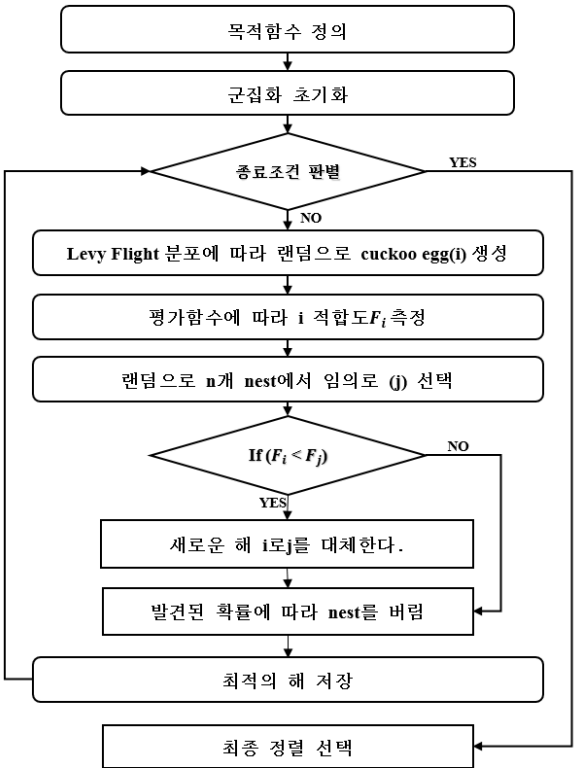
이 연구에서 제안하는 DCS 기반 알고리즘은 기존 heavy-tail 분포를 변화시킨 Levy Flight 이동과 뻐꾸기의 탁란 현상에 기반을 둔 CS 알고리즘을 알맞게 변경한 알고리즘이다[5]. Levy Flight 알고리즘 모델은 간단하고 조율해야 하는 파라미터가 적으며 먼저 좋은 해를 포함한 지역에 집중적으로 탐색하고 가끔 큰 스텝을 거쳐 검색되지 않은 지역을 탐색하기에 효율적인 강화(intensification)및 다양화(Diversification) 전략을 나타낸다. 실험 평가단계에서 DPSO 와 성능 비교한다.

3. DCS 기반 온톨로지 정렬 알고리즘

이 연구에서는 온톨로지 내의 클래스(class), 프로퍼티(property), 개체(individual) 등을 여러 종류의 개념(concept)으로 정의한다. 대응관계는 equivalence 를 지원하고 두 온톨로지 사이에 1:1 매칭을 지원한다. 두 개념 사이의 의미 유사도 측정은 개념정보에 기반을 두어 문자열 기반, 언어기반, 어휘기반 매칭 방법을 사용한다. 후보 온톨로지 정렬을 평가하기 위하여 목적함수를 정의하여 계산한다.

(그림 1)은 DCS 기반 온톨로지 정렬 흐름도를 보여주고 있다. 초기 단계에 목적 함수를 정의하고 군집화는 n 개의 둥지(nest)로 형성되었고 하나의 둥지에 하나의 알(egg)만 놓이며 각 알은 탐색공간 내에서 임의적으로 d 차원 벡터를 가지며 차원마다 두 온톨로지간의 무작위로 생성된 하나의 대응관계를 표시한다. 종료 조건을 만족하면 탐색과정을 종료하고 최종적으로 생성된 최적의 후보 정렬을 선택한다. 만족 되지 않으면 최적의 해를 기준으로 Levy Flight 분포에 따라 모방하여 뻐꾸기 알을 생성한다. 새로 생성된 알은 목적함수에 의해 평가되고 알의 적합도를 최적의 해와 비교하여 우수할 경우 최적의 해로 재설정한다. 알의 적합도가 기준치보다 낮을 때 버리고 새로 생성한다. 최적의 해를 찾고 위치와 적합도를 저장하고 다시 종료조건을 확인하여 반복한다.

그림(2)에서는 제안한 알고리즘에 대한 초기화 단계를 나타냈으며 그림(3)에서는 DCS 기반 온톨로지 정렬 알고리즘에 대한 전체적인 부분을 수도 코드로 보여준다.



(그림 1) DCS 기반 온톨로지 정렬 흐름도

Algorithm : Initialization of nests

01: *Input : N the number of nests*
02: $n = min(ccount(O_1), ccount(O_2))$
03: *For* $i = 1$ *to N do*
04: $d = rand(0, n)$
05: *For* $j = 1$ *to d do*
06: /* *randomly select classes (have not already been selected)* */
07: $m_{i,j} = CreateMatch(c_1, c_2)$ *where* $c_1 \in O_1, c_2 \in O_2$
08: *Evaluate similaryty* $f(m_{i,j})$
09: $X^*_i = X^*_i \cup m_{i,j}$
10: *End for*
11: *Order* $m_{i,j}$ *by the measurement of similarity in* X^*_i
12: *Evaluate Coordinate Alignment* $F(X^*_i)$
13: *Keep the best Nest*
14: *End For*

(그림 2) DCS 기반 온톨로지 정렬 초기화

Algorithm: DCS based Ontology Alignment

01: *objective function* $f(x), x = (x_1, ..., x_d)^T$
02: *// Generate Initialize population of n host Nests* $x_i (i = 1, ..., n)$)
03: *While* $(t < MaxGeneration)$ *or* $(Stop Criterion)$ *do*
04: *For* $i = 1$ *to n do*
$p_t \Leftarrow levyflight([0,1])$;
05: $l_{new} = p_t \otimes d_{bestnest}$;
06: $d_{new} \Leftarrow rand(l_{new}, n)$;
07: *For* $k = 1$ *to* l_{new} *do*
08: $X^*_i \Leftarrow X^*_i \cup X_{best,k}{}^*$;
09: *End for*
10: *For* $j = 1$ *to* $d_{new} - l_{new}$ *do*
11: /* *cuckoo start from their Nest to search new nest* */
12: $m_j = Get\ a\ new\ cuckoo\ randomly(j)$;
13: /* *objective function* */
14: *Compute* $Sim_j(c_1, c_2) \Leftarrow f(m_j)$;
15: $X^*_i \Leftarrow X^*_i \cup m_{new}$;
16: *End for*
17: /* *object function 2* */
18: *Build* $F(X^*_i)$ *// Evaluate* X^*_i *quility / fitness*;
19: *Abandon a fraction* (P_a) *of worse nests and built new ones*;
20: *if* $(F(X^*_i) < F(X^*_{best}))$
21: /* *Replace best by the new soluction* */
22: *Keep the best solutions or nests with quality solutions*;
23: *Rank the Alignment by the similarity*;
24: *end if*
25: *End for*
26: *End while*
27: *Post process results and visualization*;

(그림 3) DCS 기반 온톨로지 정렬 알고리즘

4. 실험 및 평가

제안 알고리즘의 성능을 알아보기 위하여 OAEI 2013 에서 제공하는 벤치마크 데이터를 사용하여 실험 및 평가하였다. 데이터는 하나의 온톨로지를 110 가지 형태로 만들어 제공한다.

100 시리즈는 온톨로지의 개념정보는 유지하고 구조를 수정한 형태이고 200 시리즈는 온톨로지의 구조정보는 유지하고 개념을 유사어나 외래어로 수정하였다. 300 시리즈는 온톨로지에서 실제로 사용되는 데이터에 가까운 형태로 제공한다.

온톨로지 정렬에서 시스템 성능 평가 척도로는 정확률과 재현율을 이용하며 수식은 아래와 같다.

$$Precision = \frac{\{Relevant\ Alignment\} \cap \{Rretried\ Alignment\}}{\{Retrieved\ Alignment\}}$$

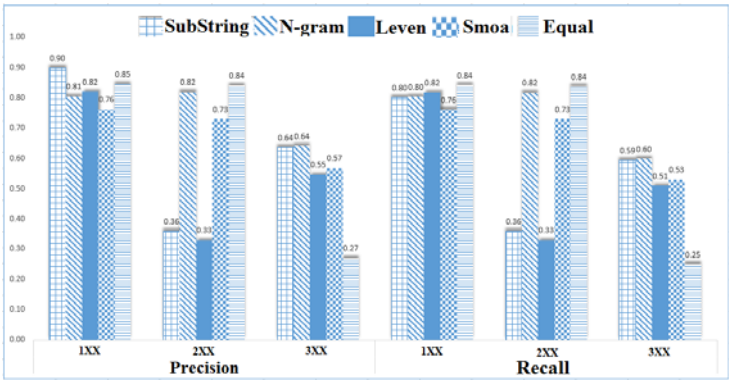
$$Recall = \frac{\{Relevant\ Alignment\} \cap \{Rretried\ Alignment\}}{\{Relevant\ Alignment\}}$$

정확률은 검색된 정렬의 총 항목 중에서 관련 정렬 항목에 일치하는 비율을 나타낸다. 재현율은 관련 정렬 총 항목 중에서 검색된 정렬의 항목이 일치하는 것을 비율로 나타낸다.

유사도 측정을 해결하기 위하여 문자열 매칭 방법 Levenshtein[6], N-gram[7], SMOA[8], SubString[9], EqualString[10]을 이용하여 실험하였다. 목적함수는 정렬을 이루는 대응관계들의 집합 크기와 정확도를 최대화 하고자 아래의 수식을 사용한다.

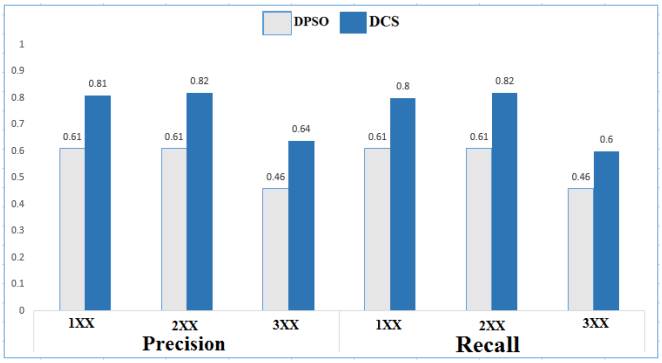
$$F(A_{opt}) = \alpha \otimes (\min(|O_1|, |O_2|) - |A_{opt}|) + (1-\alpha) \otimes \sum_{i=1}^{k} Sim_i(c_1, c_2) \quad (\alpha \in [0,1])$$

수식에서 $F(A_{opt})$ 는 후보정렬 A_{opt} 의 적합도를 나타내고 $Sim_i(c_1, c_2)$ 는 후보 정렬에서 i 번째 대응관계를 이루는 개념들의 유사도 값을 나타낸다. α 는 가중치로서 수치가 커질수록 대응관계 개수가 증가하고 반대로 작아지면 정확성이 높아진다. 실험에서는 파라미터 수치를 nest = 100, iteration = 100, p_a = 0.5, α = 0.6 으로 적용하였다.



(그림 4) 문자열 매칭 방법에 따른 성능

(그림 4)는 제안한 문자열 매칭 방법 중 온톨로지 정렬에 대한 성능을 비교하기 위한 실험 결과이다. 같은 데이터에 대하여 각 문자열 매칭 방법들의 나타내는 성능이 다르게 나타난다. 100 시리즈 데이터에 대해서는 비슷한 성능을 보이고 200 시리즈 데이터에 대해서는 N-gram, Smoa, EqualString 이 좋은 성능을 보이지만 300 시리즈에 대해서는 SubString 과 Levenshtein 이 좋은 성능을 보인다. 제안한 알고리즘은 도메인에 따라 다양한 데이터에 대하여 알맞은 매칭 방법을 적용하여 좋은 결과를 얻을 수 있다.



(그림 5) DCS 와 DPSO 정렬의 성능비교

(그림 5)는 제안한 DCS 알고리즘과 기존 DPSO 알고리즘을 구현하여 비교한 결과이다. 두 알고리즘은 같은 목적함수와 유사도 측정 함수(N-gram)를 적용하고 iteration 을 100 번 실행하였다. 데이터 100 시리즈, 200 시리즈, 300 시리즈에 대한 실험 결과에서 제안한 알고리즘은 DPSO 기반 알고리즘 보다 성능이 약 20% 향상 되었다. 제안한 알고리즘은 기존 DPSO 기반 알고리즘보다 좋은 성능을 보여준다.

5. 결론 및 향후 연구

이 연구에서는 휴리스틱 알고리즘 DCS 기반 온톨로지 정렬 알고리즘을 제안하였다. 알고리즘에서 개체는 하나의 후보정렬로서 Levy Flight 분포에 따른 탐색과 개체들 사이의 협력과 경쟁을 통해 최적의 정렬을 얻고자 한다. 제안된 알고리즘의 성능을 평가하기 위해 OAEI 에서 제공하는 벤치마크 데이터에 적용하고 기존 DPSO 기반 정렬 알고리즘과 비교하였다. 실험결과에서 제안된 알고리즘은 기존 DPSO 기반 알고리즘보다 좋은 성능을 보여준다.

향후 연구로 제안한 알고리즘을 확장하여 온톨로지 정렬 멀티 목적함수를 사용하고 자동적인 파라미터 배치를 적용하고자 한다.

참고문헌

[1] 이재호, “시맨틱 웹의 온톨로지 언어”, 정보과학회지, 제 21 권, 제 3 호, pp. 18-27, 2003.

[2] Gruber, "A Translation Approach to Portable Ontologies Specifications", Knowledge Acquisition, Vol.5, pp.199-220, 1993.

[3] Nezhadi, Shadgar, Soared, "Ontology alignment using machine learning techniques", International Journal of Computer Science & Information Technology, Vol.3, No.2, 2011.

[4] Jurgen Bock, Jan Hettenhausen, "Ontology Alignment using Discrete Particle Swarm Optimization", Information Sciences, Vol.192, pp.152-173, 2012.

[5] Xin-She Yang, Suash Deb, "Cuckoo Search via Levy Flight", Nature & Biologically Inspired Computing IEEE Publications, pp.210-214, 2009.

[6] Levenshtein, Vladimir I, "Binary codes capable of correcting deletions, insertions, and reversals", Soviet Physics Doklady, Vol.10, No.8, pp.845–848, 1965.

[7] Kondrak, "N-gram similarity and distance", String Processing and Information Retrieval, Vol.3772, pp.115-126, 2005.

[8] G. Stoilos, G. Stamou, S. Kollias, "A String Metric for Ontology Alignment", Springer-Verlag Berlin Heidelberg ISWC 2005, LNCS 3729, pp. 624–637, 2005.

[9] Baeza-Yates R, Navarro G, "A faster algorithm for approximate string matching", In Dan Hirchsberg, Gene Myers. *Combinatorial Pattern Matching (CPM'96), LNCS 1075*. Irvine, pp. 1–23, 1996.

[10] Navarro, Gonzalo, "A guided tour to approximate string matching", *ACM Computing Surveys,* Vol.33, No.1, pp.31–88, 2001.