정보관리연구, vol.42, no.3 2011, pp.167-186

동시인용정보를 이용한 동명이인 저자의 중의성 해소*

Disambiguation of Author Names Using Co-citation

강 인 수**
In-Su Kang

차 례

초 록

동시인용은 서로 다른 두 연구가 이후의 새로운 연구에서 동시 인용되는 것이다. 이 연구는 동시인용과 저자식별의 관계를 다룬다. 저자식별은 문헌에 출현한 동명의 저자명들을 실 세계 저자로 식별하는 것이다. 동시인용은, 한 사람의 관련된 연구들이 이후 또 다른 연구들에서 타인 혹은 자신에 의해 동시 인용되는 증거를 수집함으로써, 저자식별의 절차와 성능에 영향을 미칠 수 있다. 이 연구는 구글 스칼라로부터 동시인용을 자동 수집하는 절차를 제시하고 동시인용 정보를 저자식별의 기존 자질들과 효율적으로 결합하는 새로운 군집알고리즘을 제안한다. 실험을 통해 동시인용이 저자식별에 미치는 긍정적 효과를 확인하였다.

키 워 드

동시인용, 저자식별, 동시인용 정보 수집, 클러스터링, 저자식별 자질

* 이 논문은 2011학년도 경성대학교 학술연구비지원에 의하여 연구되었음.
** 경성대학교 컴퓨터학부 조교수
(Assistant Professor, Computer Science and Engineering, Kyungsung University, dbaisk@ks.ac.kr)
• 논문접수일자: 2011년 4월 4일
• 최종심사(수정)일자: 2011년 7월 14일
• 게재확정일자: 2011년 7월 14일

ABSTRACT

Co-citation means that two or more studies are cited together by a later study. This paper deals with the relationship between co-citation and author disambiguation. Author disambiguation is to cluster same-name author instances into real-world individuals. Co-citation may influence author disambiguation in terms that two or more related research works performed by the same person may be co-cited by some later studies. This article describes automated steps to gather co-citation information from Google scholar, and proposes a new clustering algorithm to effectively integrate co-citation information with other author disambiguation features. Experiments showed that co-citation helps to improve the performance of author disambiguation.

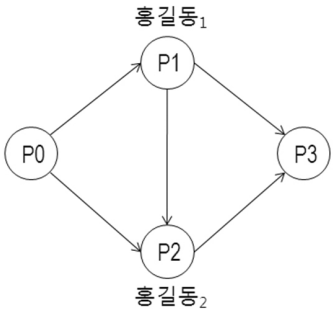
KEYWORDS

Co-citation, Author Disambiguation, Co-citation Gathering, Clustering, Author Disambiguation Features

1. 서 론

한 논문에서 다른 논문의 내용을 참조하는 인용은 적어도 이후 기술되는 세 가지 방식으로 저자식별에 영향을 미친다. 설명의 편의를 위해 〈그림 1〉과 같이 두 동명 저자명 개체 홍길동1, 홍길동2가 각각 작성한 두 논문 P1, P2에 대해 P1, P2가 각각 인용하는 논문 P3와, P1, P2를 동시에 인용하는 두 논문 P0가 존재하는 상황을 가정한다.

첫째 방식은 P1이 P2를 인용하는 것으로 "한 연구자는 자신의 이전 연구를 인용하는 경향이 있다"는 가정(McRae-Spencer and Shadbolt 2006)에 따라 홍길동1, 홍길동2를 한 사람의



〈그림 1〉 저자 식별과 인용의 형태(화살표는 인용을 표현함)

실 세계 저자로 고려할 수 있는 필요조건이다. 둘째 방식은 P1, P2 사이에 공유되는 인용 논문이 하나 이상 존재하는 상황이다. P3과 같은 공유인용논문은 홍길동1, 홍길동2가 작성한 논문 P1, P2가 다루는 특정 연구 주제와 관련된 선행 연구이다. 따라서 P3의 존재는, "한 연구자는 일정 기간 동안 유사한 특정 연구 주제를 다룬다"는 가정(Han, Giles, and Zha 2003)에 따라, 홍길동1, 홍길동2를 동일인으로 판단할 수 있는 필요조건이다. 셋째 방식은 P1, P2를 동시에 인용하는 논문이 하나 이상 존재하는 상황이다. P0와 같은 동시인용(co-citation) 논문의 존재는, "한 연구자의 유사 연구들은 이후 연구에서 동시 인용된다"는 가정에 따라, 홍길동1, 홍길동2를 동일인으로 묶을 수 있는 필요조건이다.

첫째 방식은 McRae-Spencer(2006)와 Pereira(2009)의 연구에서, 둘째 방식은 Song(2007)과 강인수(2008a)의 연구에서 직간접적으로 다루어졌다. 이 연구에서는 기존에 고려되지 않은 세 번째 방식인 동시인용이 저자 식별에 미치는 영향에 대해 살펴본다. 이를 위해 먼저 구글스칼라와 같은 웹 상의 학술문헌 사이트로부터 중의성 있는 동명 저자명 개체 쌍의 동시인용횟수를 수집하는 방법을 제시한다. 다음으로 최대군집우선 방식의 새로운 클러스터링 기법을 사용하여 공저자, 논문제목, 게재지명, 동시인용횟수 자질을 사용하는 저자 식별의 성능을 제시한다.

논문의 구성은 다음과 같다. 2장은 관련 연구를 기술한다. 3장에서는 웹 기반의 동시인용 횟수 수집에 대해 설명하고, 4장은 이 연구에서 시도될 저자명 클러스터링 알고리즘을 다룬다. 5장에서는 실험 절차와 결과를 기술하고 6장에서 결론을 맺는다.

2. 관련연구

저자식별은 사람 개입의 유무에 따라 수동, 자동으로 나뉜다. 정확한 저자식별을 위해서는 수동 방식이 사용되어야 한다. 그러나 식별 대상이 되는 전체 저자명 개체들의 수를 고려하면 현실적으로 자동 방식을 배제할 수 없다(Elliott 2010). 이 장에서는 저자식별의 자동화와 관련된 기존 연구를 기술한다.

자동 저자식별을 위한 초기 연구들은 Naïve Bayes, SVM, K-way Spectral Clustering 등의 기법(Han et al. 2004; 2005)을 기본 서지 항목(공동저자, 논문제목, 게재지 제목 등) 자질을 사용한 저자명 개체 표현의 군집화에 적용하였고 성능 향상을 위해 추가적 자질의 필요성을 지적하였다. 이후 전자메일주소, 소속, 참고문헌, 논문 첫 장, 전체 웹 혹은 웹 상의 개인출판논문리스트(Personal Publication List in Web, PPLW) 등의 자질들이 저자 식별에 시도되었다.

강인수(2008a; 2008b)는 기본 서지 항목과 함께 전자메일주소, 소속, 참고문헌 자질을 한글 저자명 식별에 사용하였다. PPLW는 저자

홈페이지의 하위 페이지나 별도 이력서 파일의 형태로 관리되며 한 저자가 작성한 출판논문의 정확한 리스트를 획득할 수 있는 주요 정보원으로 여러 연구(Aswani, Bontcheva, and Cunningham 2006; Tan, Kan, and Lee 2006; Pereira et al. 2009; Kang et al. 2010)에서 그 저자식별 유용성이 확인되었다. 전체 웹 공간 내에서 PPLW를 찾기 위해, 먼저 식별 대상 저자명이 출현한 논문제목을 검색 엔진의 질의로 던진 다음, 검색되는 URL들의 IHF (Inverse Host Frequency)를 계산하여 높은 IHF 값을 갖는 URL을 PPLW로 고려하는 방법이 주로 사용된다.

최고의 저자식별 성능(92.3~93.6%)을 보고한 방법(Song et al. 2007)에서는 먼저 중의성 있는 각 저자명에 대해 해당 저자명이 출현한 논문 원문의 첫 장과 참고문헌 텍스트로부터 PLSA 혹은 LDA 기법에 의해 저자명의 잠재 토픽 분포를 토픽 벡터의 형태로 학습한다. 이후 저자명 토픽 벡터들의 유클리디언 거리를 저자명 개체 간 거리로 계산한 다음 계층형 군집법을 통해 저자 군집을 생성한다. 그러나 이 방법은 식별 대상 저자명이 출현한 논문 원문 텍스트의 확보와 함께 잠재 토픽 학습에 사용될 토픽의 수를 미리 정해 두어야 하는 단점이 있다.

기존 저자식별 연구에서 인용 자질의 탐구는 자기인용(self-citation)과 인용논문공유[1] 의 두 가지 형태를 취하였다. McRae-Spencer (2006)는 "저자는 자신의 이전 연구를 인용하는 경향이 있다"는 가정에 기초하여 자기인용 링크로 연결된 저자명 개체들을 같은 저자 군집으로 병합하였다. Pereira(2009)는 웹을 통해 수집된, 논문의 참고문헌 리스트 페이지로부터 self-citation 관계를 얻어 저자식별에 활용하였다.

인용논문공유 측면에서 Song(2007)은 저자명 개체 표현을 위한 잠재 토픽 표현을 위해 논문 첫 장 텍스트와 함께 참고문헌 정보를 사용하였다. 강인수(2008a)는 8,675편의 한글 학술대회 논문에 출현한 23,177개 저자명 개체의 식별에 있어 공저자, 논문 제목, 게재지명, 소속, 전자메일, 인용논문공유 자질의 특성을 비교하였다. 이 중 인용논문공유 자질은 다른 자질들과 비교할 때 최소 과다군집오류와 최대 과소군집오류를 보여 저자식별력은 높으나 적용률이 낮은 특성을 나타냈다.

동시인용(co-citation)은 서로 다른 두 연구 A, B가 이후의 새로운 연구 C에서 함께 인용되는 것으로, A, B를 동시 인용하는 이후 연구들이 많을수록 A, B의 관련도가 높다고 알려져 있다(Small 1973). 동시인용 분석에서는 연구(분야)를 표현하기 위해 저자, 문헌, 저널 등을 사용한다. 이 중 저자(명)를 사용하여 연구(분야)를 표현하는 경우 저자동시인용(author co-citation)(White and Griffith 1981)이라고 부르며 이 때 저자는 그 저자가 작성한 연구 결과

1) 두 저자명 개체의 인용 논문 집합들의 공유 유무/정도.

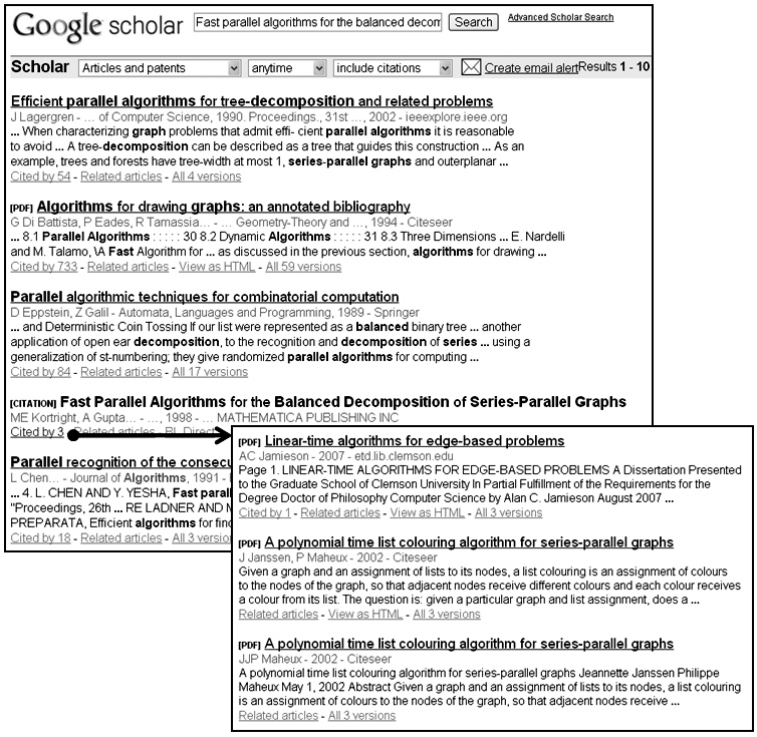
물 전체를 의미한다. 저자동시인용에 기초한 분석(author co-citation analysis, ACA)은 특정 학문 분야의 세부 분야 및 그에 대응하는 연구 커뮤니티의 사회적 구조 파악을 용이하게 하는 장점이 있다(Zhao 2006). ACA에서는 서로 다른 저자(명)들을 유사한 연구를 수행하는 저자 그룹들로 군집화하는 용도로 저자동시인용이 사용된다. 이와 달리 이 연구에서는 동명 저자(명)들의 실세계 저자를 식별하는 용도로 동시인용을 사용한다.

3. 동시인용 정보 수집 방법

인용 정보는 한 논문과 그 논문에서 참조하는 참고문헌 사이에 발생하므로 인용 정보 수집을 위해서는 논문 원문 텍스트의 참고문헌 리스트를 수동/자동으로 다룰 필요가 있다. 그러나 대용량 인용 정보의 자동 추출을 위해서는 인용 추출(Citation Extraction)과 인용 매칭(Citation Matching) (Pasula et al. 2002) 기술의 완성도는 차치하고라도 논문 원문의 확보가 전제되어야 한다. 다행히 최근 PubMed, Google scholar(구글스칼라)와 같이 인용 정보를 제공하는 웹 기반의 대용량 오픈 학술정보서비스들이 등장하였다. 이 연구에서는 이들 중 가장 광범위한 학술 문헌의 인용 정보를 제공하는 구글스칼라를 동시인용 정보의 수집원으로 사용한다.

구글스칼라는 〈그림 2〉와 같이 검색된 각 논문에 대해 해당 논문을 인용하는 다른 논문들의 리스트 페이지로 연결되는 "Cited By" 링크를 제공한다. 〈그림 2〉는 논문제목 "Fast Parallel Algorithms for the Balanced Decomposition of Series-Parallel Graphs"에 해당하는 논문을 찾기 위해 논문제목을 구글스칼라의 질의로 던지고 검색 결과의 네 번째 위치에서 해당 논문이 검색된 상황을 나타낸다. 또한 4-순위 검색 논문의 "Cited By 3" 링크로부터 해당 논문을 인용하는 세 편의 다른 논문의 확인이 가능함을 보인다.

Cited By 링크를 이용하여, 저자 식별의 대상이 되는 동명 저자명 개체 집합 $N=\{n_k\}$의 임의의 동명 저자명 개체쌍$\langle n_i,n_j\rangle$에 대해 동시인용횟수를 구축하는 단계별 절차는 다음과 같다. 먼저 N에 속한 각 저자명 개체 n_k는 n_k가 저자로 참여한 논문의 기본서지항목과 연결되어 있다고 가정한다. 단계-1에서는 n_k의 논문제목을 구글스칼라의 질의로 던져 〈그림 2〉의 좌측 상단과 같은 검색 논문 리스트 List1을 얻는다. 대부분의 경우 List1은 n_k의 논문을 최상위에 포함한다. 그러나 n_k의 논문제목 문자열의 오탈자, 수식/기호 표현 등으로 인해 n_k의 논문이 List1 내의 1-순위에 발견되지 않을 수 있다. 예를 들어 〈그림 2〉의 경우 검색된 논문이 List1의 4-순위에서 발견된다. 이러한 문제를 다루기 위해 단계-2에서는 n_k의 논문제목과 List1 내의 각 논문제목의 유사도를 계산하여 List1 내에서 최고 유사도 값을 갖는 논문 BMPaper(Best-Matching Paper)를 찾는다.



〈그림 2〉 구글스칼라 "Cited By" 링크

사용될 유사도 수식은 다음과 같다.

$$Sim(t_1, t_2) = \frac{|StemSet(t_1) \cap StemSet(t_2)|}{max(|StemSet(t_1)|, |StemSet(t_2)|)}$$

위 수식은 두 논문제목 t_1, t_2에 대해 길이가 긴 쪽의 단어 수를 기준으로 t_1, t_2 사이에 겹치는 단어 수의 비율을 계산하며 단어의 형태적 정규화를 위해 스테밍(Stemming)을 적용한다. StemSet(t)는 논문제목 t에 출현한 단어의 스템(stem)들의 집합이다. 위 수식은 재커드 계수(Jaccard coefficient)(Jaccard 1901)와 유사하지만, 두 논문제목 사이의 단어 공유 정도를 긴 논문제목 관점에서 계산함으로써 좀 더 직관적인 유사도 값을 만들며 유사도 임계치의 설정이 보다 용이해지는 장점을 갖는다. 예를 들어 n_k의 논문제목이 t_1={a,b,c,d}이고 검색된 논문제목이 t_2={b,c,d,e,f,g}라고 할 때, 재커드 계수값은 3/7이며 $Sim(t_1, t_2)$=3/6이 된다. 이 때 $Sim(t_1, t_2)$의 값 3/6으로부터 검색된 논문 제목의 50%가 n_k의 논문제목과 일치한다는 직관적 해석이 가능하다. t_1, t_2가 반대인 경우에도 마찬가지이다.

단계-3에서는 BMPaper의 Cited By 링크에 연결된(〈그림 2〉의 우측 하단과 같은) 논문 리스트 List2를 수집한다. List2의 각 논문에 부여할 고유식별자로, 아래 예와 같은 구글스

칼라 검색 논문의 Cited By 링크 URL로부터 속성 cites의 값(예: 9204292340298854520)을 추출하여 사용한다.

http://scholar.google.co.kr/scholar?cites=9204292340298854520&as_sdt=2005&sciodt=0,5&hl=en

이러한 식으로 List2에서 추출된 고유식별자들의 집합을 n_k를 인용하는 논문집합 Citing-Paper(n_k)에 대응시킨다. 마지막 단계-4에서는 N에 속한 임의의 동명 저자명 개체쌍 〈n_i, n_j〉에 대해 계산된 두 집합 CitingPaper(n_i), CitingPaper(n_j)의 교집합의 크기를 〈n_i,n_j〉의 동시인용횟수에 대응시킨다.

4. 최대군집우선 클러스터링

이 장은 저자식별을 위해 새롭게 제안하는 최대군집우선 클러스터링(Biggest First Clustering, BFC) 알고리즘(Algorithm-I)을 기술한다. 알고리즘은 중의성 있는 동명 저자명 개체들의 집합 V={n_k}와 임계치 이상의 유사도를 갖는 개체 쌍들의 집합 E={〈n_i,n_j〉}로 이루어진 그래프 G={V, E}를 입력으로 받아, V의 각 개체에 대해 저자식별자를 할당한 다음 〈개체, 저자식별자〉의 집합을 출력으로 만든다. Algorithm-I에서는 기술의 편의를 위해 〈개체, 저자식별자〉쌍의 집합이 {〈a,1〉, 〈b,2〉, 〈c,2〉}일 때 그 동등 표현으로 ClusterID(a)=1, ClusterID(b)=2, ClusterID(c)=2를 사용하였으며 InterClusterConnectivity()도 같은 방식의 표기를 사용하였다.

기본적으로 BFC 알고리즘은 주어진 그래프에서 군집화되지 않은 노드들 중 가장 큰 차수(node-degree)의 노드와 그 인접 노드들을 그래프에서 제거하여 새로운 군집으로 만들거나 기존 군집에 병합시키는 과정을 원 그래프가 공 그래프가 될 때까지 반복한다. G에 대해 처음으로 만들어지는 최대 차수 노드와 그 인접 노드들의 군집은 새로운 군집 newCluster로 생성되고 최초 군집 내 각 노드 n은 최초 0의 값을 갖는 GlobalClusterID를 부여받는다. 즉, ClusterID(n)=0. 이후부터는 군집ID를 부여받지 못한 노드들 중 최대 차수 노드와 그 인접 노드들로 새로운 군집 newCluster을 만든다. 다음으로 newCluster와 이전에 만들어진 각 군집들과의 연결도(군집 간 개체 링크의 수, InterClusterConnectivity)를 계산하여 가장 큰 연결도를 가지면서 newCluster와의 연결율이 임계치 λ이상(Line 16)인[2] 기존 군집이 존재할 경우 그 군집에 newCluster를 병합(Line 17)시킨다. 그러한 군집이 존재하지 않는 경우 newCluster를 새로운 군집으로 등록한다(Line 19).

2) $\frac{InterClusterConnectivity(maxClusterID)}{|newCluster|} \geq \lambda$

Algorithm-I BiggestFirstClustering

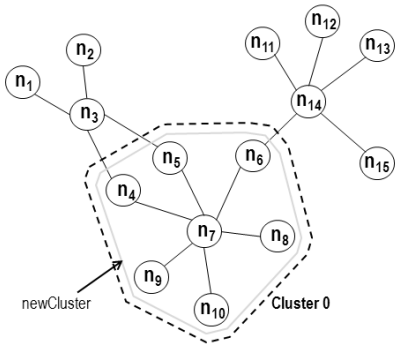
```
Input:
        Graph G=(V, E)
        λ: a cluster-connectivity threshold for merging clusters
Procedure:
 1      GlobalClusterID=0
 2      Let D be the list of nodes sorted by node-degree
 3      WHILE D is not empty
 4          Delete a next largest-degree node n from D
 5          IF ClusterID(n) does not exist          // a set of pairs of a node and its cluster identifier
 6              Empty InterClusterConnectivity()    // a set of pairs of a cluster identifier and its connectivity value
 7              Increase GlobalClusterID by 1
 8              newCluster={n}                      // a set of nodes
 9              FOR each adjacent node n_a of n
10                  IF ClusterID(n_a) exists
11                      Increase InterClusterConnectivity(ClusterID(n_a)) by 1
12                  END-IF
13                  newCluster=newCluster∪{ n_a}
14              END-FOR
15              Find a cluster id maxClusterID having the largest InterClusterConnectivity()
16              IF InterClusterConnectivity(maxClusterID) ≥ |newCluster|×λ
17                  ClusterID(x)=maxClusterID for each node x in newCluster
18              ELSE
19                  ClusterID(x)=GlobalClusterID for each node x in newCluster
20              END-IF
21          END-IF
22      END-WHILE
Output:
        ClusterID()             // a set of pairs of a node and its cluster identifier
```

〈그림 3〉은 15개 노드 집합 D={n_1, n_2, …, n_{15}}에 BFC 알고리즘을 적용한 이후 최초 세 번의 while-루프(Line 3 ~ Line 22) 동안의 주요 동작 과정을 보인 것이다. λ값은 0.4로 가정한다. 최초 while-루프에서 D로부터 차수가 가장 큰 노드 n_7을 제거하고, n_7과 n_7의 인접 노드들로 구성된 newCluster={n_4, n_5, n_6, n_7, n_8, n_9, n_{10}}를 만든다. 최초 군집의 경우 기존 군집이 존재하지 않아 interClusterConnectivity를 계산할 수 없으므로 newCluster에 속한 각 노드에 최초 0의 값을 갖는 GlobalClusterID를 부여한다. 즉 ClusterID(n_4)=0, ClusterID(n_5)=0, …, ClusterID(n_{10})=0. 〈그림 3〉의 (a)는 그 결과 얻어진 Cluster 0의 군집을 보이고 있다.

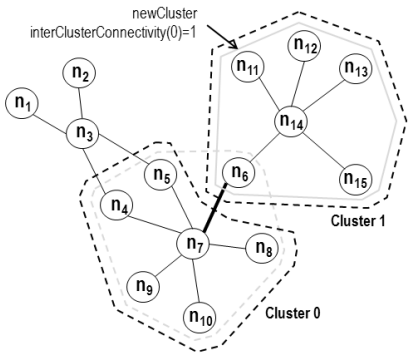
두 번째 while-루프에서 D로부터 가장 큰 차수의 노드 n_{14}를 제거하고, n_{14}과 n_{14}의 인접

노드들로 구성된 newCluster={n_6, n_{11}, n_{12}, n_{13}, n_{14}, n_{15}}를 만든다. 이후 newCluster와의 interClusterConnectivity가 가장 큰 기존 군집 Cluster 0={n_4, n_5, n_6, n_7, n_8, n_9, n_{10}}이 선택되지만, 조건 $1 \geq 6 \times \lambda$(알고리즘의 Line 16)이 거짓이므로 newCluster는 기존 군집과 병합되지 않고 군집 ID 1을 갖는 새로운 군집으로 만들어진다. 즉 ClusterID(n_6)=1, ClusterID(n_{11})=1, …, ClusterID(n_{15})=1. 〈그림 3〉의 (b)는 그 결과 얻어진 Cluster 1의 군집을 보이고 있다.

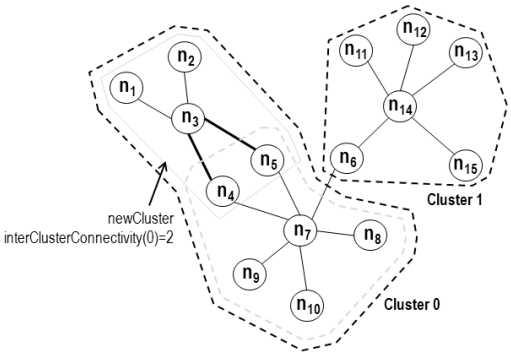
세 번째 while-루프에서 D로부터 가장 큰 차수의 노드 n_3을 제거하고, n_3과 n_3의 인접 노드들로 구성된 newCluster={n_1, n_2, n_3, n_4, n_5}를 만든다. 이후 newCluster와의 interClusterConnectivity가 가장 큰 기존 군집 Cluster 0={n_4, n_5, n_7, n_8, n_9, n_{10}}이 선택되고, 조건 $2 \geq 5 \times \lambda$(알고리즘의 Line 16)이 참이므로 newCluster는 기존 군집 0과 병합된다. 즉 ClusterID(n_1)=0, ClusterID(n_2)=0, …, ClusterID(n_5)=0. 〈그림 3〉의 (c)는 그 결과 얻어진 Cluster 0의 군집을 보이고 있다.



(a) 첫 번째 while 루프 수행 결과



(b) 두 번째 while 루프 수행 결과



(c) 세 번째 while 루프 수행 결과

〈그림 3〉 BFC 알고리즘 동작 예

5. 실 험

5.1 평가세트

동시인용자질이 저자식별에 미치는 영향을 정량적으로 평가하기 위해 펜실베니아주립대학의 CiteSeer 연구그룹이 구축한 PSU-CiteSeer-14 데이터세트를 사용하였다(Han, Zha, and Giles 2005). 이 데이터세트는 원 출처[3]에서 다운로드 되었으며 〈표 1〉에서 보는 바와 같이 14개 저자명이 출현한 총 8,453개 저자명 개체(서지레코드 형식)들을 총 479명의 실세계 저자들로 수작업 식별해 둔 것이다. 아쉽게도 서로 다른 기존 저자식별 연구들에서 상기 데이터세트의 일부 혹은 수정된 버전이 사용되어 객관적 성능 비교에 다소 어려움이 있다. 다행히 최근 Pereira(2009)의 연구에서는 같은 데이터세트에 대해 저자명 14개, 서지레코드 8,442개, 실세계 저자 480명의 거의 비슷한 통계를 보고하였다. 따라서 이 데이터세트의 사용을 통해 기존 연구와의 객관적 성능 비교가 가능하다.

저자식별의 다른 평가세트로는 KISTI(한국과학기술정보연구원) 연구그룹과 경성대학교에서 구축 보고한 최신의 저자식별 영어 데이터세트가 있다(강인수 et al. 2009; Kang et al. 2010). 그러나 이 데이터세트는 41,673개의 대용량 서지레코드로 구성되어 있어 이 데이터세트에 대해 동시인용 정보를 웹을 통해 수집하는 경우 상당한 시간의 경과가 요구되므로 현재 연구에서는 고려되지 못하였다. 다른 평가세트로 같은 KISTI 연구그룹에서 구축한

〈표 1〉 실험 평가세트 PSU-CiteSeer-14

Name	# of Name Instances	# of Real Authors
A Gupta	577	26
A Kumar	244	14
C Chen	801	61
D Johnson	368	15
J Lee	1,419	100
J Martin	112	16
J Robinson	171	12
J Smith	927	30
K Tanaka	280	10
M Brown	153	13
M Jones	260	13
M Miller	412	12
S Lee	1,464	86
Y Chen	1,265	71
Total	8,453	479

3) 〈http://clgiles.ist.psu.edu/data/nameset_author-disamb.tar.zip〉.

한글 저자식별 평가세트가 있다(강인수 et al. 2008a). 이 평가세트는 피인용 횟수가 낮은 학술대회 발표 논문들로 구성되어 있어 동시인용자질의 효용성을 보이는데 적절치 않다고 판단되어 본 실험에서 배제하였다.

5.2 동시인용횟수 수집

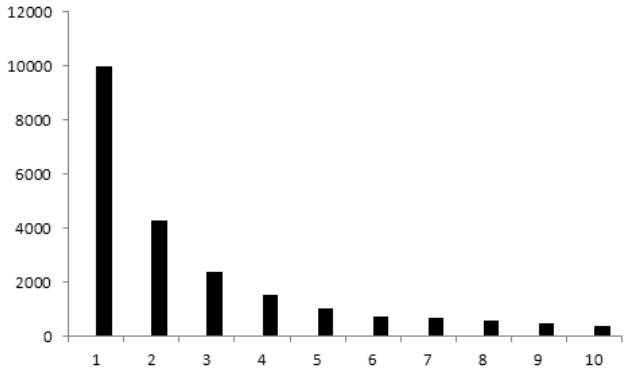
저자식별의 대상이 되는 8,453개 각 저자명에 대해, 관련된 서지레코드의 논문제목을 구글 스칼라의 질의로 사용하여 3장의 단계별 절차를 Perl 프로그램을 제작하여 자동 수행하였다. 단계-2에서 논문제목에 대한 StemSet 생성을 위해 제목 내 토큰 집합에서 불용어를 제거하고 개별 토큰에 포터스테밍(Porter stemming)을 적용하였다. 한편 BMPaper 판별을 위해 논문제목 유사도의 최저 임계치를 설정할 필요가 있다. 이를 위해 0.6 이상의 논문제목 유사도를 갖는 BMPaper 중 100개 샘플을 추출하여 선택된 BMPaper가 구글스칼라의 원 탐색 논문과 일치하는지를 수작업 검증하였다. 그 결과 [0.6, 0.7), [0.7, 0.8), [0.8, 0.9), [0.9, 1.0)의 각 유사도 구간에 해당하는 25씩의 BMPaper들의 92%, 96%, 100%, 100%가 구글스칼라 최초 탐색 논문과 일치하였다. 이후 90% 이상의 성능을 보이면서 보다 많은 저자명 개체쌍에 동시인용자질을 적용할 수 있다고 판단되는 0.6을 최저 임계치로 설정하고, 단계-3에서 0.6 이상의 유사도를 갖는 BMPaper에 대해 인용논문 식별자를 추출하였다. 그 결과 〈표 2〉와 같이 전체의 70%(5,938/8,453)에 해당하는 5,938개 저자명의 논문에 대해 하나 이상의 인용논문 식별자가 추출되었다.

단계-4에서는 인용논문 식별자를 갖는 5,938개 저자명 개체들의 임의 개체쌍으로부터 1회 이상의 동시인용횟수가 발생한 25,375개 저자명 개체쌍을 생성하였다. 이는 데이터세트 내 479개 실세계 저자에 해당하는 각 동명저자명 개체 집합 내에서 동시인용이 발생할 수 있는 최대 개체쌍의 총 수 269,156의 9.4%에 해당하며 이 비율은 동시인용자질의 적용률에 해당한다.

〈그림 4〉는 동시인용횟수에 따른 저자명 개체쌍의 수를 보인 것이다. 예상되는 바와 같이 동시인용횟수가 증가할수록 관련된 저자명 개체쌍의 수는 감소하며, 동시인용횟수가 10 이하인 개체쌍은 전체의 85.5%를 차지한다. 특이하게 최대 997까지의 큰 동시인용횟수를 갖는 소수의 개체쌍들이 발견되는데 이들은 중복 논문이거나 학술발표 이후 저널에 실린 동일/유

〈표 2〉 인용논문을 갖는 저자명 개체 수

제목유사도	[0.6, 0.7)	[0.7, 0.8)	[0.8, 0.9)	[0.9, 1.0)	1.0	Total
저자명 개체수	213 (3.6%)	368 (6.2%)	500 (8.4%)	25 (0.4%)	4,832 (81.4%)	5,938



〈그림 4〉 동시인용횟수 vs. 저자명 개체쌍 수

사제목 논문들의 저자명 개체쌍에 해당한다.

5.3 저자식별

저자식별을 위해 임의의 동명 저자명 개체쌍의 유사도를 계산하고 임계치 이상의 유사도를 갖는 개체쌍 사이에 링크를 설정한 그래프를 만든 다음 4장의 BFC-알고리즘을 적용하였다. 저자명 개체쌍 유사도는 공저자, 논문제목/게재지명, 동시인용자질들의 자질 유사도에 기초하여 계산된다. 공저자(Coauthor, Co) 자질 유사도는 두 저자명 개체 사이에 공유되는 공저자의 존재 여부에 따라 1, 0의 값을 부여하였다. 동시인용(Co-Citation, Ci) 자질 유사도는 두 저자명 개체의 논문들을 동시 인용하는 논문의 개수가 n개 이상인지 여부에 따라 1, 0의 값을 부여하였다. 논문제목(Title of the paper)과 게재지명(Title of the Publication)은 저자가 다루는 연구분야를 특정한다는 공통점이 있으므로 하나의 단위 TP(Title, Publication)로 통합하여 처리하였다. 먼저 각 저자명 개체에 대응하는 TP를 TF-IDF 가중치를 사용하는 용어 벡터로 표현해 두고, 임의의 두 TP의 코사인 유사도를 계산하여 TP 자질 유사도로 사용하였다. TP의 벡터 표현에 사용되는 용어들은 논문제목과 게재지명 내 토큰에 대해 불용어를 제거하고 포터스테밍을 적용한 것들이다.

전술한 세 자질들을 통합하는 저자명 개체쌍 유사도 수식은 다음과 같으며 Kang(2010)이 사용한 이진거리함수를 일반화한 것이다.

$$EntitySim(e_i, e_j) = \delta\left(\left(\sum_{k=1}^{K} \delta(FeatureSim_k(e_i, e_j) \geq \theta_k)\right) \geq \pi\right)$$

위 수식은 각각 K개 자질을 갖는 두 개체 e_i, e_j에 대해 임계치 θ_k 이상의 자질 유사도 $FeatureSim_k(e_i, e_j)$를 갖는 자질의 수가 π이상일 때 1의 유사도를 그렇지 않은 경우 0의 값을 부여한다. $\delta(p)$는 명제 p가 참이면 1, 거짓이면 0의 값을 갖도록 정의된다. Kang(2010)에

서는 위 수식의 π위치에 0 혹은 1의 값을 고정한 수식이 사용되었다. π의 값을 크게 설정할수록 많은 공유 자질을 갖는 상당히 유사한 개체쌍에 대해서만 링크를 설정하게 되므로 높은 정확률과 낮은 재현율의 저자식별 성능을 보일 것이다. 그 반대로 작은 π의 값을 사용하는 경우 낮은 정확률과 높은 재현율을 보일 것이다. 정확률과 재현율의 적절한 균형을 통제하는 파라미터가 자질 유사도 임계치 θ_k이다.

저자명 개체쌍의 유사도가 계산된 이후 1의 값을 갖는 개체 간에 링크를 설정하는 방식으로 그래프를 만들고 이후 BFC-알고리즘을 적용하여 저자 군집을 생성하였다. 〈표 3〉은 그 결과이다. Pereira(2009)는 본 연구의 평가세트와 거의 일치하는 데이터세트를 사용한 실험에서 랜덤하게 분할된 데이터세트의 절반을 학습에 나머지 절반을 평가에 사용하는 절차를 10회 반복하여 HAC, KWAY, SVM, WAD 각 기법의 저자식별 평균 성능을 보고하였다(〈표 5〉 참조). 본 연구에서도 Pereira의 방식을 따라 랜덤 샘플된 절반의 데이터세트를 통해 얻어진 최적 파라미터값을 사용하여, BFC-알고리즘을 나머지 절반의 데이터세트에 적용하는 절차를 10회 반복하여 평균 성능을 계산하였다. 파라미터들의 최적값을 얻기 위해 λ와 θ_{TP}의 경우 각각 {0.1, 0.2, …, 0.9}, θ_{Ci}의 경우 {1, 2, …, 10} 값들의 각 조합에 대해 학습 데이터세트에 BFC를 적용하고 최고 F1[4] 값을 갖는 파라미터 조합을 선택하였다.

〈표 3〉에서 베이스라인(Baseline)은 단일링크(Single linkage) 계층형 군집법에 해당한다. BFC는 베이스라인에 비해 높은 정확률을 보일 뿐 아니라 동시인용자질의 저자식별력을 다른 자질들과 효과적으로 결합하고 있음을 알 수 있다. 이는 BFC 군집화 과정에서 병합 조건으로 사용된 군집 연결도와 관련되어 있다. 군집 연결도를 군집 사이의 집단적 특성으로 본다면 계층형 군집법 중 군집의 집단적 특성을 사용하는 평균연결법(Average linkage), 완전연결법(Complete linkage) 등도 BFC와 대등

〈표 3〉 저자식별 성능 (Co: 공저자, TP: 논문제목/게재지명, Ci: 동시인용)

Method		Rec.	Pre.	F1	Parameters (π=1, θ_{Co}=1)
Baseline	Co	0.4330	0.1365	0.2075	
	Co+TP	0.4583	0.1348	0.2083	θ_{TP}=0.61
	Co+TP+Ci	0.5055	0.1378	0.2166	θ_{TP}=0.67, θ_{Ci}=6.9
BFC	Co	0.2310	0.5266	0.3211	λ=0.34
	Co+TP	0.2842	0.5028	0.3631	θ_{TP}=0.3, λ=0.3
	Co+TP+Ci	0.4200 (47.8%)	0.6003 (19.4%)	0.4942 (36.1%)	θ_{TP}=0.33, θ_{Ci}=1, λ=0.33

4) 저자식별 성능 지표로 일반적으로 pairwise F1이 사용된다(McCallum, Nigam, and Ungar 2000).

한 성능을 보일 것으로 예상할 수 있다. 그러나 완전연결법, 평균연결법에서는 집단적 특성으로 두 군집의 모든 개체쌍 거리의 최소값, 대푯값(centroid, average 등)을 사용하므로 본 연구에서의 군집 연결 정도와는 구별된다. BFC를 다른 군집법과 비교하는 주제는 이 연구의 범위를 벗어난다.

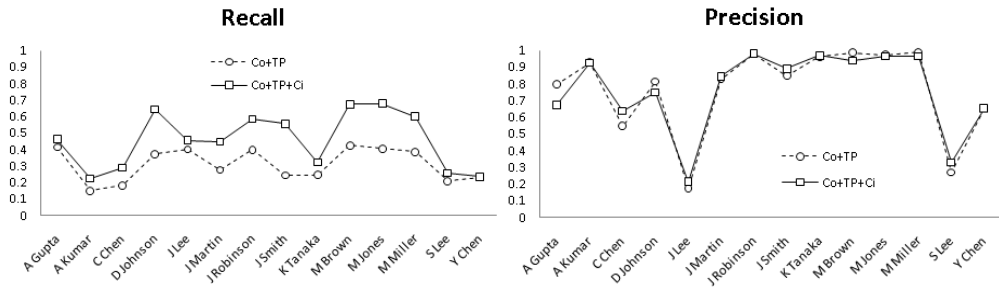
〈표 3〉에서 공저자 자질만을 사용한 BFC 저자식별의 F1 성능은 논문제목/게재지명, 동시인용자질들을 추가할 때 점증적으로 향상되었다. 특히 동시인용자질의 사용(Co+TP+Ci)은 기본 서지항목(Co+TP)에 기반한 저자식별 성능을 36.1% 향상시켰다. 〈표 4〉는 실험세트 내 14개 각 저자명의 저자 군집 성능을 보인 것으로 A Gupta와 Y Chen을 제외한 모든 저자명에 대해 동시인용자질은 거의 20% 이상의 성능 향상을 가져왔다. 이러한 결과는 서론에서 언급한 "한 연구자의 유사 연구들은 이후 연구에서 동시 인용된다"라는 가정을 지지하는 결과이다.

〈그림 5〉는 동시인용자질이 재현율, 정확률에 미치는 영향을 개별 저자명에 대해 도식화한 것이다. 그림은 동시인용자질의 추가로 재현율이 모든 저자명에 대해 향상됨을 보인다. 그러나 정확률은 모든 저자명에 대해 일관되게 향상되지 못하였다. 그럼에도 전체 14개 저자명의 평균[5] 정확률은 19.4% 상승되었는데(〈표 3〉 참조), 이는 개체 집단이 큰 일부 저자별 자질

〈표 4〉 14개 각 저자명에 대한 저자식별 성능과 동시인용자질의 오류율

Name	BFC (Co+TP+Ci)				Error rate (%)
	Rec.	Pre.	F1	Improv.	
A Gupta	0.4608	0.6724	0.5469	0.4%	12.05%
A Kumar	0.2223	0.9242	0.3584	40.2%	2.86%
C Chen	0.2889	0.6366	0.3975	46.1%	15.43%
D Johnson	0.6434	0.7492	0.6923	35.4%	7.45%
J Lee	0.4557	0.2118	0.2892	19.7%	14.15%
J Martin	0.4459	0.8439	0.5835	40.8%	1.14%
J Robinson	0.5836	0.9793	0.7314	29.5%	1.00%
J Smith	0.5563	0.8893	0.6845	81.8%	6.14%
K Tanaka	0.3210	0.9681	0.4821	23.3%	0.57%
M Brown	0.6737	0.9374	0.7839	32.3%	3.79%
M Jones	0.6789	0.9649	0.7970	39.4%	0.75%
M Miller	0.5992	0.9654	0.7395	33.7%	0.22%
S Lee	0.2569	0.3290	0.2885	22.6%	20.85%
Y Chen	0.2334	0.6519	0.3437	1.0%	10.89%
Total	0.4200	0.6003	0.4942	36.1%	9.53%

5) N(예: 14)개 동명 저자명 개체 집합에 대한 저자식별 성능(정확률, 재현율, F1)은 각 저자명 개체 집합 단위의 N개 성능들의 평균을 구하는 macro-averaging과 각 저자명 개체 집합 내에서 정의되는 임의의 개체쌍 단위의 성능을 구하는 micro-averaging 방식이 있으며 일반적으로 후자의 평균을 계산한다.



〈그림 5〉 동시인용자질로 인한 개별 저자명의 정확률, 재현율 차이

에서의 효용성을 보였다.

한편 〈그림 5〉에서 보인 것처럼 동시인용자질을 적용할 경우 총 14개 중 6개 저자명들의 정확률이 감소하였다. 정확률의 감소는 동시인용자질의 사용으로 인해 새롭게 설정된 저자명개체 간 링크에 오류가 있음을 의미한다. 즉 동명의 서로 다른 저자들의 논문들이 다른 논문들에서 동시 인용됨으로 인해 저자명 개체 사이에 부적절한 링크가 형성되었기 때문이다. 이러한 오류 링크의 비율을 계산해 보면, 실험에 사용된 구글스칼라 동시인용 정보가 적용되는 총 25,375개 저자명 개체쌍 중 9.53%(2,419개)에 해당한다. 즉 이들 2,419개 각 개체쌍은 평가세트의 정답 기준으로 실세계의 동명이인들이다. 〈표 4〉는 이러한 오류율을 저자명 기준으로 제시하고 있다. 예상되는 바와 같이 J Lee, S Lee, Y Chen과 같은 고빈도(〈표 1〉 참조) 저자명들은 동명이인의 연구 결과가 동시 인용될 가능성(〈표 4〉에서 Error rate) 또한 높았다.

본 실험에서 동시인용자질 적용으로 인한 오류가 발생한 사례들을 살펴보면 다음과 같다. 오류의 첫째 예로 아래의 1번 논문은 2번, 3번 논문을 동시 인용하고 있지만 2번, 3번 논문의 동명 저자명 "A. Gupta"에 해당하는 실세계 저자들은 서로 다른 사람이다. 이 경우 2번, 3번 논문에 출현한 "A. Gupta"들 간에 동시인용자질에 기인한 링크가 형성되지만 이는 오류 링크이므로 저자명 군집화 과정에서 잘못된 군집 결과를 만들게 되고 그 결과 정확률을 저하시킨다.

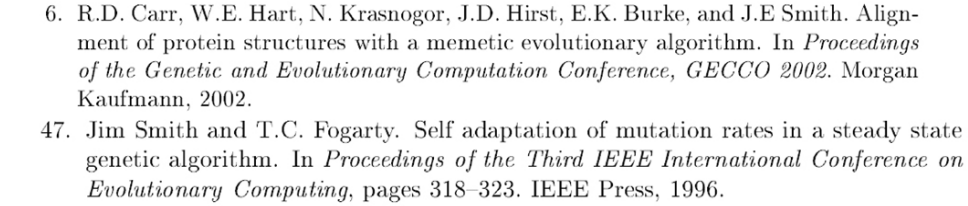
1. Sosonkina, M., Allison, D., Watson, L. (2002). Scalability analysis of parallel GMRES implementations. Parallel Algorithms and Applications, 17(4): 263-284.
2. **Gupta, A.**, Kumar, V. (1995). Performance and scalability of preconditioned conjugate methods on parallel computers. IEEE Trans. Parallel and Distrib. Systems, 6: 455-469.
3. Singh, J., Hennessy, J., **Gupta, A.** (1993). Scaling parallel programs for multiprocessors: methodology and examples. IEEE Computer, 7: 42-50.

오류의 다음 예도 이전의 경우와 마찬가지로 아래의 1번 논문에서 서로 다른 동명의 저자들이 각각 작성한 2번, 3번 논문을 동시 인용하는 경우이다. 아래 논문 세 편의 서지 정보에서 저자명은 약식 저자명 대신 완전 저자명을 복원하여 표기하였다. 실험 평가세트에 포함된 아래 2, 3번 논문의 서지 정보에는 약식 저자명 "J. Smith"가 들어 있다. 복원된 완전 저자명을 통해 알 수 있듯이 2, 3번 논문의 "J. Smith"는 다른 사람임이 확실하다. 그러나 아래 예의 경우도 1번 논문의 참고문헌 정보가 제공하는 동시인용정보에 기초한 저자식별 과정에서 2, 3번 논문의 "J. Smith"들은 동일 저자임을 표현하는 오류 링크가 설정되며 이는 결국 저자식별의 정확률을 저하시킨다.

아래의 "J. Smith"는 각각 "J.E Smith", "Jim Smith"으로 표기되어 있는데, 일반적으로 논문 작성시 참고문헌 형식을 통일시키는 점을 감안하면, "J.E Smith", "Jim Smith"는 다른 사람이라고 추정할 수 있다. 이처럼 참고문헌의 원본 항목에 접근할 경우 저자식별의 좋은 단서를 획득할 수 있다. 그러나 대용량 문헌집합에 대한 인용정보를 구축하는 경우 동일 인용에 대한 서로 다른 형식의 표기들을 정규화하는 과정에서 저자명 표기 정규화가 수행되어 원본 인용 항목에 기술된 저자식별의

1. Natalio Krasnogor, Steven Gustafson. (2003). Toward Truly "Memetic" Memetic Algorithms: Discussion and Proofs of Concept. Advances in Nature-Inspired Computation: The PPSN VII Workshops.
2. Robert D. Carr, William E. Hart, Natalio Krasnogor, Jonathan D. Hirst, Edmund K. Burke, James Smith (2002). Alignment of Protein Structures with a Memetic Evolutionary Algorithm. Proceedings of the Genetic and Evolutionary Computation Conference. pp.1027-1034.
3. Jim Smith, Terence C. Fogarty: (1996). Self-Adaptation of Mutation Rates in a Steady State Genetic Algorithm. International Conference on Evolutionary Computation. pp.318-323.

다음은 위 1번 논문의 참고문헌 리스트에서 위 2, 3번 논문 항목을 부분 캡쳐한 이미지들이다.



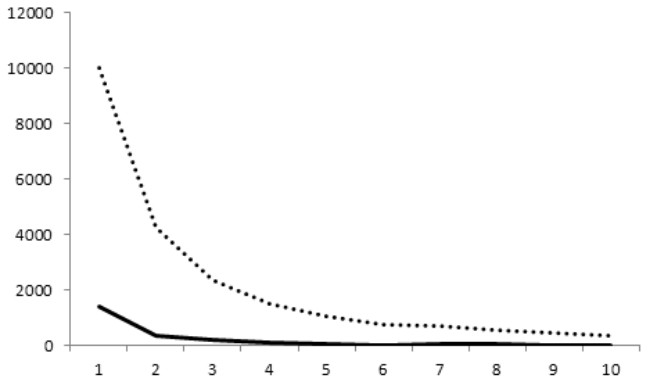
6. R.D. Carr, W.E. Hart, N. Krasnogor, J.D. Hirst, E.K. Burke, and J.E Smith. Alignment of protein structures with a memetic evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*. Morgan Kaufmann, 2002.
47. Jim Smith and T.C. Fogarty. Self adaptation of mutation rates in a steady state genetic algorithm. In *Proceedings of the Third IEEE International Conference on Evolutionary Computing*, pages 318–323. IEEE Press, 1996.

단서들이 손실될 수 있다. 위 "J. Smith"의 두 논문을 구글스칼라에서 검색한 경우에도 해당 논문의 저자명은 약식 표기로만 출력된다.

동시인용자질로 인한 오류의 문제를 다루기 위해 단순화된 약식 저자명 표기[6](Firstname initial과 lastname, 예: A. Gupta) 대신 보다 구체성이 큰 저자명 표기("Alok Gupta", "Aarti Gupta" 등)를 사용하여 동시인용 정보를 획득하는 것이 하나의 해법이 될 수 있다. 그러나 이를 위해서는 약식 저자명에 대한 완전 저자명을 복원하는 절차가 전제되어야 하므로 향후 이에 대한 연구가 요구된다.

〈그림 6〉은 본 실험에서 동시인용자질이 잘못 적용된 개체쌍의 개수 추이를 동시인용횟수 기준으로 표시한 것이며 실선, 점선은 각각 전체 개체쌍, 오류 개체쌍에 해당한다. 그림에서 알 수 있듯이 동시인용횟수가 증가할수록 동시인용자질 적용으로 인한 오류 개체쌍의 수는 급격히 감소한다. 실제 동시인용횟수가 5이상인 개체쌍에 대해서만 동시인용자질을 적용할 경우 약 85%의 오류가 제거되므로 정확률 향상이 가능하다. 그러나 이 경우 동시인용자질의 적용률은 70%이상 감소하므로 반대로 재현율이 급감하는 문제가 발생할 것이다. 따라서 학술정보서비스 구축을 위해 동시인용자질에 기초한 저자식별을 적용할 경우 동시인용횟수 임계치의 적정한 선택이 필요할 것이다.

마지막으로 〈표 5〉에서 본 연구의 결과와 기존 저자식별 방법들 HAC(Hierachical Agglomerative Clustering), KWAY(K-way Spectral Clustering), SVM(Support Vector Machines), WAD(Web Author Disambiguation)의 성능(Pereira et al. 2009)을 간접적으



〈그림 6〉 동시인용자질 오류 적용 개체쌍 개수 추이

(가로: 동시인용횟수, 세로: 개체쌍수, 점선: 전체 개체쌍, 실선: 오류 개체쌍)

6) 영문 저자명의 저자식별 연구에서는 서지레코드 표현의 보편성을 고려하여 firstname initial과 lastname이 결합된 포맷을 최초 저자명의 형태로 고려한다(Han et al. 2004).

〈표 5〉 저자식별 성능 비교 (Pereira et al. 2009)

Method	HAC	KWAY	SVM	WAD	BFC
F1	0.46	0.36	0.66	0.84	0.49
Features	PPLW	기본서지항목	기본서지항목	PPLW	Co+TP+Ci

로 비교하였다. HAC와 WAD는 개체쌍 유사도 계산을 위해 웹상의 개인출판논문리스트(Personal Publication List in Web, PPLW) 자질을 수집한 후 계층형 군집법을 적용한 것으로 두 방법의 성능 차이는 PPLW의 인식 정확률(WAD의 경우 90%)의 차이에 기인한 것으로 판단된다. 기본서지항목만을 사용한 SVM은 각 동명 저자명 개체 집합에 기록된 저자 클래스가 미리 알려져 있다고 가정하고 SVM으로 학습/분류를 수행한 것이다. 그러나 이러한 분류 기법의 사용은 본질적으로 군집문제에 해당하는 저자식별 작업을 다루는데 있어 대부분의 경우 부적절한 시도이다. 정리하면 동시인용자질을 사용한 BFC는 모든 저자명에 대한 수집 가능성을 보장하기 어려운 PPLW 자질을 사용한 경우를 제외한다면 군집 방식의 저자식별 방식에서 가장 우수한 성능을 보였다.

6. 결 론

이 연구는 학술문헌에 출현한 동명의 저자명들을 실세계 저자들로 군집화함에 있어 동시인용 정보의 효과를 다루었다. 동시인용 정보를 수집하기 위해 대용량 학술문헌집합에 대한 인용정보를 제공하는 구글스칼라를 배치모드로 검색하는 절차를 사용하였다. 수집된 동시인용자질과 기본서지항목 자질들에 기반한 동명 저자명의 군집화를 위해 새롭게 제안한 최대군집 우선클러스터링 기법을 적용하였다. 범용 영어 저자식별 평가세트를 사용한 실험에서 동시인용자질의 추가는 기본 서지항목에 기초한 저자식별 성능을 36.1% 향상시켰고, 동시인용 정보의 저자 식별력을 확인할 수 있었다. 이는 한 연구자의 관련된 연구들은 이후 연구들에서 동시 인용된다는 가정을 뒷받침하는 결과로 해석된다. 그러나 이 연구에서는 실용적 저자식별 기법을 개발하는 관점에서 동시인용과 다른 저자식별 자질들(저자명, 기관명 전거, 전자우편, 논문제목, 게재지명, 키워드 등)을 통합 적용하는 방법에 대한 탐구가 미비하였다.

한편 인용 관계로 연결된 두 문헌은 연구 주제에 있어 유사성, 연결성, 지속성 등을 가지므로 전술한 가정은 기존 저자식별 연구에서 고려되어 온 “한 연구자는 일정 기간 동안 관련된 주제의 연구를 수행한다”는 가정(Han, Giles, and Zha 2003)의 특수한 경우에 해당된다. 따라서 동시인용자질 역시 논문 제목, 게재지명 등의 연구 주제를 표현하는 저자식별 자질이 갖는 과다군집오류로부터 자유로울 수 없다.

즉 동일 혹은 관련된 주제를 다루는 동명의 서로 다른 저자들의 존재는 동시인용자질에 기반한 저자식별의 경우에도 악영향을 미칠 수 있다. 이와 관련하여 5장에서 논의한 바와 같이 기존의 약식 저자명 대신 완전 저자명 표현에 기초하여 동시인용 정보를 수집하거나 문헌의 원문으로부터 소속, 전자메일주소 등의 개인식별 정보를 추출하여 저자명 표현과 연결시키는 것은 전술한 문제를 해결하는 현실적 부분 해법이 될 것이다.

그러나 동시인용자질은 피인용이 발생하지 않은 문헌에 출현한 저자명의 식별에는 그 사용이 불가하다는 한계를 갖는다. 이와 관련하여 한 사람의 관련된 연구들이 이후 연구에서 동시인용될 확률은 동시인용자질의 저자식별 적용률과 상관 관계가 있을 것이다. 한편 자기인용(self-citation)은 타인 혹은 자신에 의한 이전 연구들의 동시인용에 비해 그 발생 빈도가 더 높을 것으로 예상되므로 동시인용자질의 낮은 적용률을 보완하는 자질이 될 수 있다. 전술한 내용들은 향후 연구에서 다룰 예정이다.

참고문헌

강인수, 이승우, 정한민, 김평. 2008a. 저자 식별을 위한 자질 비교.『한국콘텐츠학회논문지』, 8(2): 41-47.

강인수. 2008b. 저자 식별을 위한 전자메일의 추출 및 활용.『한국콘텐츠학회논문지』, 8(6): 261-268.

강인수, 김평, 이승우, 정한민. 2009. 저자 식별을 위한 대용량 평가셋 구축.『한국콘텐츠학회논문지』, 9(11): 455-464.

Aswani, N., K. Bontcheva, and H. Cunningham. 2006. "Mining Information for Instance Unification." *Proceedings of the 5th International Semantic Web Conference (ISWC)*, 329-342.

Elliott, S. 2010. "Survey of author name disambiguation: 2004 to 2010." *Library Philosophy and Practice*. [cited 2011. 6. 17]. 〈http://digitalcommons.unl.edu/libphilprac/473/〉.

Han, H., C. Giles, and H. Zha. 2003. "A Model-based K-means Algorithm for name Disambiguation." *Proceedings of Semantic Web Technologies for Searching and Retrieving Scientific Data*, Oct. 20, Florida: USA.

Han, H., C. Giles, H. Zha, and C. Li. 2004. "Two Supervised Learning Approaches for Name Disambiguation in Author Citations." *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries(JCDL)*, 296-305.

Han, H., H. Zha, and C. Giles. 2005. "Name Disambiguation in Author Citations Using a K-way Spectral Clustering method."

Proceedings of the ACM/IEEE Joint Conference on Digital Libraries(JCDL), 334-343.

Jaccard, P. 1901. "Etude Comparative de la Distribution Florale Dans une Portion Des Alpes et des Jura." *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37: 547-579.

Kang, I., P. Kim, S. Lee, and H. Jung. 2010. "Construction of a Large-scale Test Set for Author Disambiguation." *Information Processing & Management*, 47(3): 452-465.

McCallum, A., K. Nigam, and L. Ungar. 2000. "Efficient Clustering of High-dimensional Data Sets with Application to Reference Matching." *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD)*, 169-178.

McRae-Spencer, D. and N. Shadbolt. 2006. "Also by the Same Author: AKTiveAuthor, a Citation Graph Approach to Name Disambiguation." *Proceedings of ACM/IEEE Joint Conference on Digital Libraries(JCDL)*, 53-54.

Pasula, H., B. Marthi, B. Milch, and S. Russell. 2002. "Identity Uncertainty and Citation Matching." *NIPS*, 1401-1408.

Pereira, D., B. Ribeiro-Neto, N. Ziviani, and A. Laender. 2009. "Using Web Information for Author Name Disambiguation." *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries(JCDL)*, 49-58.

Small, H. 1973. "Co-citation in the Scientific Literature: A New Measure of the Relationship Between Two Documents." *Journal of the American Society for Information Science*, 24(4): 265-269.

Song, Y., J. Huang, I. Councill, and J. Li. 2007. "Efficient Topic-based Unsupervised name Disambiguation." *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries(JCDL)*, 342-351.

Tan, Y., M. Kan, and D. Lee. 2006. "Search Engine Driven Author Disambiguation." *Proceedings of ACM/IEEE Joint Conference on Digital Libraries(JCDL)*, 314-315.

White, H. and B. Griffith. 1981. "Author Cocitation: A Literature Measure of Intellectual Structure." *Journal of the American Society for Information Science*, 32(3): 163-171.

Zhao, D. 2006. "Towards All-author Co-citation Analysis." *Information Processing & Management*, 42: 1578-1591.