사회연계망 구축을 위한 인용 매칭에서의 인용 필드 분해 영향 분석

구희관* 강인수** 정한민** 이승우** 성원경**
*과학기술연합대학원대학교 응용정보과학
**한국과학기술정보연구원 정보기술개발단
{ hkkoo, dbaisk, jhm, swlee, wksung }@kisti.re.kr

Influence of Citation Field Segmentation on Citation Matching for Social Network Construction

HeeKwan Koo* In-Su Kang** Hanmin Jung** Seungwoo Lee** Won-Kyung Sung**
*Practical Information Science, UST
**Information Service Research Lab., KISTI

요 약

인용 매칭(Citation Matching, CM)은 동일한 논문을 지칭하는 인용레코드(Citation Record)를 군집화하는 것으로 인용 관계를 가진 사회연계망 구축시 필요한 기술의 하나이다. 인용 매칭의 전단계로써, 인용 레코드를 저자, 논문 제목, 게재지명, 발행연도 등의 필드로 구분하는 인용 필드 분해가 고려될 수 있다. 본 논문은 인용 필드 분해(Citation Field Segmentation, CFS)와 인용 매칭의 상관관계를 분석하고자 한다. 즉, 인용 필드 분해가 인용 매칭에 필수적인 단계인지를 밝히고 개별 인용 필드가 인용 매칭에 미치는 영향을 분석한다. 실험을 통해 인용 필드 분해를 한 인용 매칭(CFS-based CM)이 인용 필드 분해를 적용하지 않은 인용 매칭(CFS-free CM)에 비해 1% 내외의 성능의 차이를 보이므로, 인용매칭의 성능에 크게 영향을 미친다고 보기 어려웠다. 이는 인용 레코드의 서로 다른 필드 들 사이에서 어휘 중복 비율이 크게 낮기 때문에 따로 필드를 구별하지 않아도 필드가 구별되는 특성때문이었다.

1 서론

오늘날 연구자는 정보통신과 컴퓨터 공학의 발달로 인해 전자도서관이나 인터넷을 통해 논문을 쉽게 획득할 수 있다. 그러나 연구자가 몇 개의 단어로 검색된 많은 검색 결과 중에 필요한 논문을 찾는 것은 쉽지 않다. 연구자들의 논문에서 추출된 인용은 연구자가 필요한 논문을 찾는 데 있어 이용 가능한 수단이 될 수 있다. 예를 들어, CiteSeer [1] 는 키워드 검색에 의한 논문 검색 결과를 인용 순으로 정렬해서 보여줌으로써, 같은 분야를 연구하는 다른 연구자들의 논문에서 인용을 많이 받은 논문을 쉽게 찾을 수 있도록 도움을 주고 있다. 단순히 논문 선택의 문제를 떠나, 자신의 연구 분야에 대한 변화된 연구의 흐름을 빠르게 받아들여야 하는 경우나, 정해진 분야의 범주를 넘어 여러 다양한 분야를 대상으로 하는 연구에 대한 요구가 나날이 증가되어 가는 상황에서, 인용에 대한 수집과 이를 가공한 정보는 연구자에게 꼭 필요한 정보가 되고 있다.

문헌과 문헌과의 인용 관계를 형성하기 위한 전단계로서, 인용 매칭(Citation Matching, CM)은 동일한 논문을 지칭하는 인용레코드(Citation Record)를 군집화(Clustering)하는 것을 말한다. 인용 레코드는 논문에서 수집된 하나의 인용을 지칭하며, 저자, 논문 제목, 게재지명 등의 다양한 필드로 구성이 된다.

그림 1과 2는 인용 필드 분해와 인용 매칭의 예를 각각 보이고 있다. 그림 1에서처럼 인용 필드 분해(Citation Field Segmentation)는 인용 레코드를 각각의 필드(저자, 논문제목, 발행연도 등)로 구분하는 것이다.

인용 필드 분해를 적용한 인용 매칭 방법(CFS-based CM)은 필드별 유사도를 비교하여 동일 인용 여부를 판별하는 것을 말한다. 이는 각 필드마다 다양하게 유사도를 적용할 수 있는 장점을 가진다. 저자 필드만을 예로 들어 설명하면, 그림 2

[1] http://citeseer.ist.psu.edu/

의 “Ion Stoica”라는 저자이름을 서로 비교하는 경우, 유사도 측정 방법을 영문 이름의 첫글자 및 성(예, I. Stoica)을 이용하거나, 성만을 이용하는 등(예, Stoica)의 유사도 측정 방법을 필드마다 다르게 적용할 수 있다. 이외에도, 논문 제목 필드의 가중치를 다른 필드보다 높게 적용하는 방식의 필드별 가중치를 서로 다르게 할당할 수 있는 유형을 갖는다.

인용 필드 미분해 기반 인용 매칭 방법(CFS-free CM)은 필드 분해를 적용하지 않고 인용 레코드 레벨의, 편집거리(Edit Distance)나 단어단위에 가중치를 부여한, 유사도 비교 방법을 수행하여 특정 임계값 내의 유사도를 갖는 인용을 군집화하는 방법이다. 이는 인용 필드 분해의 오류로 인한 인용 매칭의 성능저하를 피할 수 있는 장점이 있다.

본 논문은 인용 매칭에서 인용 필드 분해의 유용성 여부를 알아보기 위해 인용 필드 분해가 인용매칭에 주는 영향을 분석하고자 한다. 다시 말하면, 인용 필드 분해가 필수적으로 선행되어야 하는지, 생략 가능한 것인지에 관해 밝히고 그 결과에 대한 분석을 수행한다.

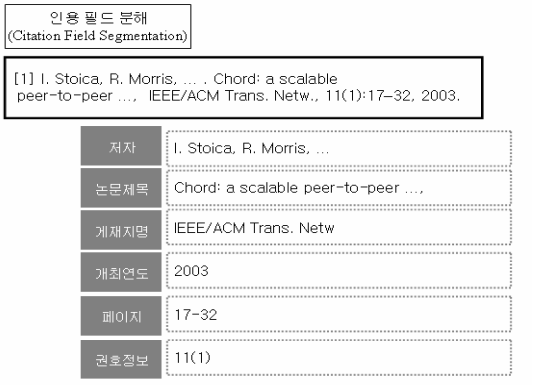

그림 1. 인용 필드 분해의 적용 예

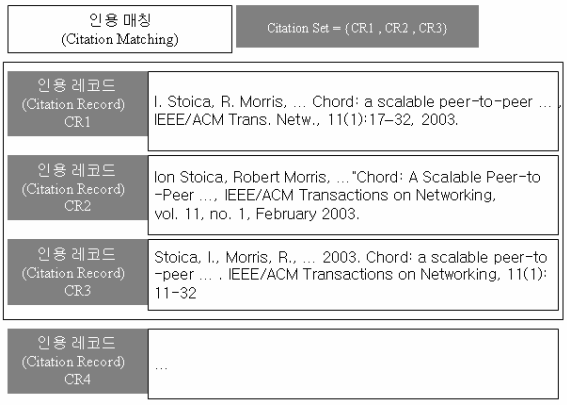

그림 2. 인용 매칭의 적용 예

본 논문의 구성은 2장에서 관련연구를 다루고, 3장과 4장에서는 각각 실험계획 및 실험 결과를 기술하고, 5장에서 결론을 기술한다.

2. 관련연구

2.1 CiteSeer

인용 매칭을 자동적으로 시도한 잘 알려진 예는 CiteSeer를 들 수 있다. CiteSeer는 현재 약 80만여 개 논문의 인용을 추출하여 인용에 관련한 다양한 서비스를 제공하고 있다. 인용 매칭을 위해 CiteSeer가 제안한 대표적 알고리즘은 단어와 구 비교 알고리즘(Word and Phrase Matching Algorithm)이었다. 단어와 구 비교 알고리즘은 비교되는 두 개의 인용 레코드(Citation Record)에서 일치하는 단어 수와 연속으로 발생하는 단어 열에 대해 가중치를 부여하여, 군집화하는 알고리즘이다. 이 알고리즘의 대표적인 문제점으로, 인용 레코드 간의 비교에서 연속된 단어의 일치가 대부분 제목에서 발생할 수밖에 없기 때문에, 주로 제목을 이용하는 인용 매칭 알고리즘이라는 한계를 가진다[14][15].

최근 CiteSeer는 일괄적인 인용 매칭 알고리즘의 해결을 위해 검색엔진을 이용한 인용 매칭 방법을 제안하였다. 한 개의 인용레코드를 임의로 선택한 후, 검색엔진에 질의하여, 검색된 일정한 편집거리(Edit Distance) 내에 인용들을 대상으로 인용매칭을 수행하였다. 그러나 검색엔진에 사용된 인용 필드가 저자와 논문 제목만으로 한정되어 사용되었기 때문에, 인용된 논문의 게재지 명 및 권호정보, 발간연도 등의 다양한 정보를 이용하여 인용매칭을 수행하기에는 무리가 있다[12].

2.2 인용 필드 분해

인용 필드 분해는 크게 규칙 기반 방법(Rule Based Method)과 통계 기반 방법(Stochastic Method)으로 나눌 수 있다. 규칙 기반 방법의 장점은 구현이 비교적 단순하고, 규칙을 이해하기 쉬우며, 규칙이 적용되는 사례에 한해 높은 정확성을 보이는 것이다. 규칙 기반 방법의 단점은 규칙이 증가할수록 규칙의 통제가 어렵다는 것이다. 통계기반 방법의 장점은 잘 정의된 방법들과 도구들을 활용할 수 있다는 것이다. 통계기반 방법의 단점은 학습 데이터가 필요하며, 생성된 결과가 학습데이터에 의존적이라는 것이다.

인용 필드 분해에서 통계기반 방법은 다양하게 적용되었다. HMM (Hidden Markov Model)을 기반으로 논문의 헤더(논문제목, 저자, 소속기관, 요약 등)와 인용(저자, 논문제목, 게재지명, 출판연도 등)

을 분해하거나[1], HMM보다 더 많은 다양한 Feature를 이용하는 CRF(Conditional Random Field)를 적용하여 분해하거나[3][7], CRF로 학습하는 학습 단어 단위를 조절하는 Semi-CRF 등을 적용하여 인용 필드를 분해하였다[11]. 이외에도 ME(Maximum Entropy), SVM(Support Vector Machine) 등의 방법들도 인용 필드 분해에 사용되었다[9].

인용 필드 분해에 규칙기반 방법 또한 다양하게 적용되었다. 정규식(Regular Expression)을 적용하여 인용 필드를 분해하거나, 대규모 인용 필드 추출 템플릿(Template)을 생성하고 이를 서열 정렬(Sequence Alignment)을 적용하여 분해하거나[10], 인명사전, 일반명사, 게재지명 등의 사전기반 규칙을 적용하여 단어 단위로 태깅하고 이들을 결합 규칙을 적용하여 분해하거나[6], 저자와 논문에 대한 축적된 지식을 이용한 인용 필드 분해를 하였다[5].

2.3 인용 매칭

인용 매칭에 사용되는 방법은 다시 나누면, 동일한 논문을 지칭하는 인용 레코드를 저자, 논문 제목 등으로 구별하는 인용 필드 분해를 사용하는 방법 ([2], [4], [11], [12])과 이를 인용 레코드 레벨에서 유사도 비교 알고리즘을 이용하는 경우([13],[14],[19])로 나누어 볼 수 있다.

인용 매칭 역시 크게 구분하면 규칙 기반 방법과 통계 기반 방법으로 구분할 수 있다. 규칙 기반 인용 매칭 방법은 통계 기반 방법에 필요한 학습데이터를 생성할 필요가 없어, 실제 시스템에 빠르게 적용할 수 있는 장점을 갖는다 [2][11][12][14][19]. 일례로는 CiteSeer, eprint의 OPCIT[2] (The Open Citation Project) 등을 들 수 있다[16]. 검색엔진을 이용한 인용 매칭 방법으로 다시 나누면, 논문에 기술되어 하나의 논문을 지칭하는 인용 레코드를 저자, 논문 제목 등으로 구별하는 인용 필드 분해를 사용하는 방법 ([2][11][12])과 이를 하나의 인용 레코드로 사용하여 비교 알고리즘을 이용하는 경우([13], [14], [19])로 나누어 볼 수 있다.

인용 매칭에서의 통계 기반 학습 방법으로는 Markov Chain Monte Carlo (MCMC)를 적용하거나[8], CRF (Conditional Random Field)와 MRF(Markov Random Field) 등이 적용되었다[3][18].

3. 실험 계획

3.1 개요

기존 CFS-based CM은 인용 필드 분해가 수행된 인용 레코드에 대해 단지 저자, 논문 제목, 게재지 명, 출판연도 등의 필드를 선택적으로 이용하여 인용 매칭을 수행하였다([2], [12], [11]). CFS-free CM([19])에서는 인용 필드 분해를 수행하지 않고, TF·IDF를 기본으로 한 인용 매칭 방법을 적용하여 가장 좋은 성능을 보였다. 그러나 이 두 가지 방법을 직접적으로 비교해서 인용 매칭에 어느 방법이 적절한 가에 대한 실험이 진행된 적이 없기 때문에, 인용매칭을 수행하고자 할 때, 우선적인 선택의 기준이 명확하게 제공되지 않았다.

공통의 테스트 셋을 이용해 CFS-based CM과 CFS-free CM에 대해 인용 매칭의 성능을 비교한다. 테스트 셋은 인용 필드 분해를 위해, 기존의 자동화된 분해 방법을 적용할 수도 있겠으나, 본 연구에서는 자동 인용 필드 분해의 오류가 인용 매칭에 미치는 영향을 배제하기 위해 수작업으로 분해된 인용 필드들을 사용한다. 이를 위해, 본 연구에서는 수작업으로 인용 필드 분해가 수행된 McCallum 테스트 셋을 사용할 것이다[2][19].

3.2 실험 데이터 셋

본 논문의 실험 데이터로는 McCallum[2]의 인용 매칭 테스트 셋[3] 을 사용하였다. 이 테스트 셋은 인공지능 관련 논문들의 인용 레코드(Citation Record)를 수집하여 인용 필드 분해 및 인용 군집을 수작업으로 수행하여 만들어진 테스트 셋이다. 인용 군집이란, 같은 논문을 가리키는 서로 다른 인용 레코드들을 하나의 클러스터로 묶는 것이다. 하나의 인용 레코드는, 저자, 논문 제목, 게재지 명칭(논문지/학술대회 논문집), 연도, 권호정보, 페이지 번호 등의 주요 인용 필드뿐만 아니라, 출판사, 학술대회 개최 지역/장소, 편집자 등 실제 인용 레코드에 출현하는 모든 항목들에 대해 개별 필드명을 부여하여 총 16개의 필드로 구성되어 있다. 전체 테스트 셋은 수작업으로 만든 사람에 따라 3개의 파일로 나뉘어져 있으며, 총 1,879개의 인용 레코드로 구성되어 있다. 이 중, 본 논문에서는, 논문 제목이 누락된 인용 레코드를 제외하고 1,838개의 인용 레코드에 대해 실험을 수행하였다. 1,838개의 인용 레코드 집합에서, 인용 군집의 수는 187개이며, 평균 인용 군집의 크기는 약 10였다. 추가적으로, 본 연구에서, 인용 필드 구별을 사용하지 않는 인용 매칭 실험을 수행하기 위해, McCallum 테스트 셋의 각 인용 레코드에 필드 구별을 하지 않는 원

[2] http://opcit.eprints.org/

[3] http://www.cs.umass.edu/~mccallum/data/cora-refs.tar.gz

시 인용 레코드를 Reference라는 이름의 필드로 만들어 추가하였다. 또한, <year> 필드가 존재하지 않고 <date>로 연도가 태깅이 되어 있는 경우에는, <date>필드에서 연도를 추출하여 <year>필드로 추가하였다. 이렇게 구성된 테스트 셋에서 하나의 인용레코드에 대한 예는 다음과 같다

aha1987 <DocID>1</DocID><author> Aha, D. and Kibler, D. </author> <title> Learning Representative Exemplars of Concepts: An Initial Case Study. </title> <booktitle> In Proceedings of the Fourth International Conference on Machine Learning, </booktitle> <pages> pages 24-30, </pages> <address> U. C. Irvine, CA, </address><year>1987. </year> <publisher>Morgan Kaufmann. </publisher><Reference> Aha, D. and Kibler, D. Learning Representative Exemplars of Concepts: An Initial Case Study. In Proceedings of the Fourth International Conference on Machine Learning, pages 24-30, U. C. Irvine, CA, 1987. Morgan Kaufmann. </Reference>

3.2 실험 방법

본 실험의 목적은 인용 필드 분해가 인용 매칭에 미치는 영향을 파악하는 것이다. CFS-based CM의 성능과 CFS-free CM의 성능을 비교한다면, 인용 필드 분해가 인용 매칭에 어떤 영향을 미치는 가를 파악할 수 있다.

검색엔진은 인용 매칭 관련 여러 연구에서 사용된, 자바기반 오픈소스 검색엔진, Lucene 2.2[4] 버전을 사용하였다[11][12].

그림 3는 인용 매칭 실험에 관해 전체적인 절차를 보여준다. 전체 절차를 개략적으로 설명하면 다음과 같다.

첫 단계는 색인 단계이다. 이 단계에서 문자와 숫자를 제외한 모든 기호를 제거하고 모든 문자를 소문자로 변환하는 전처리 단계를 거친다. 다음으로 인용 매칭 테스트 셋을 색인으로 구성할 때는, 전체 테스트 셋 내의 모든 필드에 대해 필드별 색인을 생성한다. 즉, 16개의 기본 인용 필드 이외에 추가된 <Reference>, <DocID>를 색인으로 생성한다. 색인 생성 방법은 문자 외에 숫자도 포함해 색인을 생성하여, 숫자로 구성되어 있는 페이지 정보나 권호 정보를 검색 가능하도록 하였다.

[4] http://lucene.apache.org/

색인용어 추출기법은 6가지 방법을 비교했다. 전처리만 수행한 색인(No Applying Index)와 불용어 목록에 " a, an, the "만 추가한 색인(Applying StopWord1)과 180개의 불용어[5] 를 사용한 색인(Applying StopWord2)과 포터스테머[6] (Porter Stemmer)를 적용한 색인과 불용어1과 포터스테머를 함께 적용한 색인(Applying StopWord1 + PorterStemmer)와 불용어2와 포터스테머를 함께 적용한 색인(Applying StopWord2 + PorterStemmer)들로 구성했다.

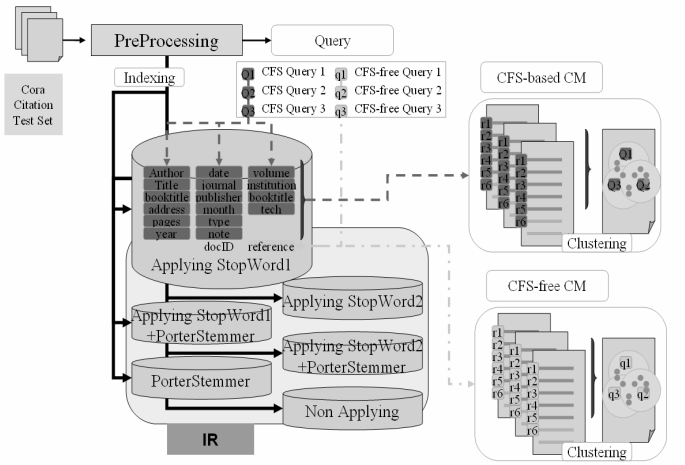

그림 3. 전체 인용 매칭 절차

그림 3에서 검색엔진을 이용해 검색 결과를 생성할 때는, CFS-based CM의 결과를 생성하기 위해서 다중 필드 질의(Multi Field Query)를 생성하여 검색엔진에 질의하고, 검색엔진이 각 필드별 질의를 하나의 검색 결과로 병합하여 생성한다. CFS-free CM의 검색 결과를 생성할 때는, 인용 레코드 내에서 <Reference> 필드의 단어를 이용해서 단일 필드 질의(Single Field Query)를 생성하고 검색 결과를 생성한다.
검색에 이용된 검색 모델은 벡터 모델이며, Lucene이 사용하는 유사도 계산 방법은 다음과 같다.

$$score(q,d) = coord(q,d) \times queryNorm(q) \times \left(\sum_{t \in q} (tf(t \in d) \times idf(t)^2 \times Boost(t.field \in d) \times Norm(t,d)) \right)$$

질의(q)와 문서(d)의 유사도는 TF(Term Frequency)와 IDF(Inverse Document Frequency)의 제곱값과 색인을 생성할 때 설정한 가중치(Boost(t.field ∈ d))와 필드에 포함된 단어 개수

[5] http://rdsweb2.rdsinc.com/help/stopword_list.htm
[6] http://www.tartarus.org/martin/PorterStemmer/

및 길이를 정규화한 값(Norm(t,d))을 곱한 값의 합에 문서 내에 포함된 개별 검색어의 개수에 대한 조절값(coord(q,d))과 개별검색어의 가중치 값에 제곱의 합을 정규화한 값(queryNorm(q))의 곱으로 구성된다.

생성된 검색 결과는 일정한 임계치(Θ)를 군집을 생성하는 기준으로 사용한다. 즉 특정 임계치(Θ)값 이상의 인용 레코드들은 같은 인용 군집 내에 포함되어 있다고 가정하고 이를 이용하여 군집화를 수행한다. 이렇게 임계치 이내의 결과값을 이용해 군집화를 수행하기 때문에 군집의 최대 크기를 보장하는 단일 링크 방법이 적용되었다.

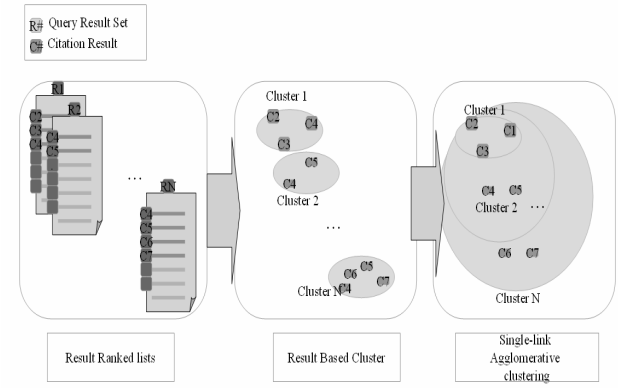

그림 4. 인용 레코드의 군집화 방법

(Single-link Agglomerative Clustering)

그림 4는 실험결과에 대해 성능평가를 수행하기 위해서 검색 결과 내에 특정 임계값 이상의 인용에 대해 군집화(Single-link Agglomerative Clustering)를 수행하는 것을 보여준다. 이것은 검색 결과 내에 인용에 대해 관계를 설정하게 함으로써, 군집화를 수행한다. 예를 들어, 그림 3의 좌측의 첫 번째 단계에서 임계치 이내의 검색 결과인 R1의 "C2, C3, C4"는 중앙의 인용 군집 1에 포함되게 된다. 그리고 마지막 단계에서 서로 같은 인용을 포함하고 있는 군집끼리 하나의 군집으로 생성되는 것을 보여준다. 두 번째 단계의 인용 군집 1과 인용 군집 2는 "C4"라는 인용을 함께 포함하고 있기 때문에 하나의 군집이 되며, 이와 마찬가지로 군집 N과 군집 2는 "C4, C5"라는 인용을 공통으로 포함하고 있기 때문에 하나의 군집으로 형성이 된다.

인용 매칭 성능은 각각의 군집의 F1[7] 성능을 이용하였다.

[7] F1 = 2 * Precision * Recall / (Precision + Recall)

4. 실험 결과

4.1 인용 필드 분해의 영향 평가

인용 필드 분해의 영향에 대한 성능을 측정한 결과는 다음과 같다. 인용 검색 결과는 질의어와 인용 필드에 각각 전처리적용, 불용어1(S1), 불용어2(S2), 포터스테머(P), 불용어1과 포터스테머의 조합(S1P), 불용어2와 포터스테머의 조합(S2P)의 조합을 적용하고 성능을 측정하였다. 불용어1은 "a, an, the"의 관사들로 구성하였고, 불용어2 는 180개의 불용어 리스트로 구성하였다.

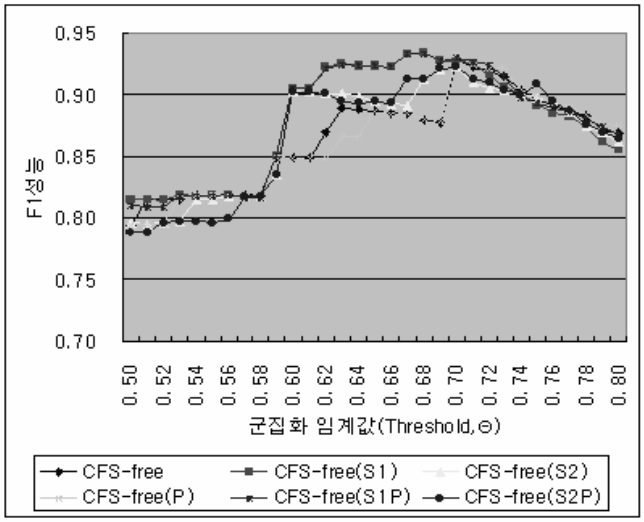

그림 5. 인용 필드 미분해의 인용 매칭 성능 결과

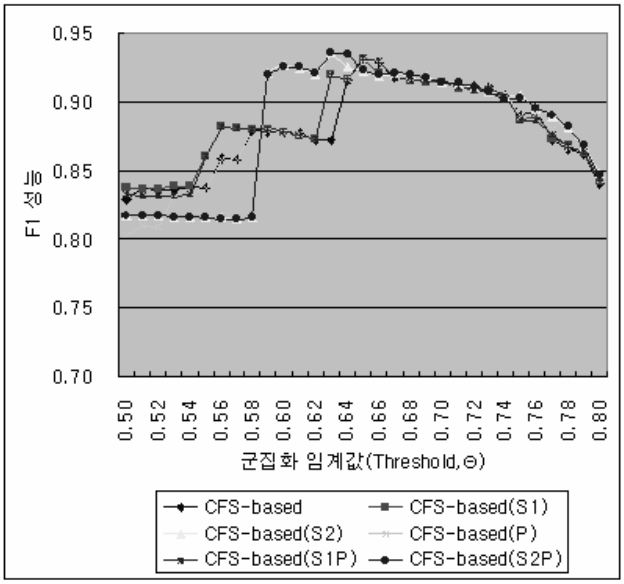

그림 6. 인용 필드 분해의 인용 매칭 성능 결과

그림 5와 6는 CFS-free CM 및 CFS-based CM

의 성능 결과를 그래프로 표현한 것이다. X축은 각 군집화를 시작하는 인용레코드의 임계 유사도 값이며, Y축은 F1의 성능이다. 두 그래프에서 공통적으로 임계값 0.6에서 0.7까지의 성능이 가장 좋은 성능을 보인다. 그림 5의 CFS-free CM의 성능은 불용어 목록 1과 포터스테머를 함께 적용한 결과가 가장 좋은 성능을 보였으며, 그림 4의 CFS-based CM의 성능은 불용어 목록 2와 포터스테머를 함께 적용한 것이 성능이 높았다.

“a, an, the"로 구성된 불용어1을 적용했을 때 성능의 증가 이유는 인용레코드의 제목에서 불용어 처리를 통해 제목의 유사도를 증가시킨 것 때문으로 보인다. 또한 포터스테머는 인용 레코드의 제목에서 발생한 복수형의 “ s ”(예, applications, application) 불일치를 제거하여 인용 레코드 내에 단어의 일치를 증가시켰기 때문에 정확도가 증가하는 경향을 보였다.

표 3은 가용한 모든 필드를 사용한 CFS-based CM의 성능이 CFS-free CM의 성능에 비해 1%내외로 측정된 것을 보여준다. 오류를 동반할 수 있는 자동적인 인용필드분해를 고려한다면 인용 필드 분해가 인용 매칭에 필요한 단계라고 할 수 없을 것이다.

CFS-based CM과 CFS-free CM의 성능이 비슷한 결과를 보인 이유는 인용 필드 분해를 적용하지 않아도 각각의 필드에 사용되는 어휘들이 필드별 종속성을 보이기 때문인 것으로 예상된다. 따라서 표 4와 표 5와 같이 필드의 어휘들에 대한 공유 비율을 산출한다면 타당성을 보여줄 수 있을 것이다.

표 3. 인용 필드 분해의 적용 여부에 따른 인용 매칭의 성능 차이 (F1)

		No	S1	S2	P	S1P	S2P	평균
CFS-free	최대	0.92	0.93	0.92	0.93	0.93	0.92	0.92
	최소	0.79	0.81	0.79	0.79	0.80	0.78	0.79
	평균	0.86	0.87	0.86	0.86	0.88	0.86	0.87
	편차	0.03	0.04	0.04	0.04	0.04	0.04	0.04
CFS-based	최대	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	최소	0.82	0.83	0.81	0.80	0.83	0.81	0.82
	평균	0.88	0.88	0.88	0.87	0.88	0.88	0.88
	편차	0.03	0.02	0.04	0.03	0.03	0.04	0.03

표 4 개별 인용 어휘 필드간 공유 비율 (%)

	A	T	J	P	V	Y	Pa	Add
Author(A)	-	4.87	0.29	1.15	0.29	1.15	0.00	0.57
Title(T)	2.15	-	3.16	8.96	0.13	1.14	0.88	0.76
Journal(J)	1.79	44.64	-	41.07	0.00	0.00	7.14	5.36
Proc(Pa)	1.48	26.20	8.49	-	4.80	2.21	8.49	2.95
Vol(V)	2.44	2.44	0.00	31.71	-	2.44	24.39	2.44
Year(Y)	14.81	33.33	0.00	22.22	3.70	-	3.70	11.11
Page(Pa)	0.00	3.10	1.77	10.18	4.42	0.44	-	3.98
Address (Add)	1.79	5.36	2.68	7.14	0.89	2.68	8.04	-
Publisher (Pub)	8.54	13.41	12.20	19.51	2.44	0.00	3.66	13.41
Editor(E)	23.30	20.39	3.88	9.71	0.97	0.97	3.88	5.83
Month(M)	0.00	0.00	0.00	0.00	0.00	0.00	9.09	0.00
Type(Type)	0.00	12.24	2.04	22.45	10.20	0.00	16.33	2.04
Note(Note)	1.83	26.22	7.32	21.34	3.66	1.22	6.10	3.66
Institution (Ins)	1.52	19.70	10.61	19.70	0.00	1.52	13.64	36.36
Date(Date)	9.30	23.26	0.00	16.28	4.65	37.21	4.65	6.98
Tech(Tech)	0.00	17.50	0.00	12.50	0.00	0.00	15.00	2.50

	Pub	E	M	Type	Note	Ins	Date	Tech
Author(A)	2.01	6.88	0.00	0.00	0.86	0.29	1.15	0.00
Title(T)	1.39	2.65	0.00	0.76	5.43	1.64	1.26	0.88
Journal(J)	17.86	7.14	0.00	1.79	21.43	12.50	0.00	0.00
Proc(P)	5.90	3.69	0.00	4.06	12.92	4.80	2.58	1.85
Vol(V)	4.88	2.44	0.00	12.20	14.63	0.00	4.88	0.00
Year(Y)	0.00	3.70	0.00	0.00	7.41	3.70	59.26	0.00
Page(Pa)	1.33	1.77	0.44	3.54	4.42	3.98	0.88	2.65
Address (Add)	9.82	5.36	0.00	0.89	5.36	21.43	2.68	0.89
Publisher (Pub)	-	14.63	0.00	1.22	17.07	6.10	0.00	0.00
Editor(E)	11.65	-	0.00	0.97	6.80	2.91	0.97	0.00
Month(M)	0.00	0.00	-	0.00	18.18	0.00	63.64	0.00
Type(Type)	2.04	2.04	0.00	-	4.08	10.20	4.08	24.49
Note(Note)	8.54	4.27	1.22	1.22	-	3.66	2.44	1.22
Institution (Ins)	7.58	4.55	0.00	7.58	9.09	-	1.52	19.70
Date(Date)	0.00	2.33	16.28	4.65	9.30	2.33	-	0.00
Tech(Tech)	0.00	0.00	0.00	30.00	5.00	32.50	0.00	-

표 5. 개별 인용 어휘 필드간 공유 비율 (다른 필드 집합)

	필드 어휘 개수	다른 필드 집합과 중복 어휘 개수	다른 필드와 공유 어휘 비율
Author	6189	146	2.36
Title	11885	547	4.6
Journal	1398	120	8.58
Proc	6430	398	6.19
Vol	339	0	0
Year	1198	11	0.92
Page	3017	14	0.46
Address	1871	26	1.39
Publisher	1475	13	0.88
Editor	1797	105	5.84
Month	32	0	0
Type	420	4	0.95

Note	519	66	12.72
Institution	1321	80	6.06
Date	511	34	6.65
Tech	804	1	0.12
평균			3.60

표 4는 개별 인용 레코드에서 필드별 유일한 어휘들을 수집한 후, 다른 필드의 어휘들과 겹치는 비율을 계산한 결과를 보여준다. 평균 0.25%미만의 어휘 중복 비율을 보이기 때문에 인용 필드 분해를 하지 않아도 군집화의 성능이 큰 차이가 없었던 이유라고 볼 수 있다.

표 5는 개별 필드와 다른 필드 간의 개별 인용 어휘 중복 비율을 보여준다. 저자 필드(Author)는 6189개의 인용 레코드 레벨의 개별 어휘를 가지고 있으며, 저자필드를 제외한 모든 필드의 어휘와 중복이 되는 비율이 2.36%로 상당히 낮은 것을 보여준다.

표 6. 중복 어휘 비율 요약

	A	T	J	P	V	Y	Pa	Add
개별중복비율 (어휘)	0.49	1.62	0.08	1.06	0	0.01	0.06	0.16
개별중복비율 (타필드)	2.36	4.6	8.58	6.19	0	0.92	0.46	1.39
어휘집합중복비율	4.6	16.84	3.49	16.26	2.41	3.4	8.33	6.53

	Pub	E	M	Type	Note	Ins	Date	Tech
개별중복비율 (어휘)	0.04	0.17	0	0.04	0.14	0.15	0.05	0.01
개별중복비율 (타필드)	0.88	5.84	0	0.95	12.72	6.06	6.65	0.12
어휘집합중복비율	4.87	4.1	1.2	4.59	9.47	7.07	9.69	3.45

	평균
개별중복비율 (어휘)	0.26
개별중복비율 (타필드)	3.60
어휘집합중복비율	6.64

표 6은 중복 어휘 비율을 요약한 것이다. 개별 인용 어휘의 비율(표 4)와 다른 필드 집합과 중복 어휘 비율(표 5)와 인용 필드별 어휘 집합 간의 비율을 요약한 것이다. 전체 어휘 집합 대비 중복 비율은 필드별 어휘를 집합으로 생성한 후에 중복 비율을 계산한 것이다. 평균 6.64% 정도의 어휘 중복 비율을 보인다

그림 8은 표 4의 비율을 그래프로 보여준다. 실질적인 인용 레코드의 필드 별 중복을 보여주는 개별 필드 별 중복 비율이 가장 낮은 비율로 고르게 분포되어 있는 것을 볼 수 있다.

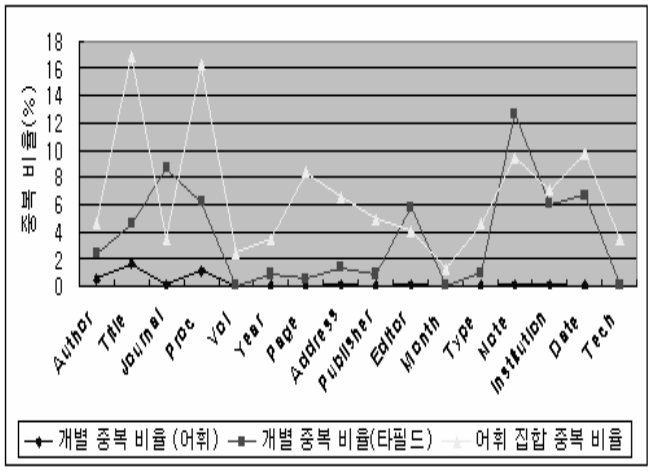

그림 8. 중복 어휘 비율 비교 그래프 (%)

6. 결론

본 논문은 인용 매칭에서 인용 필드 분해의 유용성 여부를 알아보기 위해, 인용 매칭을 수행하고자 할 때, 인용 필드 분해의 영향을 분석하였다.

필드 분해를 사용한 인용 매칭방법의 성능이 다소 높았지만, 인용 필드 분해를 하지 않은 인용매칭의 결과와는 크게 차이가 나지 않았다. 그 이유는 필드 별 어휘가 거의 중복되지 않아서 필드 별로 질의를 하는 결과와 유사한 결과가 생성되었기 때문이다. 따라서 현재까지 알려진 자동적인 인용 필드 분해의 성능을 고려한다면, 오히려 필드 분해를 사용하지 않는 것이 더 효과적일 것이다. 추후에는 한국어로 테스트 셋을 구축하여 이를 적용해 볼 계획이다. 또한 다양한 군집화 방법을 적용하여 이를 적용해야 할 것이다.

인용 매칭은, 하나의 실체를 가진 여러 형태적 변이형을 어떻게 형태적 군집화를 통해 이를 하나의 실체를 갖는 변이형인가를 판별하여 같은 인용으로 묶어 내는가에 대한 문제이다. 인용의 대상이 주로 논문 등을 대상으로 수집되고 있으나, 오늘날에는 웹저널이나 블로그 등 다양한 매체에서의 인용이 발생하기 때문에 자동적인 인용 매칭의 방법은 통합적인 인용을 측정할 수 있는 방법으로도 새로운 의미를 가질 수 있을 것이다. 더욱이, 문헌의 인용이 잘 구성이 된다면, 저자 별로 인용을 측정해 저자의 전문성 평가나 공저자 네트워크에 인용을 반영하는 등의 다양한 연구에 적용 가능할 것이다.

참고 문헌

[1] Andrew McCallum, Kamal Nigam, Jason Rennie, Kristie Seymore, Automating the Construction of Internet Portals with Machine Learning. Information Retrieval. 3(2). 2000.
[2] Andrew McCallum, Kamal Nigam, Lyle Ungar, Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching. In Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000). 2000.
[3] Andrew McCallum and Ben Wellner, Toward Conditional Models of Identity Uncertainty with Application to Proper Noun Coreference. IJCAI Workshop on Information Integration on the Web, 2003
[4] Andrew McCallum, Information extraction: distilling structured data from unstructured text, Social Computing, 3(9), 2005.
[5] Min-Yuh Day, Richard Tzong-Han Tsai, Cheng-Lung Sung, Chiu-Chen Hsieh, Cheng-Wei Lee, Shih-Hung Wu, Kun-Pin Wu, Chorng-Shyong Ong, Wen-Lian Hsu, Reference metadata extraction using a hierarchical knowledge representation framework, Decision Support Systems archive, 43(1), 2007
[6] Dominique Besagni and Abdel Belaïd, Citation recognition for scientific publications in digital libraries, Document Image Analysis for Libraries, Proceedings. First International Workshop on, 2004.
[7] Fuchun Peng and Andrew McCallum, Accurate information extraction from research papers using conditional random fields. In Proceedings of Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics annual meeting, 2004.
[8] Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, Ilya Shpitser. Identity Uncertainty and Citation Matching, Advances in Neural Information Processing Systems 15 (NIPS 2002). 2003.
[9] Hui Han, C.L.Giles, E. Manavoglu, Zha Hongyuan, Zhang Zhenyue, Edward A. Fox, Automatic document metadata extraction using support vector machines, Proceedings. 2003 Joint Conference on Digital Libraries, 2003.
[10] I-Ane Huang, Jan-Ming Ho, Hung-Yu Kao, Wen-Chang Lin, Extracting Citation Metadata from Online Publication Lists Using BLAST, Proc. of the Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-04), 2004.
[11] Imran R. Mansuri, Sunita Sarawagi, Integrating Unstructured Data into Relational Databases, Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), 2006.
[12] Isaac G. Councill, Huajing Li, Ziming Zhuang, Sandip Debnath, Levent Bolelli, Wang-Chien Lee, Anand Sivasubramaniam, C. Lee Giles, "Learning metadata from the evidence in an on-line citation matching scheme," Joint Conference on Digital Libraries 2006 (JCDL 2006), 2006.
[13] Mikhail Bilenko and Raymond J. Mooney, Adaptive Duplicate Detection Using Learnable String Similarity Measures, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003), Washington DC, August 2003.
[14] Steve Lawrence, C. Lee Giles, Kurt Bollacker, Digital libraries and autonomous citation indexing. IEEE Computer, 32(6), 1999.
[15] Steve Lawrence, C. Lee Giles, Kurt D. Bollacker, Autonomous citation matching, Proceedings of the third annual conference on Autonomous Agents, Seattle, Washington, United States , 1999.
[16] Stevan Harnad and Leslie Carr, Integrating, Navigating and Analyzing Eprint Archives Through Open Citation Linking (the OpCit Project). Current Science 79(5), 2000.
[17] Sunita Sarawagi, V. G. Vinod Vydiswaran, Sumana Srinivasan, Kapil Bhudhia: Resolving citations in a paper repository. SIGKDD Explorations 5(2), 2003.
[18] Wei Li and Andrew McCallum, A Note on Semi-supervised Learning using Markov Random Fields. Technical Note, 2004.
[19] William W. Cohen, Pradeep Ravikumar, Stephen Fienberg, A Comparison of String Metrics for Matching Names and Records, KDD Workshop on Data Cleaning and Object Consolidation, 2003.