사회망 기반 순환 탐지 기법을 이용한 저자명 명확화 기법

(Name Disambiguation using Cycle Detection Algorithm Based on Social Networks)

신 동 욱 † 김 태 환 † 정 하 나 † 최 중 민 ††
(Dongwook Shin) (Taehwan Kim) (Hana Jeong) (Joongmin Choi)

요 약 이름은 사람을 구별하기 위한 특징이지만 여러 사람이 하나의 이름을 공유하는 경우와 한 사람이 여러 이름을 사용하는 경우 때문에 이름만으로는 사람을 명확히 구별할 수 없다. 이러한 문제는 정보 검색 분야에서 문서 검색이나 웹 검색, 데이터베이스 통합 등에 영향을 미친다. 특히 서지 정보에는 저자들 중 동명이인이 존재하거나 한 저자가 축약된 이름 혹은 잘못된 철자를 사용하기도 하기 때문에 에러 정보가 많이 포함되어 있다. 이러한 문제를 해결하기 위해 데이터베이스에 입력된 자료 중 이름에 대한 정보를 명확하게 해야 한다. 본 논문에서는 저자간의 관계로부터 구축된 사회망을 이용해 이름의 모호성을 해결하는 방법을 제안하고 컴퓨터 과학 서지정보를 제공하는 DBLP(Digital Bibliography & Library Project) 데이터를 기반한 실험을 통해 제안한 시스템의 성능의 효율성을 평가하였다.

키워드 : 사회망, 이름 모호성 해결, 식별 불확실성, DBLP, 순환 탐지

Abstract A name is a key feature for distinguishing people, but we often fail to discriminate people because an author may have multiple names or multiple authors may share the same name. Such name ambiguity problems affect the performance of document retrieval, web search and database integration. Especially, in bibliography information, a number of errors may be included since there are different authors with the same name or an author name may be misspelled or represented with an abbreviation. For solving these problems, it is necessary to disambiguate the names inputted into the database. In this paper, we propose a method to solve the name ambiguity by using social networks constructed based on the relations between authors. We evaluated the effectiveness of the proposed system based on DBLP data that offer computer science bibliographic information.

Key words : Social Networks, Name Disambiguation, Identity Uncertainty, DBLP, Cycle Detection

1. 서 론

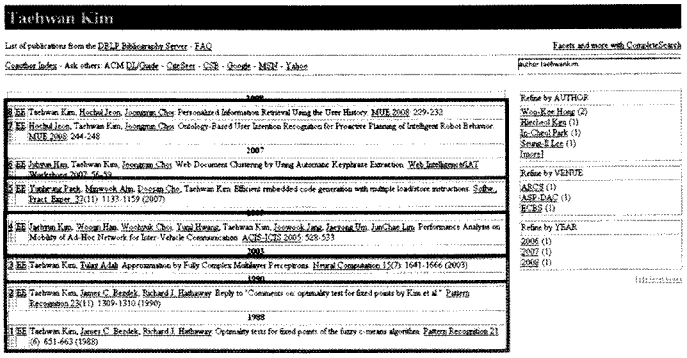
이름의 모호성(name ambiguity)은 식별 불확실성(identity uncertainty) 문제의 특별한 경우로, 객체들이 유일한 식별자로 표시되지 않는 경우이다. 많은 연구자들이 여러 분야에서 식별 불확실성 문제를 해결하기 위한 다양한 방법을 활발히 연구 중이다.

이름은 사람을 나타내는 가장 기본적인 특징이지만 개인을 나타내는 유일한 식별자가 될 수 없고, 이로 인해 데이터의 모호성이 존재하게 된다. 간단한 예로, Carnegie Mellon University의 교수인 Tom Mitchell에 대한 정보를 검색하기 위해 'Tom Mitchell'의 질의를 입력하면 CMU 교수 외에도 연기자 'Thomas Mitchell', 재즈 기타 연주가 'Tom Mitchell', 미식축구 선수 'Tom Mitchell' 등 5명의 동명이인에 대한 정보가 검색 결과의 상위에 존재한다.

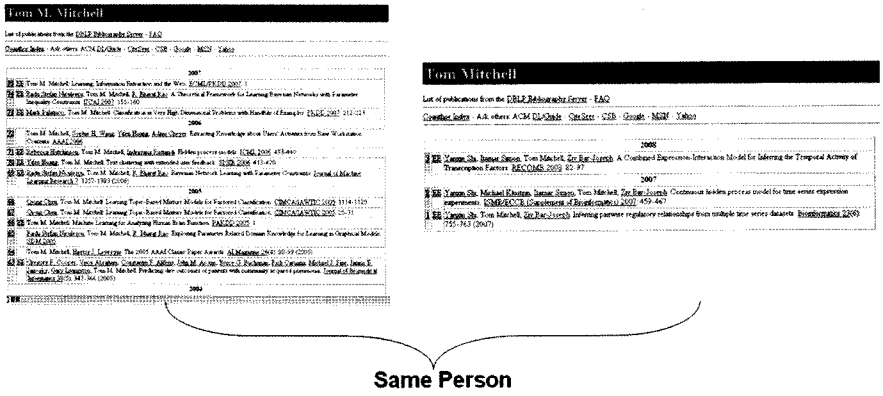
이름의 모호성은 문서 검색이나 웹 검색의 성능에 영향을 미치고, 데이터베이스 통합 시 개체 무결성을 보장

† 학생회원 : 한양대학교 컴퓨터공학과
foremostdw@gmail.com
kimth@islab.hanyang.ac.kr
rarara00@gmail.com
†† 종신회원 : 한양대학교 컴퓨터공학과 교수
jmchoi@hanyang.ac.kr
논문접수 : 2008년 10월 13일
심사완료 : 2009년 2월 6일

정보과학회논문지: 소프트웨어 및 응용 제36권 제4호(2009.4)



(a)



(b)

그림 1 이름의 모호성의 예

하지 못한다. 전자도서관의 경우 이름의 모호성으로 인해 출판물의 자료 수집 시 수집되는 자료의 질에 영향을 미치거나 서지학 자료의 검색 시 성능을 감소시킨다.

그림 1은 컴퓨터 과학 서지정보를 제공하는 DBLP 웹 사이트에 실제로 존재하는 이름의 모호성에 관련된 예이다. 그림 1의 (a)의 경우 'Taehwan Kim'이라는 이름을 DBLP 웹 사이트에 질의한 결과로 한양대, 서울대, 서강대, MITRE Corporation, Glynn Scientific Inc.에서 연구원으로 활동 중인 동명이인 5명에 대한 정보가 함께 제공되는 것을 확인할 수 있고, 그림 1의 (b)의 경우, 'Tom Mitchell'과 'Tom M. Mitchell'은 동일한 사람이지만 서로 다른 사람으로 판단하여 각 저자명에 해당하는 일부의 정보만 제공하는 것을 확인 할 수 있다.

서지 정보의 경우, 논문의 제목, 논문의 분야, 저자의 이메일 정보, 소속 정보, 공동 저자 정보 등 이름의 모호성을 해결하기 위한 특징들을 포함하고 있다. 특히 공동 저자 정보를 이용하여 각 논문의 저자 정보를 통해 저자들 간의 관계를 알 수 있다. 논문의 공동 저자들은 서로 친밀한 관계를 가지고, 이러한 저자들 간의 관계는 동명이인이나 한 저자가 여러 이름을 사용하는 경우를 탐지하는데 유용한 정보가 된다. 또한 연구자들은 자신이 관심 있는 특정 분야에 대한 연구를 진행하기 때문에 한 저자가 작성한 논문들을 대표할 수 있는 중요 단어들은 저자가 관심 있는 특정 분야를 대표하는 단어일 가능성이 높다. 이러한 서지 정보의 특징을 바탕으로 구축된 사회망(Social Networks)은 저자명의 모호성을 해결하기에 적합하다.

본 논문에서는 이러한 사회망을 이용하여 저자명의 모호성을 해결하기 위한 방법을 제안한다. 웹 상에서 주요 컴퓨터 과학 학회의 학회 발표용 논문(proceeding)과 학술지(journal)의 서지정보(bibliographic information)를 제공하는 DBLP(Digital Bibliography & Library Project)[1] 자료를 이용하여 저자의 이름과 논문의 제목, 출판사 정보 등을 추출한다. 또한 추가적인 정보수집을 위해 논문의 제목을 웹 검색 엔진에 질의하여 추출된 정보를 이용하여 사회망을 구축한 후, 이를 기반으로 저자들간의 관계와 저자의 후보 주제어 비교를 통해 저자명의 모호성을 해결하고자 한다.

1) http://www.Informatik.Uni-Trier.DE/~ley/db/index.html

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안한 시스템에 관한 관련 연구를 분석하고, 3장에서는 전체 시스템 구조에 대하여 기술한다. 4장에서는 사회망을 구축하는 일련의 과정에 대하여 언급한다. 5장에서는 시스템이 동명이인을 탐지하는 방법에 대하여 기술하고, 6장에서는 여러 이름을 사용하는 저자를 탐지하는 방법에 대하여 설명한다. 7장과 8장에서 시스템에 대한 평가와 결론을 내린다.

2. 관련 연구

여러 분야에서 불확실성 문제를 해결하기 위해 활발한 연구가 진행되었다. 데이터베이스 분야에서는 불확실성 문제를 해결하기 위해 레코드 연계(Record linkage) [1], 중복 레코드 탐지(Duplicate record detection)[2], 중복 정리(Merge/Purge)[3], 데이터 연관성(data association)[4], 데이터베이스 경화(database hardening)[5] 등의 연구가 진행되었고, 자연어처리 분야에서도 불확실성 문제를 해결하기 위해 Cross-Document Coreferences [6], name matching[7] 등의 방법이 연구되었다. 이 밖에도 인공지능 분야에서 군집화(clustering)[8], 확률 모델(probabilistic model)[9], 그래프(graph theory)[10] 등을 이용하여 불확실성을 해결하기 위한 연구가 진행되었고, 온톨로지를 통해 불확실성을 해결하려는 시도[11]도 있었다.

본 논문에서 제안하는 시스템은 이러한 연구 중 군집화, 확률 모델, 그래프를 적절하게 결합한 방법으로 군집화, 확률 모델, 그래프를 이용한 연구와 유사하다.

Ying Chen[8]은 비지도 학습(unsupervised learning)을 통해 뉴스 문서 내 이름의 모호성 해결을 위한 방법을 제안하였다. 문서에서 문법적, 의미적 특징을 추출한 후, 이 특징들을 이용한 군집화를 통해 이름의 모호성을 해결하고자 하였으나 문서 내 단어의 유사도를 통해 군집화할 뿐 사람간의 관계는 고려하지 않았다. 제안한 시스템에서는 사람간의 관계를 고려하며 유사한 특징을 가진 정점들간의 군집화 효과를 위해 최장 순환 탐지 알고리즘을 이용하여 발견된 최장 순환 내의 정점들간의 유사성이 존재한다고 판단하여 이름의 모호성 해결을 위한 유사도 평가시 최장 순환 내의 정점들을 하나의 가상 군집(cluster)으로 활용한다.

Hui Han[9]은 이름의 모호성을 해결하기 위해 베이시안 확률 모델과 SVM(Support Vector Machine)의 두 가지 지도학습(supervised learning) 방법을 제안하였다. 확률 모델을 사용하여 저자의 이름과 논문의 용어들간 유사도 측정을 통한 모호성 해결 방법을 제안하였으나, 역시 사람들간의 관계성을 고려하지 않았다. 제안한 시스템에서는 구축된 사회망을 기반으로 탐지된 최장 순환 내 정점들의 후보주제어들간 유사도 평가를 통해 사람들간의 관계성도 함께 고려한다.

Osmar R. Zaiane[10]은 DBLP 자료를 이용하여 사회망을 구축한 후, 정점들간의 관계성을 찾아내기 위해 random walk 알고리즘을 사용하여 저자들의 정보를 기반으로 한 정점의 순열을 찾아낸 후, 정점의 순열을 기반으로 저자들간의 커뮤니티를 구축하였다. 사회망을 구축한 후, 저자들간의 관계성을 고려하여 이름의 모호성을 해결하고자 시도한 점에서 제안한 시스템과 유사하지만 저자들간의 유사성을 판단할 때, 논문 제목 내 용어만을 이용하여 유사도를 판단하기 때문에 정확도가 떨어지는 문제가 발생하였다. 이름의 모호성을 해결하기 위한 많은 연구가 활발하게 진행되었지만 기존의 연구는 사람간의 관계를 고려하지 않고, 이름의 유사성과 용어의 유사성만을 이용하거나 사람간의 관계를 이용하더라도 사람을 나타내는 특징인 이름의 모호성을 해결하기에 부족하거나 완벽하게 구축된 자료 집합을 이용하여 구축하여 실세계에 적용하기에는 적합하지 못하였다.

본 논문에서는 사회망을 구축하고 이를 분석함으로써 저자명의 모호성을 해결하는 시스템을 제안한다.

3. 시스템 구조

본 논문에서 제안하는 사회망 분석(SNA, Social Network Analysis)을 이용한 저자명 모호성 해결 시스템의 전체 시스템 구조는 그림 2와 같다.

Information Extractor는 사회망을 구축하기 전 전처리 과정에 관련된 부분으로 DBLP Data Extractor가 DBLP 자료 집합에서 필요한 정보를 추출한 후, Web Search Engine Extractor에 논문의 제목을 제공하여 논문의 요약문과 저자의 소속 정보를 추출한다. 그 후 Noun Phrase Extractor가 논문의 요약문에서 논문을 대표하는 명사구를 추출한다. 정보 추출 시 특정 자료에 과적합(overfitting)되지 않은 시스템을 구축하기 위해 완벽한 자료 집합이 아닌 불순도를 포함하는 실세계의 자료 집합을 추출한다.

Social Networks Constructor는 추출된 정보를 기반으로 저자간의 관계를 고려하여 사회망을 구축한다.

Namesake Detector는 동명이인을 탐지하는 모듈이다. 먼저 Cycle Detector에서 저자 정보를 포함하는 각 정점에서 이분되는 최장 순환을 탐지하고, Similarity Measurer에서 최장 순환에 포함되는 정점들의 중요 용어들 간 유사도 비교를 통해 동명이인을 탐지한 후, Namesake Splitter를 통해 동명이인의 정보들이 결합되어 있다고 판단되는 정점의 정보를 분할한다.

Multiple Name Detector는 한 저자가 여러 이름을 사용하는 경우, 여러 정점으로 분할된 저자의 정점을 결합하기 위한 모듈이다. Similar Name Searcher가 각 정점

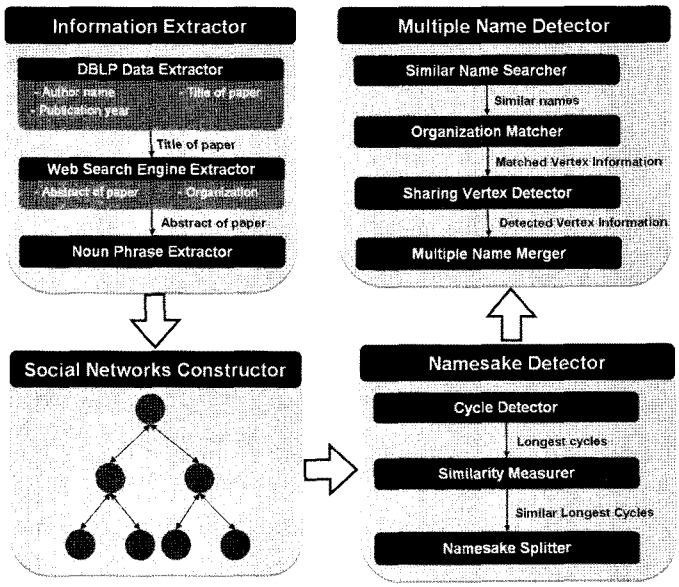


그림 2 시스템 구조

과 유사한 저자명을 가지는 정점을 탐지한 후, 두 가지 경우를 통해 같은 사람인지 판단한다. 첫 번째 경우는 탐지된 정점들의 소속이 같고 정점들의 중요 용어들 간 유사도가 실험을 통해 정해진 기준치 보다 높은 경우 Organization Matcher를 통해 같은 사람으로 판단한다. 두 번째 경우, 유사한 저자명을 가진다고 판단된 정점들이 같은 정점을 공유하는지 여부를 Sharing Vertex Detector를 통해 검사한 후, 같은 정점을 공유하는 경우 같은 사람으로 판단한다. 같은 사람으로 판단된 정점들의 정보는 Multiple Name Merger를 통해 결합된다.

4. 서지정보를 이용한 사회망 구축

사회망을 구축하기 위해 DBLP 자료와 웹을 통해 정보를 수집한 후, 수집된 정보를 기반으로 저자를 정점으로 하는 사회망을 구축한다.

4.1 정보 추출

자료의 추출은 DBLP 자료를 대상으로 하였다. DBLP 자료 중 inproceeding에 해당하는 논문에서 저자명, 논문 제목, 출판연도 등의 정보를 추출한 후 추가적인 정보수집을 위해 추출된 논문의 제목을 웹 검색 엔진에 질의하여 논문의 요약문 정보와 저자의 소속 정보를 추출하였다. 이때 웹 상의 정보의 추출은 표 1에 해당하는 사이트에 대해서 이루어졌다.

4.2 후보 주제어 추출

논문을 대표할 수 있는 후보 주제어를 추출하기 위해 각 논문의 요약문을 기반으로 형태소분석을 통해 요약문 내의 명사구를 추출하였다[12].

표 1 정보 추출 사이트 목록

사이트	URL
Portal ACM	http://portal.acm.org
IEEE	http://ieeexplore.ieee.org http://doi.ieeecomputersociety.org
Sciencedirect	http://sciencedirect.com
Society for Imaging Science and Technology	http://imaging .org
Springer	http://www.springer.com
웹 상의 PDF file	

논문의 요약문은 논문의 전체적인 내용을 간략하게 나타낸 것으로 특히 요약문 내의 명사구들은 문장을 대표하는 주어나 목적이 되는 목적어들을 포함하기 때문에 논문을 대표하는 후보 주제어일 가능성이 높다. 명사구가 아닌 명사만 고려할 경우에는 추출되는 후보 주제어들이 일반적인 단어로 의미상 논문의 주제를 나타내기에 모호하거나 의미 없는 단어일 가능성이 높다. 예를 들어 'machine learning'은 기계학습이라는 의미로 논문을 대표할 수 있는 후보주제어지만 'machine'과 'learning' 각각은 기계, 학습으로 특정 분야의 논문을 대표하기에는 모호하다.

논문의 요약문에서 주제어로 적합하다고 판단되는 명사구를 추출한 후 수식어와 같은 불필요한 문장 구성요소를 제거하여 후보 주제어를 추출한다.

그림 3은 요약문에서 형태소 분석을 통해 요약문 내 후보 주제어를 추출하는 과정을 보이는 예이다. 이때 형태소 분석은 Stanford Parser[2]를 이용하였다. 후보 주

제어는 명사구와 발생빈도(frequency)의 쌍으로 추출한다. 형태소분석은 각 문장단위로 행해지며, 하나의 문장을 S라고 하였을 때, 그림 3의 파서(Parser) 부분과 같이 S를 최상위 부모(root)로 하는 트리구조가 형성된다. 이 중 명사구 NP에 해당하는 정보만 추출된다. NP 중 PRP(Personal pronoun, 인칭 대명사), DT(Determiner, 한정사), RB(Adverb, 부사), CD(Coordinating conjunction, 등위 접속사), JJR(Adjective, Comparative, 형용사, 비교급) 등과 같이 수식어나 대명사 혹은 비교급 형용사 등 불필요한 문장 구성요소를 제거하여 후보 주제어를 추출한다.

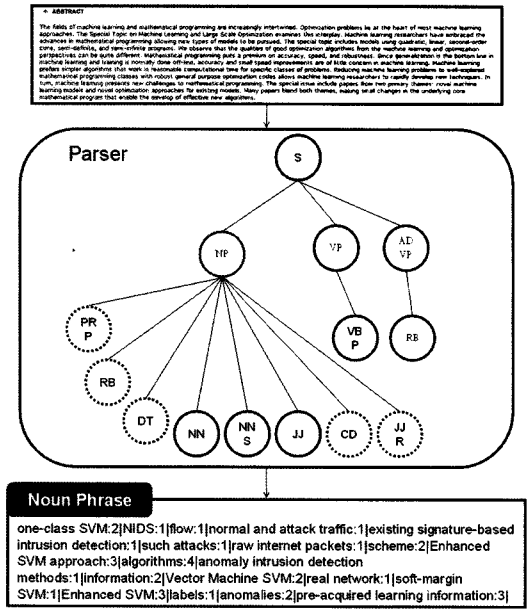


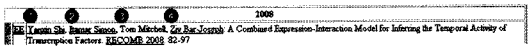
그림 3 후보 주제어 추출

4.3 사회망 구축

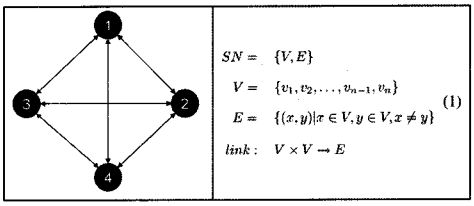
사회망은 식 (1)과 같이 정의할 수 있다. 사회망은 $SN = \{V, E\}$로 정의된다. 이때 V는 모든 정점들의 집합이고, E는 연결선, 즉 정점의 쌍의 집합이다. $link$는 임의의 정점의 쌍이 존재할 때 이 정점에 연결선을 연결하는 의미이다.

그림 4와 같이 추출된 저자명과 소속 정보를 기반으로 논문의 각 저자들을 정점으로 생성한 후, 공동 저자들을 연결선으로 연결하여 사회망을 구축한다.

그림 4의 (a)를 기반으로 구축되는 사회망은 그림 4의 (b)와 같다.



(a)



(b)

그림 4 사회망 구축

5. 동명이인 탐지

이름의 모호성에는 크게 두 가지 경우가 존재한다. 첫 번째는 여러 저자가 하나의 이름을 공유하는 경우이고 두 번째는 한 저자가 여러 이름을 사용하는 경우이다. 이 절에서는 첫 번째 경우에 해당하는 이름의 모호성을 해결한다.

5.1 순환 탐지

사회망은 정점들간의 관계를 연결선으로 표현한 것이다. 정점간의 연결선이 존재하지 않는다면 정점간의 관계도 존재하지 않는다고 가정할 수 있다. 특히 서지정보의 경우 한 정점이 여러 저자의 정보를 포함하고 있을 가능성이 높다. 그러므로 특정 정점을 기준으로 이분되는 순환이 존재한다면 동명이인인 여러 저자의 정보가 결합되어 있을 가능성이 높다.

이 논문에서는 Johnson의 cycle enumeration algorithm[13]을 기반으로 한 순환 탐지 알고리즘을 통해 이러한 순환을 탐지하였다. 그 후, 탐지된 순환 중 이분되는 최장 순환을 추출한다. 이분되는 최장 순환 탐지 알고리즘은 알고리즘 1과 같다.

알고리즘 1에서 C는 순환 탐지 알고리즘에서 탐지된 순환들의 집합이다. setDescendingOrder() 함수는 순환의 길이를 내림차순으로 정렬해주는 함수이다. getFirstElement()와 getNextElement()는 각각 배열 내에 첫 번째, 다음 요소를 가져오는 함수이다.

알고리즘 1의 4-9라인은 순환 간의 포함관계를 고려하여 특정 순환의 부분 순환인 순환들을 제거하기 위한 내용으로, 순환 탐지 알고리즘을 통해 탐지된 순환들 중 가장 긴 길이를 가지는 순환을 기준으로 하여 각 순환들이 가장 긴 길이를 가지는 순환에 속하는지 여부를 판단한 후, 가장 긴 길이를 가지는 순환에 속하지 않는 순환들을 탐지한다. 위에서 언급한 각 순환이 최장 순환에 속하는지 여부는, 예를 들어 특정 순환 c_1을 구성하는 정점의 집합이 최장 순환 c_2를 구성하는 정점에 속할 때 순환 c_1은 순환 c_2에 속한다고 판단한다. 그 후,

2) http://nlp.stanford.edu/software/lex-parser.shtml

Algorithm 1 Detect Longest Cycles

```
1:  procedure VOID DETECTLONGESTCYCLES(C)                    ▷ C : Set of Cycles
2:      cycleList ← setDescendingOrder(C)
3:      targetCycle ← getFirstElement(cycleList)
4:      while cycleList is not Empty do
5:          compCycle ← getNextElement(cycleList)
6:          if compCycle ⊆ targetCycle then
7:              longestCycles.add(compCycle)
8:          end if
9:      end while
10:     while longestCycles is not Empty do
11:         cycle ← getNextElement(longestCycles)
12:         find mc ∈ longestCycle such that cycle ∩ mc ≠ ∅
13:         while such mc exists do
14:             lc ← cycle ∪ mc
15:             longestCycles.remove(mc)
16:             longestCycles.remove(cycle)
17:             longestCycles.add(lc)
18:         end while
19:     end while
20: end procedure
```

알고리즘 1 최장 순환 탐지 알고리즘

10-19 라인에서 탐지된 순환들 중 같은 정점을 공유하는 순환들을 결합하기 위해 각 순환간 비교를 통해 같은 정점을 공유하는 순환을 탐지하여 결합한다.

5.2 유사도 측정

시스템은 이분된 최장 순환 간 유사도 측정을 통해 동명이인을 탐지한다. 각 최장 순환에 속하는 정점들의 후보 주제어들의 단어 빈도를 통해 가중치를 부여한다. 그 후, 텍스트의 유사성 판단을 위해 코사인 측정 방식을 이용하여 유사도 측정을 한다.

$$sim(c_1, c_2) = \frac{c_1 \bullet c_2}{\|c_1\| * \|c_2\|} \quad (2)$$

c_1 : 순환1 내 모든 정점들의 후보주제어 벡터

c_2 : 순환2 내 모든 정점들의 후보주제어 벡터

sim 함수는 비교대상이 되는 각 순환의 순환 내 정점들의 후보주제어 벡터를 입력 받아 결과로 순환간의 유사도 w를 반환하는 함수이다. 순환 내 정점들의 후보주제어 벡터는 4.2절에서 추출된 후보주제어를 이용하여 순환 내 정점들의 후보주제어를 벡터로 구성한다. 이때 후보주제어 벡터는 순환 내 정점들이 포함하는 후보주제어들의 용어 빈도수(Term Frequency)를 이용하여 가중치를 부여한 후 벡터 형태로 표현한다. 그 후 코사인 측정 방식을 이용하여 순환 내 정점들의 후보 주제어 벡터간 유사도 측정 후 유사도 w에 대한 평가를 한다. 본 논문에서는 유사도 w가 기준치 0.4보다 작은 경우 동명이인으로 판단하여 분할한다. 이 때, 기준치는 실험을 통해 결정하였다(기준치 결정을 위한 실험 데이터는 7절에서 기술하였다).

5.3 동명이인 분할

여러 저자의 정보를 포함하고 있다고 판단되는 정점에 대해서는 각 저자의 정보를 분할하게 된다.

저자의 정보가 분할되는 예는 그림 5와 같다.

구축된 사회망을 이용하여 1번 정점을 기준으로 이분된 최장 순환을 찾는다. 최장 순환을 찾은 후 그 순환에 속하는 정점들의 후보 주제어들을 모두 합친 후, 순환간의 후보 주제어의 유사도를 평가한다. 유사도가 기준치보다 높은 경우는 같은 사람으로 판단하여 분할이 발생하지 않는다.

하지만 유사도가 기준치보다 작은 경우, 정점이 동명이인의 정보를 포함하고 있다고 판단하여 이분된 순환에 따라 정점을 분할한다.

6. 저자의 다른 이름 탐지

한 저자가 여러 이름을 사용하는 경우에 대한 모호성을 제거하기 위해 유사한 이름을 검색한다. 검색된 유사한 이름 중 소속이 같고 유사도가 기준치 이상이거나 같은 저자와의 연결성이 존재할 경우 같은 사람으로 판단한다.

6.1 유사 이름 검색

특정 정점을 기준으로 유사한 저자명을 가진 정점을 탐지하기 위해 LCS(Longest Common Subsequence) 방법을 사용한다.

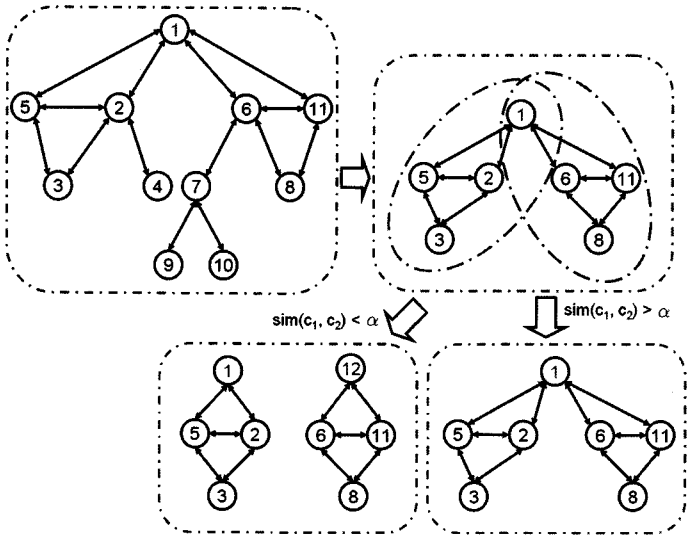


그림 5 동명이인 분할 과정

저자명을 공백이나 마침표를 통해 토큰화한 후, LCS를 이용해 유사도를 측정한다. 식 (3)과 같은 방법으로 측정한 유사도가 실험을 통해 정한 기준치 0.8보다 높은 경우 유사한 이름으로 판단한다.

$$nameSim(s_1, s_2) = \frac{\sum_{i \in s_1} maxLCS(s_{1i})}{maxLength(s_1, s_2)} \quad (3)$$

nameSim(s_1, s_2) 함수는 저자명의 유사성을 판단하기 위한 함수이고, *maxLCS*(s_{1i}) 함수는 s_1 저자명의 각 토큰을 s_2 저자명의 각 토큰과 비교한 최대 LCS를 반환하는 함수이다. *maxLength*(s_1, s_2) 함수는 유사도를 정규화(normalization) 하기 위해 비교 대상이 되는 두 저자명 s_1, s_2 중 긴 문자열의 길이를 반환하는 함수이다.

유사 이름을 검색하는 예는 그림 6과 같다. 이 예에서는 'Tom Mitchell'과 'Tom M. Mitchell'이 동일 인물인지를 판단하려고 한다.

s_1 = "Tom Mitchell", s_2 = "Tom M. Mitcehll"로 하고 각 저자명을 공백이나 마침표를 통해 토큰화한 후

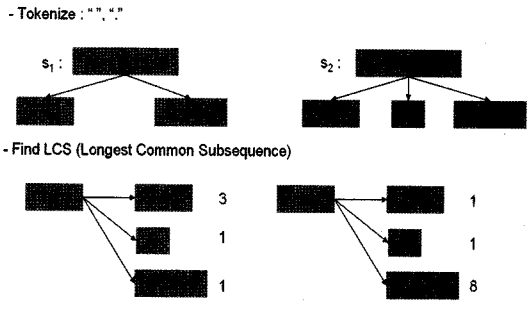


그림 6 유사 이름 검색

LCS 방식을 이용하여 각 토큰 사이의 유사도를 계산한다. 위의 예에서 s_1은 "Tom"과 "Mitchell"의 2개 토큰, s_2는 "Tom", "M", "Mitchell"의 3개 토큰으로 분리되므로 총 6개의 토큰간 유사도를 구할 수 있다. 이때 가장 높은 유사도를 보이는 값을 모두 더한 후, 최장 문자열을 가지는 저자명으로 나누어 정규화(normalization)를 한 후 기준치와의 비교를 통해 유사 이름을 판단한다. 위의 예에서는 (3+8)/12=0.917의 값이 나와서 기준치를 0.8로 했을 경우 기준치를 초과하므로 동일 인물이라고 판단한다.

6.2 소속 비교

소속은 저자명과 유사하게 저자를 식별 가능한 유일한 식별자는 아니지만 저자를 대표하는 특징이다. 저자명이 유사한 저자가 같은 소속을 가질 경우, 식 (2)를 통한 저자의 후보주제어들 간 유사도 평가 후 유사도가 기준치 0.1보다 높을 경우 같은 저자로 판단하여 두 정점을 결합한다. 유사도 평가는 각 저자의 후보주제어뿐만 아니라, 저자의 공동저자의 후보주제어도 함께 고려하여 평가한다.

이 때 기준치는 실험을 통해 결정하였다(기준치 결정을 위한 실험 데이터는 7절에서 기술하였다).

6.3 저자 공유 탐지

유사한 저자명을 가진 정점들이 같은 정점을 공유하는지 판단한다. 같은 정점을 공유한다는 것은 정점간의 연결성이 존재한다는 것을 뜻하므로 두 정점은 같은 저자일 가능성이 높다. 유사한 저자명을 가진 정점이 같은 정점을 공유할 경우 같은 저자로 판단하여 두 정점을 결합한다.

6.4 다른 이름을 가진 같은 저자 정보 결합

유사한 저자명을 가진 저자들 중 같은 사람으로 판단된 경우, 저자의 정보를 포함한 정점들을 결합한다.

저자의 정보를 결합하는 예는 그림 7과 같다.

유사 이름 검색 단계에서 유사한 저자명을 가진다고 판단된 정점들에 대하여 두 가지 경우를 통해 같은 사람인지 판단한다.

첫 번째 경우, 그림 7의 (a)와 같이 1번 정점의 저자명과 8번 정점의 저자명이 유사하다고 판단된 경우 소속 정보가 일치하는지 비교를 한 후, 소속 정보가 일치할 경우 저자의 후보주제어와 저자와 연결된 공동 저자들의 후보주제어를 기반으로 유사도 평가를 한 후, 유사도가 기준치 이상일 때 같은 사람으로 판단하여 결합한다.

혹은 그림 7의 (b)의 1번 정점과 11번 정점이 7번 정점을 함께 공유하는 것처럼 유사한 저자명을 가진 정점이 같은 정점을 공유할 경우 같은 사람이라고 판단하여 결합한다.

7. 실험 평가

7.1 시나리오

시스템의 실험을 위해 구축한 사용자 인터페이스는 그림 8과 같다. 모델을 구축하는 과정에서 각 단계의 정보를 포함하여 최종 결과뿐만 아니라 중간 단계 결과 정보도 함께 제공한다.

(a) 검색화면에 저자명을 입력하게 되면, 수집된 DBLP 자료에서 해당하는 정보를 검색하여 (b) 검색 결과를 보여준다. 그 후 저자명의 모호성을 해결하기 위한 각 단계를 진행하며 단계별로 진행과정이 보여지게 된다. 그림 8에 보이는 (c), (d), (e), (f)가 각 단계에 해당하는 그래프 정보이다. (c)는 수집된 DBLP 자료로 구축된 그래프로 입력된 저자명에 해당하는 하나의 정점(빨강색 정점)이 존재하는 것을 확인할 수 있다. 동명이인 탐지 단계를 거친 (d) 그래프는 입력 정점의 분할로 인해 입력된 저자명에 해당하는 3개의 정점(빨강색 정점)이 존재한다. 소속 비교 단계를 거친 (e) 그래프는 합쳐지는 정점이 없이 그대로 3개의 정점이 존재하고, 저자 공유 탐지를 거친 (f) 그래프는 같은 저자 정보를 공유하는 정점 간 결합이 발생한다. 모든 단계를 거쳐 시스템은 사용자에게 저자명의 모호성을 해결한 최종 결과 (g) 를 보여준다. 위의 예에서는 입력한 저자명에 해당하는 두 명의 저자가 존재한다고 판단한다.

7.2 성능 평가

7.2.1 기존의 평가 방식을 이용한 성능 평가

본 논문에서는 동명이인 탐지 성능의 기준치를 결정하기 위해 전체 데이터 중 약 1000개의 문서와 관련된 약 2000개의 정점 중 무작위 추출(Random Sampling) 방법[14]을 적용하여 임의로 선택한 10명의 저자에 해당하는 약 100개의 순환에 대하여 실험하였다.

실험은 식 (4)에 해당하는 정확률(Precision)과 재현률(Recall), F-지수(F-measure)를 이용하였다.

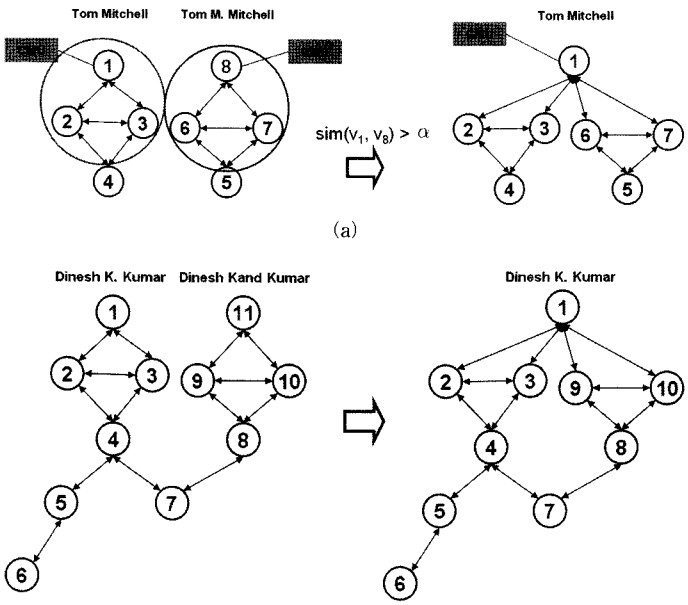


그림 7 같은 저자 정보 결합

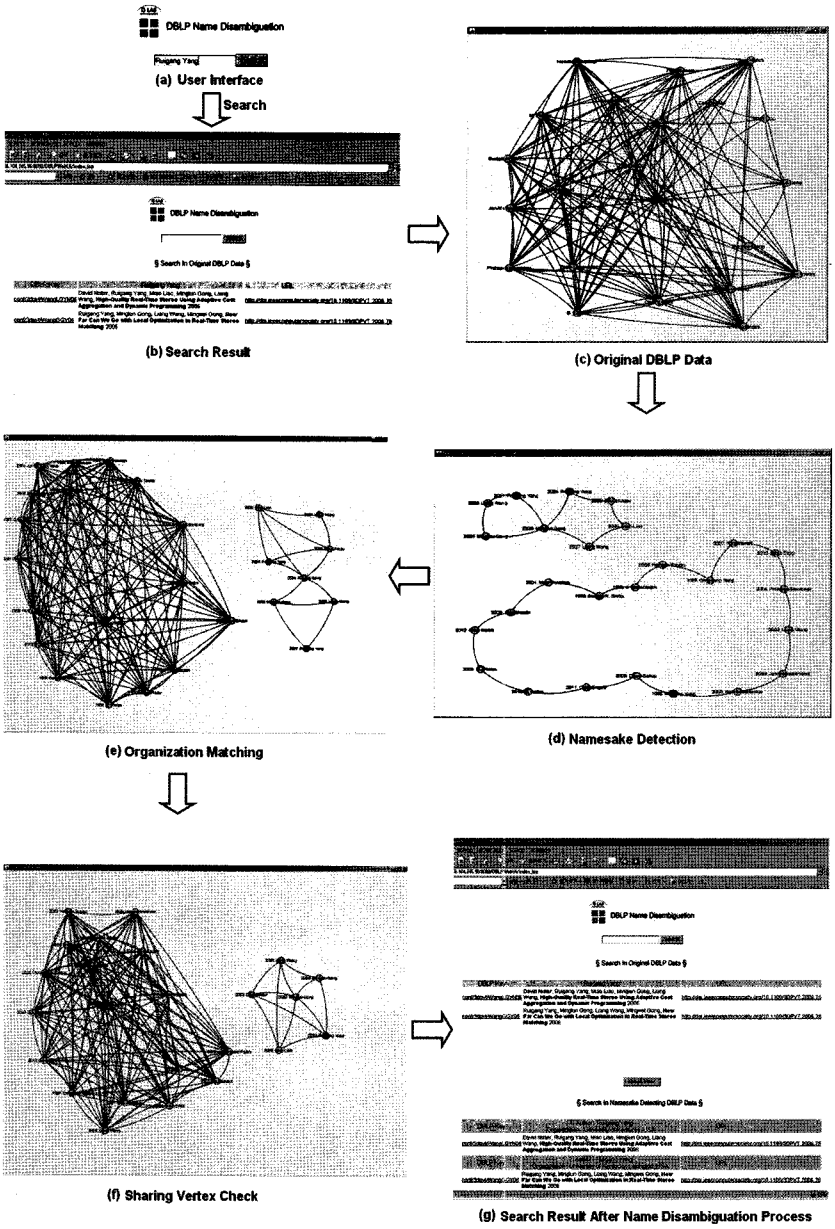


그림 8 시나리오

$$Precision = \frac{\text{시스템이 발견한 올바른 저자수}}{\text{시스템이 발견한 저자수}}$$

$$Recall = \frac{\text{시스템이 발견한 올바른 저자수}}{\text{올바른 저자수}} \quad (4)$$

$$F-measure = \frac{2*(Precision*Recall)}{Precision+Recall}$$

정확률의 경우 시스템이 찾아낸 저자의 수 중 시스템이 발견한 올바른 저자의 수로 평가하였고, 재현률의 경우 올바른 저자의 수 중 시스템이 발견한 올바른 저자의 수로 평가하였다. F-지수는 기존의 평가방식을 동일하게 적용하였다. 기준치를 0.1부터 0.9까지 다양하게 설정하여 동명이인 탐지 성능을 측정한 실험 결과는 아래 그림 9와 같다. 이 결과에서 보면 기준치가 0.4일 때 가장 성능이 좋은 것을 알 수 있었으며, 이 결과를 토대로 동명이인 탐지 성능측정을 위한 기준치는 0.4로 설정하였다.

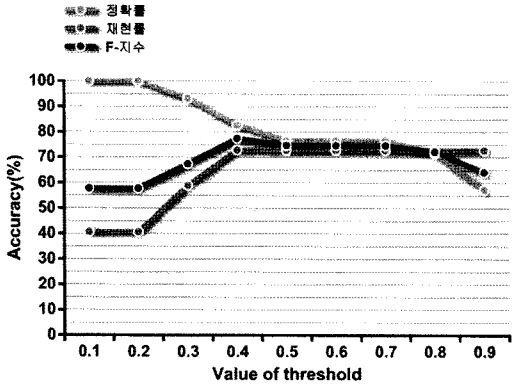


그림 9 기존의 평가방식을 이용한 동명이인 탐지 성능 평가

저자의 다른 이름 탐지 성능평가를 위한 기준치에 대한 실험은 전체 데이터 집합에서 약 1000개의 문서와 관련된 2000개의 정점 중 무작위 추출(Random Sampling) 방법을 적용하여 임의로 선택한 20명의 저자에 대하여 실험하였다.

실험은 식 (4)의 정확률과 재현률, F-지수를 이용하였다. 앞의 실험과 유사하게 기준치를 다양하게 변경하면서 유사이름 탐지성능을 측정한 결과는 그림 10과 같다. 이 결과를 토대로 유사이름 탐지 성능을 위한 기준치는 0.1로 설정하였다.

이렇게 설정된 기준치를 토대로 본 논문에서는 성능 평가 시 실세계와 유사한 환경에서의 성능 평가를 위해 불확실한 정보를 포함하는 실제 데이터를 기반으로 실험하였다.

시스템의 성능 평가를 위해 약 2,500개(요약문 정보 포함 : 1600, 미포함 : 900)의 서지 정보 데이터와 관련된 약 5000개의 정점(소속 정보 포함 : 2000, 미 포함 : 3000)에서 무작위 추출(Random Sampling) 방법을 적용하여 임의로 선택한 100개의 저자명에 해당하는 약 250개의 정점과 관련된 약 550개의 문서를 통해 수식 (4)에 해당하는 평가 방법을 사용하여 정확률, 재현률, F 지수를 측정하였다. 성능 평가 시 여러 저자의 정보가 결합된 경우, 확률을 기반으로 포함될 가능성이 가장 높은 저자에 포함되도록 하였다. 예를 들어 시스템이 A라는 저자명을 가진 저자가 '1, 2, 3, 4, 5' 논문의 저자라고 판단하였을 때, 실제 A라는 저자명을 가진 저자가 두 명 존재하고, 그 중 A'저자가 '1, 2, 3'논문의 저자이고, A"저자가 '4, 5' 논문의 저자일 때, 시스템이 찾아낸 A라는 저자의 정보는 A'에 해당하는 저자로 판단한다. 다른 경우로 한 저자의 정보가 분할된 정보의 경우, 위의 경우와 유사하게 확률을 기반으로 포함될 가능성이 가장 높은 저자에 포함되도록 하였다. 예를 들어 시스템이 B라는 저자명을 가진 저자가 B', B"두 명이 존재하고, B'저자가 '1, 2, 3'논문의 저자이고, B"저자가 '4'논문의 저자라고 판단하였을 때, 실제 B라는 저자가 한 명 존재하고 B', B" 두 명의 저자가 동일 인물일 경우 B'정보는 올바른 정보로 B"정보는 잘못된 정보로 판단한다.

성능 평가는 저자명의 모호성을 해결하는 각 단계의 중간과정 각각에 대한 평가와 단계별 성능의 변화 과정에 대하여 평가하였다. 실험 결과는 그림 11과 같다.

동명이인 탐지 단계와 저자의 다른 이름 탐지 단계 등 저자명 명확화 과정을 거치지 않고 저자명과 소속정보를 기반으로 구축된 사회망만을 이용한 경우, 정확율 83.2%, 재현율 83.1%, F 지수 83.1%의 성능을 보였다. (그림 11에서 "SN"으로 표기된 부분) 이 결과를 통해 저자명 명확화 과정을 거치지 않고 저자명과 소속 정보를 이용하여 구축된 사회망을 이용하였을 때, 저자명의 모호성을 어느 정도 해결한 결과를 보인다. 하지만 표 2에 해당하는 동명이인 탐지 단계별 변화 과정을 살펴보

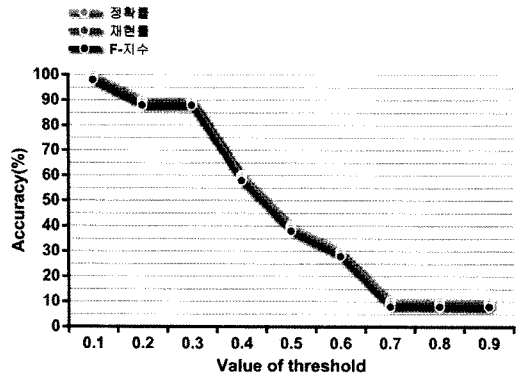


그림 10 기존의 평가방식을 이용한 저자의 다른 이름 탐지 성능 평가

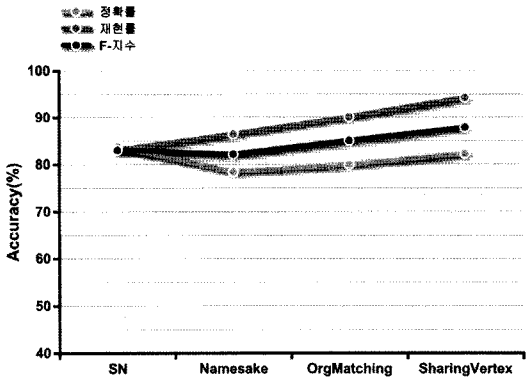


그림 11 기존의 평가방식을 이용한 성능 평가

표 2 동명이인 탐지 단계별 변화 과정

동명이인 탐지 단계별 변화 과정			
	올바른 정보	여러 저자의 정보 결합	저자의 정보가 분할
SN	52.2%	14.3%	33.5%
NameSake	55.6%	5.2%	39.2%
OrgMatching	55.2%	5.6%	39.2%
SharingV	61.4%	5.6%	33.0%

표 3 저자의 다른 이름 탐지 단계별 변화 과정

저자의 다른 이름 탐지 단계별 변화 과정			
	올바른 정보	여러 저자의 정보 결합	저자의 정보가 분할
SN	56.8%	0	43.2%
NameSake	54.3%	0	45.7%
OrgMatching	85.4%	2.4%	12.2%
SharingV	80.8%	13.5%	5.77%

면 여러 저자의 정보가 결합된 경우에 해당하는 "여러 저자의 정보 결합" 경우가 14.3%로 다른 단계에 비해 높게 나타난다. 저자명과 소속 정보만을 이용하였을 때, 여러 저자의 정보가 결합된 정점이 올바르게 분할되지 못하는 경우의 발생 빈도가 다른 단계에 비해 높이 나타나는 것을 알 수 있다.

그림 11에서 "Namesake"로 표기된 부분은 구축된 사회망에서 동명이인 탐지를 한 후의 결과이다. 동명이인 탐지 단계에서는 정확률 78.4%, 재현률 86.5%, F 지수 82.2%로 구축된 사회망을 이용한 단계와 성능의 큰 변화는 없지만 정확률의 감소와 재현률의 증가가 발생한다.

동명이인 탐지 단계는 여러 저자의 정보가 결합된 정점의 정보를 올바르게 분할하기 위한 단계로 여러 저자의 정보가 한 정점으로 결합된 비율이 저자명과 소속정보를 기반으로 구축된 사회망(표 2에서 "SN"으로 표기된 부분) 단계의 14.3%에서 5.2%로 감소하였고, 이로 인해 재현률이 증가하였다. 하지만 저자의 정보를 분할하는 과정에서 잘못된 분할로 인해 한 저자의 정보가 여러 저자로 잘못 판단된 비율이 33.5%에서 39.2%로 증가하였고, 이를 통해 정확률의 감소를 야기시켰음을 알 수 있다. 소속 비교 단계는 한 저자가 여러 이름을 사용하는 경우에 의해 발생하는 저자명의 모호성을 해결하기 위한 단계로, 소속 비교 단계(OrgMatching) 후 표 3의 저자의 다른 이름 탐지 성능의 증가를 보인다. 구축된 사회망과 동명이인 탐지 단계의 경우, 시스템이 올바른 정보를 탐지한 경우가 약 55%(구축된 사회망 : 56.8%, 동명이인 탐지 : 54.3%)이고 한 저자의 정보가 여러 저자로 잘못 분할된 경우가 약 45%(구축된 사회망 43.2%, 동명이인 탐지 45.7%)의 비율을 보인다. 하지만 소속 비교 단계를 거친 후 올바른 정보의 경우 85.4%, 저자의 정보가 분할된 경우가 12.2%로 저자의 다른 이름 탐지 성능의 증가를 보인다. 마지막으로 저자 공유 탐지 단계 후 동명이인 탐지 성능이 61.4%로 증가하고, 저자의 정보 결합 시 잘못된 결합으로 인해 저자의 다른 이름 탐지 성능이 80.8%로 감소한다.

저자명의 모호성을 해결하기 위한 모든 처리과정을 거친 후, 시스템의 성능은 정확률 82.3%, 재현율 94.3%, F-지수 87.9%로 불확실한 정보를 포함하는 실제 데이터를 기반한 실험 환경에 비해 만족할만한 성능을 보이는 것을 확인할 수 있다.

7.2.2 개선된 평가 방식을 이용한 성능 평가

7.2.1의 기존의 평가 방식을 이용한 성능 평가를 통해 시스템의 성능을 평가하였지만, 기존의 평가 방식은 저자명의 모호성을 해결하기 위한 시스템의 성능 평가 방식에는 적절하지 못한 부분이 존재한다고 판단하였다. 기존의 평가 방식을 이용한 성능 평가 시 여러 저자의 정보가 결합된 경우나 한 저자의 정보가 분할된 경우, 확률을 기반으로 포함될 가능성이 가장 높은 저자에 포함되도록 하여 성능을 평가하였다. 하지만 완벽하게 일치(Exact matching)하는 경우와 적절하게 일치(Appropriate match)하는 경우의 정보의 질(Quality of Information)은 차이가 존재한다. 예를 들어 실제 A라는 저자가 '1, 2, 3, 4, 5'논문의 저자일 경우, 시스템이 A라는 저자가 '1, 2, 3, 4, 6'논문의 저자라고 판단하는 첫 번째 경우와 A라는 저자가 '1, 2, 3, 4, 5'논문의 저자라고 판단하는 두 번째 경우, 두 경우 모두 저자의 정보를 탐지한 경우이다. 하지만 정보의 질적인 측면에서 볼 때, 두 번째 경우가 좀 더 정확한 검색 결과를 제공한다.

기존의 평가 방식을 이용한 실험의 경우, 위의 예와 같은 정보의 질적인 평가까지는 고려하지 못하므로 개선된 성능 평가 방식을 통해 실험하였다.

성능 평가는 식 (5)에 해당하는 정확률과 재현률, F-지수를 이용하여 기존의 평가 방식을 이용한 성능 평가

와 동일한 실험 환경에서 실험을 진행하였다.

$$Precision = \frac{(1+\alpha) * P * P_{sub}}{P + (\alpha * P_{sub})}$$

$$P = \frac{\text{시스템이 발견한 올바른 저자수}}{\text{시스템이 발견한 저자수}}$$

$$P_{sub} = \frac{\text{시스템이 발견한 일부정보가 올바른 저자수}}{\text{시스템이 발견한 저자수}}$$

$$Recall = \frac{(1+\alpha) * R * R_{sub}}{R + (\alpha * R_{sub})}$$

$$R = \frac{\text{시스템이 발견한 올바른 저자수}}{\text{올바른 저자수}} \quad (5)$$

$$R_{sub} = \frac{\text{시스템이 발견한 일부정보가 올바른 저자수}}{\text{올바른 저자수}}$$

$$F-measure = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

식 (5)에서 정확률과 재현률을 계산할 때, 저자의 일부 정보만 포함하는 정점에 대한 평가도 함께 하기 위해 *P_sub*과 *R_sub*을 함께 고려한다. 이때, 일부 정보만 일치하는 경우에 대하여 페널티를 부가하기 위해, 가중치 α를 부여한다. 이 때 가중치는 0.5로 정하였다. P와 R의 경우 식 (4)에 해당하는 기존 평가 방식의 정확률과 재현률과 동일하며, F-지수는 기존의 평가 방식을 동일하게 적용하였다.

기준치를 0.1부터 0.9까지 다양하게 설정한 후, 개선된 평가방식을 이용하여 동명이인 탐지 성능을 측정한 실험 결과는 아래 그림 12와 같다. 기존의 평가방식을 이용한 실험과 성능의 차이는 존재하지만 동일하게 기준치가 0.4일 때 가장 성능이 좋은 것을 알 수 있었으며, 이 결과를 토대로 개선된 동명이인 탐지 성능측정을 위한 기준치는 0.4로 설정하였다.

기존의 평가방식을 이용한 저자의 다른 이름 탐지 성능 평가와 동일한 실험 환경에서 개선된 평가방식을 이용한 저자의 다른 이름 탐지 성능 평가를 위한 기준치

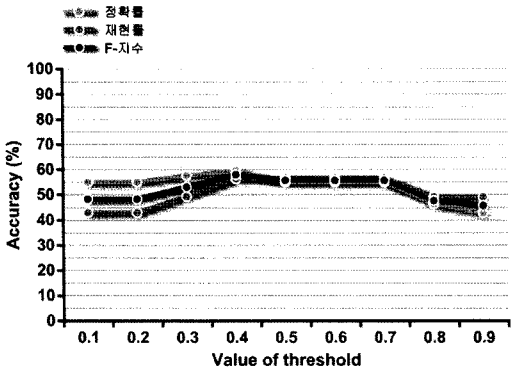


그림 12 개선된 평가방식을 이용한 동명이인 탐지 성능 평가

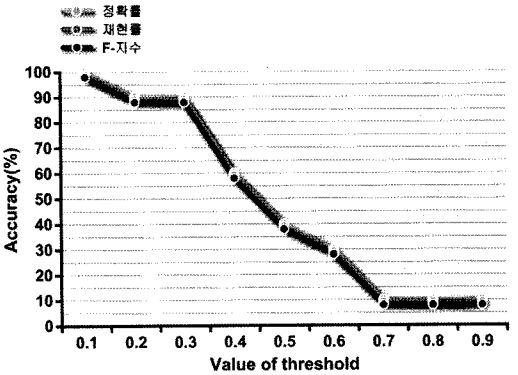


그림 13 개선된 평가방식을 이용한 저자의 다른 이름 탐지 성능 평가

에 대한 실험 결과는 그림 13과 같다. 이 결과를 토대로 개선된 평가 방식을 이용한 유사이름 탐지 성능을 위한 기준치는 0.1로 설정하였다.

이렇게 설정된 기준치를 토대로 기존의 평가방식을 이용한 성능 평가와 동일한 실험 환경에서 제안한 시스템의 성능을 평가하였다. 성능 평가를 위한 정확률과 재현률, F-지수는 식 (5)에 해당하는 방법을 사용하였고, 가중치는 0.5로 정하였다.

성능 평가는 저자명의 모호성을 해결하는 각 단계의 중간과정 각각에 대하여 평가하였다. 실험 결과는 그림 14와 같다.

그림 14의 실험 결과를 통해 개선된 평가방식을 이용한 성능 평가 결과는 7.1.2절의 기존의 성능 평가 방식을 이용한 실험 결과에 비해 성능이 현저히 감소된 것을 확인할 수 있고, 실제로 올바르게 판단된 정보 중 다수의 정보가 적절하게 일치(appropriate matching)하는 경우에 해당한다. 이 결과를 통해 제안한 시스템의 성능 평가 시 정보의 질적인 측면도 함께 고려하여 성능을 평가하기에 개선된 평가방식을 이용한 성능 평가가 적

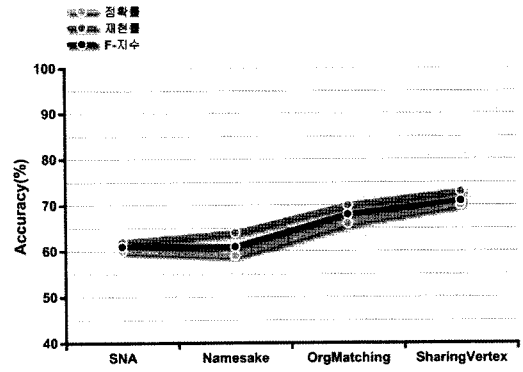


그림 14 개선된 평가방식을 이용한 성능 평가

합하다고 판단하였다.

동명이인 탐지 단계와 저자의 다른 이름 탐지 단계 등 저자명 명확화 과정을 거치지 않고 저자명과 소속정보를 기반으로 구축된 사회망만을 이용한 경우, 정확률 60%, 재현률 62%, F 지수 61%의 성능을 보였다(그림 14에서 "SN"으로 표기된 부분). 이 결과를 통해 저자명과 소속 정보만을 이용하여 저자명의 모호성을 어느 정도 해결하였지만 만족스러운 결과를 보이지 못하였다.

그림 14에서 "Namesake"로 표기된 부분은 구축된 사회망에서 동명이인 탐지를 한 후의 결과이다. 동명이인 탐지 단계에서는 정확률 59%, 재현률 64%, F 지수 61%로 성능의 큰 변화가 없는 것을 확인할 수 있고, 이러한 결과는 기존의 평가 방식을 이용한 성능 평가 방식과 동일하게 여러 저자의 정보가 결합된 정점의 정보를 올바르게 분할하기 위한 단계로 여러 저자의 결합된 정보의 분할을 통한 재현률의 증가와 저자의 정보를 분할하는 과정에서 잘못된 분할로 정확률의 감소가 발생하였다. 소속 비교 단계와 저자 공유 탐지 단계 후 저자의 다른 이름 탐지를 통해 성능이 증가하는 것을 확인할 수 있다.

저자명의 모호성을 해결하기 위한 모든 처리과정을 거친 후, 시스템의 성능은 정확률 70%, 재현률 73%, F 지수 71%로 정보의 질까지 고려한 성능 평가 방식을 이용하여 불확실한 정보를 포함하는 실제 데이터를 기반한 실험 환경에 비해 만족할만한 성능을 보이는 것을 확인할 수 있다.

8. 결 론

본 논문에서는 사회망을 이용한 저자명 명확화 기법을 제안하였다.

저자들간의 관계를 기반으로 구축된 사회망을 이용하여 이름의 문법적 유사도 뿐만 아니라, 저자들간의 의미적 관계까지 고려하였다. 또한 저자의 소속 정보나 논문의 요약문 정보 등 시스템의 성능에 영향을 미치는 중요한 특징들을 포함하지 않은 불확실한 정보를 포함하는 실제 데이터에 대한 실험에서도 만족할만한 성능을 보였다.

향후 과제로는 유사도 평가 방식과 순환 탐지 알고리즘의 정제를 통해 불확실한 데이터에 강화(Robust)된 시스템을 구축하고, 좀 더 방대한 양의 데이터에서 시스템의 성능을 평가하고자 한다. 또한 에러 데이터를 포함하지 않는 데이터를 이용하여 다른 시스템과의 비교 평가를 통해 시스템의 성능을 평가하고자 한다.

참 고 문 헌

[1] Dunn, H. L, Record Linkage. American Journal of Public Health 36, pp. 1412-1416, 1946.

[2] D. Bitton, D. .J. DeWitt, Duplicate Record Elimination in Large Data Files. ACM Transactions on Database Systems, pp. 255-265, 1983.

[3] M. Hernandez, S. Stolfo, The merge/purge problem for large databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 127-138, 1995.

[4] Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski, Lukasz A. Kurgan, Data Mining: A Knowledge Discovery Approach. 2003.

[5] W. W. Cohen, H. A. Kautz, D. A. McAllester, Hardening soft information sources. In Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, pp. 255-259, 2000.

[6] Amit Bagga, Coreference, cross-document coreference, and information extraction methodologies. Computer science, Information Systems, pp. 4234, 1999.

[7] L. Karl Branting, A Comparative Evaluation of Name-Matching Algorithms. ICAIL'03, pp. 224-232, 2003.

[8] Y. Chen, J. Martin, Towards Robust Unsupervised Personal Name Disambiguation. EMNLP and CNLP, pp. 190-198, 2007.

[9] H. Han, C. L. Giles, H. Zha, C. Li, K. Tsioutsiouliklis, Two supervised learning approaches for name disambiguation in author citations. In proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 296-305, 2004.

[10] O. Zaiane, J. Chen, and R. Goebel, DBconnect:. Mining research community on dblp data. In. Proceedings of WebKDD/SNAKDD 2007.

[11] Jorge Gracia, Vanesa Lopez, Mathieu d'Aquin, Marta Sabou, Enrico Motta, Eduardo Mena, Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching. International Semantic Web Conference ISWC-2007, 2007.

[12] Dongwook Shin, Jinbeom Kang, Joongmin Choi, Jaeyoung Yang, Detecting Collaborative Fields Using Social Networks, 2008 Fourth International Conference on Networked Computing and Advanced Information Management, pp. 325-328, 2008.

[13] D. B. Johnson, Finding all the elementary circuits of a directed graph, SIAM J. Comput. Vol 4. pp. 77-84, 1975.

[14] G. V. Cormack, O. Lhotak, C. R. Palmer, Estimating precision by random sampling, Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 273-274, 1999.

신 동 욱

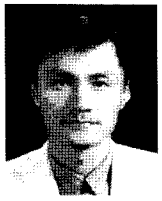
2007년 경원대학교 소프트웨어대학 졸업(학사). 2009년 한양대학교 대학원 컴퓨터공학과 졸업(석사). 2009년~현재 한양대학교 대학원 컴퓨터공학과 재학(박사) 관심분야는 정보검색, 정보추출, 데이터마이닝, 인공지능

김 태 환

2007년 한양대학교 컴퓨터공학과 졸업(석사). 2007년~현재 한양대학교 컴퓨터공학과 박사 과정. 관심분야는 웹마이닝, 웹지능, 정보추출, 시맨틱웹과 온톨로지, 인공지능

정 하 나

2008년 경원대학교 소프트웨어대학 졸업(학사). 2008년~현재 한양대학교 대학원 컴퓨터공학과 재학(석사). 관심분야는 정보검색, 정보추출, 웹데이터마이닝, 온톨로지, 인공지능



최 중 민

1984년 서울대학교 컴퓨터공학과 졸업(학사). 1986년 서울대학교 대학원 컴퓨터공학과 졸업(석사). 1993년 State University of New York at Buffalo, Computer Science 졸업(박사). 1993년~1995년 한국전자통신연구원(ETRI) 인공지능연구실 선임연구원. 1995년~현재 한양대학교 컴퓨터공학과 교수. 관심분야는 웹마이닝, 웹지능, 정보추출, 시맨틱웹과 온톨로지, 인공지능