개념간 관계의 추출과 명명을 위한 통계적 접근방법

김 희 수[†] · 최 익 규[††] · 김 민 구[†††]

요 약

온톨로지는 차세대 시맨틱 웹을 위한 논리의 기반을 구성하기 위해 제안되었다. 이러한 온톨로지는 특정 분야에 대한 지식을 정형화된 형태로 표현함으로써 기계에 의한 지식의 이해를 가능하게 하고, 이를 사용하여 사용자의 요구에 알맞은 지능화된 서비스를 제공할 수 있게 한다. 하지만, 온톨로지의 구축과 유지는 많은 사람의 시간과 노력을 요구한다. 본 고에서는 온톨로지 구축 방법의 일환으로, 문서로부터 온톨로지를 구성하는 개념간의 관계를 정의하는 자동화된 방법을 제안한다. 본 고에서 제안된 방법은 특정 분야의 문서에 존재하는 개념을 기반으로 개념간의 연관 규칙을 형성하는 개념 쌍을 찾고, 두 개념 사이에 존재하는 내용의 군집화를 통해 두 개념간의 관계를 설명하는 패턴을 찾는다. 마지막으로 패턴간의 군집화를 사용하여 개념 사이의 일반화된 관계를 명시한다. 본 고에서는 제안된 방법을 검증하기 위한 방법으로 TREC(Text REtrieval Conference)에서 제공하는 문서집합을 사용하여 개념간의 관계를 추출, 평가하였으며, 그 결과 제안된 방법은 개념간의 관계를 설명하는 유용한 정보를 제공할 수 있음을 보여준다.

키워드 : 온톨로지 자동 구축, 정보 추출, 관계 추출

A Statistical Approach for Extracting and Naming Relation between Concepts

Hee-soo Kim[†] · Ikkyu Choi[††] · Minkoo Kim[†††]

ABSTRACT

The ontology was proposed to construct the logical basis of semantic web. Ontology represents domain knowledge in the formal form and it enables that machine understand domain knowledge and provide appropriate intelligent service for user request. However, the construction and the maintenance of ontology requires large amount of cost and human efforts. This paper proposes an automatic ontology construction method for defining relation between concepts in the documents. The Proposed method works as following steps. First we find concept pairs which compose association rule based on the concepts in domain specific documents. Next, we find pattern that describes the relation between concepts by clustering the context between two concepts composing association rule. Last, find generalized pattern name by clustering the clustered patterns. To verify the proposed method, we extract relation between concepts and evaluate the result using documents set provide by TREC(Text Retrieval Conference). The result shows that proposed method cant provide useful information that describes relation between concepts.

Key Words : Automatic Ontology Construction, Information Extraction, Relation Extraction

1. 서 론

인터넷과 전자 문서 기술의 발전은 웹이라는 거대한 정보의 저장소를 만들었다. 하지만 웹은 단순히 정보의 저장을 위한 공간의 개념을 뛰어넘어 사람과 사람간의 의사소통을 위한 중요한 수단으로 사용되고 있으며, 더 나아가서 개인과 기업, 기업과 기업간의 정보의 사용 및 공유를 통해 이익을 창출하는 기능을 담당하게 되었다. 이 과정에서 검색엔진을 비롯한 정보 처리 시스템들은 사용자의 요구를 효율적으로 처리하기 위해 사용되었다. 하지만 웹 상에 존재하는 대부분의 정보는 사람의 직관적 이해를 중심으로 작성되었으며, 이것은 정보 처리 시스템이 사용자의 요청을 처리함에 있어서 한계를 드러나게 하는 원인이 되었다. 이를 해결하기 위한 방법으로 컴퓨터가 이해할 수 있는 정형화된 형태로 정보를 표현하고자 하는 시맨틱 웹(Semantic Web)이 제안되었다[10].

시맨틱 웹의 논리적 기반이 되는 온톨로지(ontology)는 "특정 분야에 대한 공유될 수 있는 개념들의 정형화된 기술"로 정의되며[3], 개념(concept)은 그것의 특징을 설명하기 위한 속성(property)의 집합으로 표현된다. 또한, 속성은 그것이 적용되는 범위를 표현하기 위한 제약(facet)을 갖는다[11]. 온톨로지는 RDF와 DAML+OIL, OWL 등의 정형화된(formal) 언어를 사용하여 특정 분야 또는 세상에 존재하는

※ 본 연구는 21세기 프론티어 연구 개발 사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스 컴퓨팅 및 네트워크 원천기반 기술사업이 지원과, 과학기술부의 국가지정연구실 사업의 일환으로 지원받아 수행되었음. (과제번호 : M10302000087-03J0000-04400)
† 준 회 원 : (주)엔텔리아 프로토콜 1팀 연구원
†† 준 회 원 : 아주대학교 프로그래밍 전문강사
††† 정 회 원 : 아주대학교 컴퓨터공학과 교수
논문접수 : 2005년 3월 9일, 심사완료 : 2005년 6월 1일

개념들을 표현함으로써, 기계가 지식 또는 정보를 이해할 수 있도록 한다. 그러나 대부분의 온톨로지는 특정 분야의 전문가 혹은 지식 공학자에 의해서 만들어지기 때문에, 온톨로지의 구축에는 많은 시간과 비용이 요구된다. 게다가, 이미 구축된 온톨로지는 시간의 흐름에 따라 새로운 개념의 등장 및 기존 개념의 변화 등을 반영해야 하며, 이는 온톨로지의 유지 비용을 증가시키는 원인이 된다. 따라서 온톨로지의 구축과 유지를 위한 비용을 감소하기 위한 방법으로써, 문서 집합으로부터 온톨로지를 자동으로 구축하고자 하는 많은 연구들이 수행되었다[1, 7-9, 14].

본 고에서는 온톨로지 구축 과정 중, 통계적 접근 방법을 통한 자동화된 개념간의 관계 정의 방법을 제안한다. 본 고의 구성은 다음과 같다. 제 2장에서는 온톨로지 자동 구축 방법 및 이와 관련된 연구에 대해서 살펴보고, 제 3장에서는 온톨로지 자동 구축에 관련된 개념간의 관계 추출 방법을 제안한다. 제 4장에서는 실험 결과를 통해 제안한 방법을 평가하고, 실험 시 발생하는 문제점에 대해 분석한다.

2. 관련연구

온톨로지 구축 과정은 일반적으로 다음의 네 과정을 포함한다. 먼저, 개념을 정의하고, 두 번째로 개념 사이의 상-하위 관계를 밝힌다. 세 번째로 속성의 정의 및 속성에 허락되는 값을 기술하고, 마지막으로 개념을 구체화하는 개체를 생성한다[11]. 본 장에서는 온톨로지 구축 과정에서 개념간의 상-하위 관계와 속성을 정의하는 단계에 속하는, 개념간의 관계를 발견하기 위해 기존 연구에서 사용된 방법들을 언어적 접근 방법과 통계적 접근 방법으로 분류하고, 각 연구의 구체적 방법을 설명한다.

문서에는 그 문서를 작성한 사람의 지식이 반영되어 있으며, 다수의 문서 집합은 특정 개인의 지식을 반영함과 동시에 일반적인 지식을 반영한다. 따라서 문서 집합으로부터 지식을 발견하고자 하는 연구가 수행되어 왔다[1, 7-9, 14]. 문서로부터 지식을 추출하거나 구축하기 위한 방법은 문서를 다루는 접근 방법에 따라 크게 두 가지로 구분될 수 있다. 첫 번째 접근 방법은 문서에 사용된 글의 언어적 특성을 사용하는 것이다[1, 14]. 두 번째 접근 방법은 문서에 사용된 단어의 빈도 등과 같이 문서 집합의 통계적 특성을 사용하는 것이다[6-9, 12-13].

2.1 언어적 접근 방법

문서로부터 지식을 구축하기 위한 언어적 접근 방법은 문장의 문법 및 특정 의미를 갖는 패턴과 같은 문법적인 요소를 사용한다. 예를 들어, 문서 상에 'dogs and other animals'와 같은 어구에서 'dog'는 'animal'의 하위 개념을 나타낼 것이라는 가정하에, '*A*s and other *B*s'라는 일반화된 패턴으로 만든다. 그리고 이 패턴을 사용하여 문서로부터 얻어진 *A*와 *B*의 개체들 간의 상-하위 관계를 찾는다.

Hearst의 연구에서는 (그림 1)에 보여진 것과 같이 문서

NP_0 such as $\{NP, NP, \cdots, (\text{and}\mid\text{or})\}$ NP → hyponym(NPs, NP_0)

such NP_0 as {NP,}* {(or | and)} NP → hyponym(NPs, NP_0)

NP {, NP}* {,} or other NP_0 → hyponym(NPs, NP_0)

NP {, NP}* {,} and other NP_0 → hyponym(NPs, NP_0)

NP_0 {,} including $\{NP,\}$* {or | and} NP → hyponym(NPs, NP_0)

NP_0 {,} especially $\{NP,\}$* {or | and} NP → hyponym(NPs, NP_0)

(그림 1) 개념간의 하위관계를 위한 언어적 패턴

로부터 하위관계를 추출하기 위한 여섯 개의 패턴을 정의하고 그 결과를 WordNet과 통합하는 과정을 수행했다[1]. 하지만 이러한 패턴에 의한 관계의 추출은 언어 사용 방법의 다양성으로 인해 기대하는 관계를 발견하기 어렵다는 단점이 있다.

문서에 등장하는 패턴을 사용하는 방법과는 다른 언어적 접근 방법은 자연어 처리 파서(parser)를 사용하여 문서의 원본에 문법 정보를 추가하고 이것을 사용하여 지식을 구축하는 것이다. Cimiano의 연구에서는 개념의 분류체계를 표현하기 위해 자연어 처리 파서를 사용하여 개념과 속성을 추출하고, FCA(formal concept analysis)를 사용하여 분류체계를 구축한다[14]. 하지만 Cimiano가 제안한 방법은 문법 정보, 즉 목적어와 동사를 각각 개념과 속성으로 단순 사상시킴으로써, 많은 개념에 비해 그 개념을 설명하는 속성이 제한적이며, 이러한 방법을 통해 얻어진 분류체계는 낮은 정확도를 갖는다는 단점이 있다[16]. 이것은 하나의 동사가 여러 가지의 의미를 갖는데 반해 문서로부터 추출된 동사만으로 개념의 속성을 정확하게 표현하기 어렵다는 사실에 기인한 문제점으로 생각할 수 있다. 또한, 위 방법은 소수의 문서가 특정 분야에 대해 정확하고 객관적으로 기술되어 있으면 좋은 결과를 보일 수 있으나, 검증되지 않은 다수의 문서를 대상으로 했을 경우에는 좋은 결과를 기대하기 어렵다.

2.2 통계적 접근 방법

통계적 접근 방법은 문서 집합이 갖는 통계적인 분포를 기반으로 개념과 관계를 추출하고 그것을 기반으로 지식을 구축한다. 문서 집합이 갖는 통계적인 분포에는 단어의 빈도 또는 문서의 빈도, 단어 사이의 종속성[6] 등이 있으며, 단어-가중치 방법(term-weighting scheme)을 적용하여 이러한 통계적 분포를 가공하고, 군집화 (clustering) 및 데이터 마이닝(data mining) 기법을 통해 지식을 발견한다.

문서로부터 지식을 얻기 위한 방법의 일환으로 정보 검색(information retrieval) 분야에서는 개념간의 분류체계를 구성하기 위한 연구가 이루어져왔다[6, 9, 12-13]. 이러한 연구들은 개념을 정의하는 방법에 따라 다시 두 가지로 분류할 수 있다. 한 가지는 개념을 단어들의 집합으로 간주하여 분류체계를 구축하기 위해서 계층적 군집화(hierarchical clustering)를 이용하는 것이다. 하지만 계층적 군집화를 사용한 방법은 형성된 군집(cluster)에 대한 명명(naming) 문제와 사용자에게 직관적인 정보를 제공하지 못하는 문제가 있다[9].

다른 한 가지는 문서에 존재하는 어구(phrase)를 개념으로 간주하고 어구간의 종속성 및 통계적 특성을 사용하여 분류 체계를 구축하는 것이다[6, 12-13]. 하지만 이러한 방법들은 근본적으로 사용자의 정보 접근의 효율성을 향상시키기 위한 목적으로 제안되었기 때문에, 구축된 분류체계를 엄밀한 의미에서 상-하위 관계로 간주하기 어렵다.

Snowball 시스템은 문서로부터 특정 관계를 추출하기 위한 방법을 제시한다[8]. 이 시스템에서는 문서 집합에서 중요한 관계로 고려되는 개념 쌍의 예들을 사용하여, 그 개념 쌍이 형성하고 있는 관계와 동일한 관계를 갖고 있는 새로운 개념 쌍을 찾는다. Snowball 시스템의 결과는 문서로부터 특정 관계를 구성하는 개념 쌍들과 그것을 연결하는 관계에 관련된 패턴들로, 이것은 온톨로지를 구축에 있어서 중요한 정보로 사용될 수 있다. 하지만 이 시스템을 사용하기 위해서는 관계를 형성하는 대표 개념 쌍들의 선출과정이 요구되며, 이 개념 쌍들의 정확도는 시스템의 성능에 크게 영향을 준다.

Karlsruhe 대학과 FZI 연구 센터에서 개발한 KAON 시스템은 문서로부터 온톨로지를 생성을 돕기 위한 도구(tool)로써, 통계적 접근 방법을 사용하여 온톨로지의 개념간의 관계를 발견한다[7]. KAON 시스템에서 온톨로지를 구축하는 방법은 다음과 같다. 먼저 저 수준의 파서(shallow parser)를 사용하여 문서를 처리하고, 개념으로 사용될 하나의 단어 이상을 포함하는 명사절을 추출한다. 그 다음 선택된 개념간의 일반화된 연관 규칙(generalized association rule)[5]을 찾아 개념간의 관계의 가능성을 제시한다. KAON시스템은 개념간의 관계를 직접적으로 추출하기보다 개념간의 관계로 가능성이 있는 후보들을 제시하고 연관 규칙의 지지도(support) 및 신뢰도(confidence)를 그 관계의 이름으로 제공함으로써 사용자에게 추출된 개념간의 관계를 온톨로지의 관계로 사용할 것인가에 대한 판단과 그 관계의 이름에 대한 명명(naming)을 맡긴다. 이 시스템을 이용한 온톨로지의 생성 방법은 온톨로지의 생성과 유지하는데 있어서 상당 부분에 사람의 참여를 요구하며, 추출된 개념간의 관계를 설명하기 위한 직관적인 정보를 제공하지 않는다.

앞서 살펴본 각 방법론의 주요 문제점은 다음과 같다. 언어적 접근 방법 중 하나인, 패턴에 의한 개념의 추출 방법의 단점은 특정 관계와 관련된 패턴을 발견하는 것이 쉽지 않고, 패턴을 찾았다 하더라도 언어의 특성상 많은 예외가 존재한다는 것이다. 또한, 통계적 접근 방법의 단점은 추출된 관계를 설명하기 위한 직관적인 정보를 제공하기 어렵다는 것이다. 본 고에서는 통계적 접근 방법을 사용하여 문서 집합으로부터 개념간의 관계를 추출하고 관계의 이름을 부여하는 방법을 제안함으로써, 앞서 언급된 관련 연구의 단점을 보완한다.

3. 개념간의 관계 추출

온톨로지 구축 과정 중, 개념의 속성을 정의하는 단계에서는 개념간의 관계를 찾고, 그 관계를 설명하는 속성들로 정의한다. 이것은 온톨로지의 구축 과정에서 특정 분야에 존재하는 모든 개념 사이의 관계를 기술하는 부분으로, 온톨로지의 사용 범위와 목적, 개념들에 대한 정확한 이해를 요구하며, 관계를 발견하고 정의하기 위해서 많은 시간과 노력이 요구된다. 제안된 방법은 특정 분야에 대한 충분히 큰 문서 집합을 대상으로 통계적 분석을 사용하여 문서 집합에 존재하는 개념간의 관계를 추출하고, 추출된 관계에 명명과정을 수행하여 그 결과를 제시함으로써 온톨로지의 구축 과정을 돕는다.

본 고에서 제안하는 관계 추출 방법은 "두 개의 개념간의 관계는 동일 문장에 등장하는 두 개념을 연결하는 내용(context)으로 설명된다"라는 가정을 기반으로 한다. 간단한 예로, "John is father of Bill."이라는 문장을 생각해보자. 이 문장은 사람과 사람간의 관계인 'father-of'관계를 표현하고 있다. 문서에 존재하는 문장으로부터 중요한 개념쌍을 찾기 위한 방법으로 KAON 시스템에서 사용된 연관 규칙(association rule)을 사용한다[7]. 연관 규칙 '$A \rightarrow B$'는 '개념 A가 문장에서 사용될 때, 개념 B도 같이 사용된다'는 것을 의미하며, 연관 규칙을 위해 계산되는 지지도(support)와 신뢰도(confidence)에 임계값을 적용하여 연관 규칙으로 생성되는 개념 쌍을 제한한다. 다음 수식은 연관 규칙이 갖는 지지도(support)와 신뢰도(confidence)를 나타낸다[4]. 지지도는 두 개의 개념 쌍이 동일 문장에 나타날 확률을, 신뢰도는 하나의 개념이 다른 특정 개념을 수반하는 확률을 나타낸다. 제안된 방법에서는 문장에 포함된 모든 개념 쌍과 개념 쌍 사이에 존재하는 단어들을 내용(context)으로 추출하고, 추출된 개념 쌍에 대한 연관 규칙을 찾는다.

$$Support(A \rightarrow B) = P(A \wedge B)$$

$$Confidence(A \rightarrow B) = \frac{P(A \wedge B)}{P(A)}$$

연관 규칙을 구성하는 두 개념 사이에 존재하는 내용은 개념 쌍이 형성하는 관계를 설명하기 위해 사용된다. 그러나, 각각의 내용은 그 표현을 사용한 특정 개인의 지식을 반영한 것이므로, 일종의 동의가 필요하다. 이를 위해서 내용의 군집화를 수행한다. 군집화의 결과로 생성된 군집에 포함된 내용의 수는 그 군집에 대한 동의를 나타내며, 군집화 과정을 통해 발견된 군집의 중심(centroid)은 두 개의 개념 사이에 존재할 수 있는 어휘적 패턴으로 간주될 수 있다. 하지만, 각 군집은 군집에 속한 내용의 수와 군집의 응집도가 다양하기 때문에, 좋은 군집의 선택이 필요하다. 따라서, 어휘적 패턴의 선택을 위해 포함된 내용의 수와 군집의 응집도에 임계값을 적용하여 패턴을 선택한다.

그러나 이러한 과정의 결과로 얻어진 패턴은 사용자에게 직관적인 설명을 제공하지 못한다는 군집화의 문제점을 가지고 있기 때문에, 사용자에게 익숙하게 재구성될 필요가 있다. 제안된 방법에서는 사용자에게 직관적인 정보를 제공

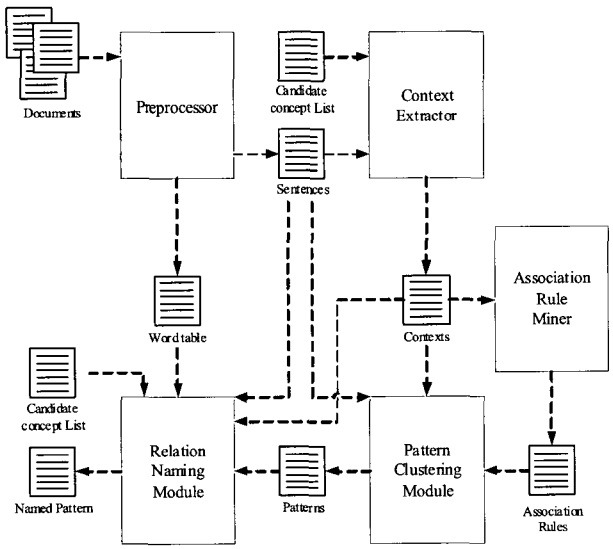
하기 위한 방법으로, 추출된 패턴 즉, 개념간의 관계에 이름을 부여함으로써 사용자에게 직관적인 정보를 제공한다. 또한, 각 군집은 특정 개념 쌍 사이에 존재하는 관계만을 설명하기 때문에, 많은 수의 불필요한 패턴을 생성한다. 따라서, 의미 있는 패턴을 찾기 위한 방법으로 특정 개념간의 관계를 나타내는 패턴의 일반화가 요구된다. 제안된 방법에서는 일반화된 패턴을 찾기 위해 재 군집화 과정을 수행하여 일반화된 패턴을 발견한다. 또한, 이 때 생성되는 군집의 중심을 사용하여 군집에 속한 각 내용의 가중치를 계산하고, 가장 높은 가중치를 갖는 내용을 개념간의 관계의 이름으로 결정한다. 다음 수식은 내용(cx)의 가중치를 계산하는 방법을 설명한다.

$$score(cx) = \frac{\sum_{w_i = i\text{th word of } cx} w_i.weight}{\text{length of } cx}$$

(단, $w_i weight$는 cx가 속한 군집의 중심에 있는 w_i의 가중치)

이 수식은 내용의 가중치가 일반화된 패턴을 구성하는 단어들의 가중치에 의해 결정되는 것을 의미하며, 내용에 포함된 단어의 수에 의해 내용의 점수가 받는 영향을 줄이기 위한 방법으로 일정 길이에 대한 가중치를 계산한다.

문서 집합으로부터 온톨로지의 개념들 사이의 관계를 추출하기 위해 제안된 방법을 구현한 REX(Relation EXtractor) 시스템의 구조는 (그림 2)와 같다.



(그림 2) REX 시스템의 구조

이 시스템은 개념간의 관계를 추출하기 위해 다섯 단계의 과정을 거친다. 먼저 전처리기(Preprocessor)에서는 많은 양의 문서를 효율적으로 다루기 위해서 문서 집합을 문장 단위로 구분하고, 각 문장을 단어의 ID들의 연속(sequence)된 형태로 표현한다. 다음 단계인 내용 추출기(Context Extractor)에서는 입력된 개념 목록으로부터 개념을 단어의 연속된 형

```
Preprocessor()
{
  initialize B+-tree;
  while ((token = gettoken()) != EOF) {
    if token.type is WORD then{
      word = kstem(token);
      find word from B+-tree;
      if word does not exist in B+-tree then {
        add word into B+-tree;
        add word ID into sentence;
        }
      else if token.type is END_OF_SENTENCE) then {
        write sentence into output file;
        initialize sentence;
      }
    }
  }
}
ContextExtractor()
{
  make concept_index_structures using concepts;
  for each sentence s do
    for i-th word wi of s do {
      if there is a concept c1 which starts with wi and matches with sentence then{
        for j -th word wj of s do
          if there is a concept c2 which starts with wj and matches with sentence then{
              write c1, c2, and context;
            j = j + c2.length;
          }
        i = i + c1.length;
        continue ;
      }
}
AssociationRuleMiner()
{
  for each concept ci do
    for each concept cj, ci ? cj do {
      find contexts cxs which contain ci and cj;
      support = number of cxs / total number of contexts;
      confidence = number of cxs / number of contexts containing ci;
      if (support > minsup AND confidence > minconf) then
        write ci? cj, support, confidence and its cxs;
    }
}
PatternClusteringModule()
{
  for each association rule ar do {
    ClusterSet CS = {};
    for each context cx of ar do {
      convert cx into vector v
      if CS is empty then {
        make new cluster c with v;
        add v into CS;
      }
      find bestCluster bc matched with v using threshold;
      if bc exists then
        add v into bc;
      else {
        make new cluster c with v;
        add v into CS;
        }
    }
    write CS;
  }
}
RelationNamingModule()
{
  ClusterSet CS = Cluster all clusters generated in procedure ClusterContext();
  for each cluster c in CS do {
    bestScore = 0;
    for each context cx of c do {
      calculate cx.score using centroid of c;
      if cx.score > bestScore then
        bestContext = cx;
    }
    write concepts of association rules of c and bestContext;
  }
}
```

(그림 3) REX 시스템의 모듈별 알고리즘

태로 표현하고, 전처리기에서 생성한 각 문장 중, 두 개의 개념 쌍이 존재하는 문장에 대해서 두 개념 사이의 존재하는 단어들의 연속인 내용(context)을 추출한 후, 개념 쌍과 내용을 출력한다. 연관 규칙 생성기(Association Rule Miner)에서는 내용 추출기에서 추출된 내용으로부터 연관 규칙을 생성한다. 네 번째 단계인 패턴 군집화 모듈(Pattern Clustering Module)에서는 각 연관 규칙에 대한 내용의 군집화를 수행하고, 그 결과로 생성된 패턴의 재 군집화를 통해 일반화된 패턴을 찾는다. REX 시스템에서는 방대한 양의 내용에 대한 군집화의 속도를 향상시키기 위해 simple single-pass clustering 알고리즘[2]을 사용한다. 마지막으로 관계 명명 모듈(Relation Naming Module)에서는 일반화된 패턴에 속한 내용의 가중치 계산 과정을 통해 관계에 이름을 붙인다. (그림 3)은 각 단계에서 수행되는 알고리즘을 설명한다.

4. 실험 및 평가

제안된 방법을 검증하기 위해 사용된 실험 데이터는 TREC(Text REtrieval Conference)에서 제공하는 Ziff 문서 집합(Information from Computer Select disks-1989, 1990, copyrighted by Ziff Davis)으로, 컴퓨터와 관련된 기사들의 요약을 담고 있다. Ziff 문서 집합은 약 800MB 크기의 집합으로, 본 고에서 제안한 방법을 구현한 REX 시스템의 전처리 결과로 4,170,525개의 문장과 483,102 종류의 단어가 추출되었다.

4.1 개념의 선택

REX 시스템은 문서 집합과 그 문서 집합에 나타나는 개념들을 기반으로 개념간의 관계를 추출하기 때문에, 제안된 시스템은 미리 정의된 개념들을 필요로 한다. 이 실험에서 개념은 문서 집합에 나타나는 연속된 단어로 정의하였으며, Ziff 문서 집합에 "DESCRIPT" 태그에 의해 기술된 회사와 제품의 이름 중 출현빈도가 높은 3,800여 개를 개념으로 선택하였다. <표 1>은 선택된 개념들의 예를 보여준다.

〈표 1〉 문서로부터 선택된 개념의 예

개 념	개 념
IBM Corp	SQL Server
Microsoft	IBM PS/2
Apple Computer	dBASE IV
Apple Macintosh	Nantucket
Digital Equipment	OS/2 2.0

4.2 매개 변수

REX 시스템에서는 관계의 추출을 위해 7가지의 매개 변수를 제안한다. 먼저, 내용의 최대 길이(maximum length of context)는 내용 추출기에서 추출되는 내용에 포함된 단어의 길이를 제한한다. 이것은 문장에서 두 개념 사이의 거리가 특정 값보다 크면, 두 개념간의 관계는 고려 대상에서 제외됨을 의미한다. 최소 지지도(minsup)와 최소 신뢰도(minconf)는 연관 규칙 생성 알고리즘에서 사용되는 값으로, 중요한 관계를 형성하는 개념 쌍들을 선택하기 위해 사용된다[4]. 패턴 군집화 모듈에서 사용되는 내용간의 최소 유사도(minimum similarity between contexts)는 내용들을 군집화하는 과정에서 새로운 군집을 생성할 것인지, 아니면 가장 가까운 군집에 포함할 것인지를 결정하기 위해 사용된다[2]. 관계 명명 모듈에서 남은 세 개의 매개 변수가 사용되는데, 군집에 속한 내용의 최소 수(minimum number of contexts in a cluster)와 군집에 있는 내용간의 최소 평균 유사도(minimum average similarity between contexts in a cluster)는 패턴 군집화 과정에서 생성된 군집들을 여과(filtering)하기 위해 사용된다. 즉, 군집의 중심이 패턴으로 간주되기 위해서는 적정 수의 내용들을 포함하고 있어야 하며, 그 중심과 내용들 사이의 유사도가 만족할 만한 수준이 되어야 한다는 것이다. 마지막으로 패턴들간의 최소 유사도(minimum similarity between patterns)는 패턴 군집화 과정에서 패턴들로부터 일반화된 패턴을 찾기 위한 군집화 단계에서 패턴이 군집에 포함되기 위한 조건으로 사용된다. <표 2>는 REX 시스템에서 사용하는 각 매개 변수에 대한 요약이며, <표 3>은 매개 변수에 따른 실험 결과의 변화를 보여준다.

〈표 2〉 REX 시스템의 매개 변수들

매개 변수	설 명
maximum length of context	내용 추출기에서 추출되는 내용의 최대 길이
Minsup	연관규칙이 생성되기 위한 최소 지지도
Minconf	연관규칙이 생성되기 위한 최소 신뢰도
minimum similarity between contexts	내용의 군집화를 위한 최소 유사도
minimum number of contexts in a cluster	군집의 여과에 사용되는 군집에 속한 내용의 수
minimum average similarity between contexts in a cluster	군집의 여과에 사용되는 군집에 속한 내용과 군집의 중심과의 평균 유사도
minimum similarity between patterns	패턴의 군집화를 위한 최소 유사도

〈표 3〉 매개 변수들에 따른 결과의 변화

매개 변수	범 위	변화	결 과
maximum length of context	$n \geq 1$	증가	추출되는 내용의 수 증가
Minsup	$0 \leq x \leq 1$	증가	연관 규칙의 수 감소
Minconf	$0 \leq x \leq 1$	증가	연관 규칙의 수 감소
minimum similarity between contexts	$0 \leq x \leq 1$	증가	생성되는 군집의 수 증가
minimum number of contexts in a cluster	$n \geq 1$	증가	관계가 되는 군집의 수 감소
minimum average similarity between contexts in a cluster	$0 \leq x \leq 1$	증가	관계가 되는 군집의 수 감소
minimum similarity between patterns	$0 \leq x \leq 1$	증가	일반화되는 관계가 되는 군집의 수 감소

4.3 실험 및 평가

본 연구의 실험에서는 온톨로지의 구축을 위해서 임의의 매개 변수의 값을 설정한 뒤, REX 시스템으로부터 추출된 개념간의 관계들과 그것들의 이름들에 대한 가치를 평가한다. 실험에서는 앞서 설명한 Ziff 문서와 그 문서 집합에 존재하는 개념을 사용하며, 실험을 위한 매개 변수의 값은 <표 4>와 같다.

〈표 4〉 실험을 위한 매개 변수 설정값

매개 변수	값
maximum length of context	5
Minsup	0.0
Minconf	0.05
minimum similarity between contexts	0.6
minimum number of contexts in a cluster	0.6
minimum average similarity between contexts in a cluster	0.6
minimum similarity between patterns	2

실험에서는 <표 3>에 기술된 매개 변수 값들로 REX 시스템을 설정한 후, Ziff 문서 집합과 3,800여 개의 개념들로부터 36,501개의 내용들을 추출하였으며, 최종적으로 473개의 두 개념간의 관계들을 추출하였다. 또한, 111개의 일반화된 관계를 추출하였다. (그림 4)는 REX 시스템에서 추출된 개념 사이에 존재하는 관계들의 예를 보여준다.

(그림 4)과 같이 REX시스템은 입력으로 사용되는 개념 사이에 존재하는 관계와 그것의 이름을 제시함으로써, 온톨로지를 구축하는 과정에서 사용자에게 직관적인 정보를 제공한다. 반면, (그림 4)의 17-20번 예는 추출된 관계의 이름 중 관계의 이름으로 사용하기에 부적절하다고 판단되는 것들을 보여준다. 우선, 17번 예는 개념간 관계의 설명에는 도움을 줄 수 있지만 정확도가 떨어지는 것을 보여주며, 18-19번 예는 정보 검색 분야에서 불용어(stopword)로 사용되는 단어들이 개념 사이에 존재함으로 발생하는 문제를 보여준다. 마지막으로 20번 예는 'NCR Corp'과 같이 하나의 개념으로 간주되어야 할 어구가 입력된 개념의 정확한 매칭을 사용함으로써, 개념과 내용으로 분리되어 발생하는 적절하지 못한 예를 설명한다.

본 실험에서는 REX 시스템의 결과로 추출된 473개 관계와 그 관계를 설명하는 이름이 개념간의 관계를 설명함에 있어서 얼마나 유용한가를 평가하기 위한 방법으로 추출된 개념 쌍과 관계의 이름에 대해 세 명의 도메인 전문가의 판단을 사용하였다. <표 4>는 추출된 개념간의 관계들의 이름에 대한 유용성에 대한 평가를 보여준다. 또한, 관계를 형성하는 개념의 형태(type, 즉, "회사-회사" 혹은 "회사-제품", "제품-제품", "제품-회사")에 대한 군집화의 기대값을 보여준다.

이 실험 결과는 개념 쌍과 연관 규칙의 지지도와 신뢰도만으로 개념간의 관계를 설명한 기존의 KAON 시스템에 비해, 본 연구에서 제안한 방법은 온톨로지의 개념간의 관계 정의에 있어서 직관적이고 유용한 정보를 사용자에게 제공함으로써, 온톨로지의 구축과 유지에 유용하게 사용될 수 있음을 보여준다.

```
1   [Concept1:          Apple ][Concept2:          System 7.0 ]
    [Concept1:         Novell ][Concept2:            NetWare ]
    [Concept1:      Microsoft ][Concept2:          SQL Server ]
    relation name : announced
2
    [Concept1:      Microsoft ][Concept2:               OS / 2 ]
    relation name : developed

3   [Concept1:      Microsoft ][Concept2:         Windows 3.0 ]
    relation name : designed

4   [Concept1: Apple Computer ][Concept2:           Microsoft ]
    [Concept1:          Xerox ][Concept2:               Apple ]
    [Concept1:          Apple ][Concept2:           Microsoft ]
    relation name : ' s suit against

5   [Concept1:      Microsoft ][Concept2:               OS / 2 ]
    relation name : is now marketing

6   [Concept1:          3 Com ][Concept2:           Microsoft ]
    [Concept1:         Proteon ][Concept2:  Texas Instruments ]
    relation name : is working with

7   [Concept1:  SQL Solutions ][Concept2:              Sybase ]
    [Concept1: Computervision ][Concept2:      Prime Computer ]
    relation name : a subsidiary of

8   [Concept1:          3 Com ][Concept2:           3 +Open ]
    [Concept1:      Printware ][Concept2:         PostScript ]
    relation name : demonstrated its

9   [Concept1:         Novell ][Concept2:            NetWare ]
    [Concept1:  Ashton - Tate ][Concept2:              dBASE ]
    relation name : offers

10  [Concept1:  Oracle Server ][Concept2:             Oracle ]
    [Concept1: Microsoft Mail ][Concept2:The Network Courier ]
    [Concept1:       Parlance ][Concept2:          DECwindows ]
    relation name : is based on

11  [Concept1:          Ingres ][Concept2:ASK Computer Systems ]
    [Concept1:            TOPS ][Concept2:    Sun Microsystems ]
    relation name : division of

12  [Concept1:Computer Library ][Concept2: Ziff Communications ]
    relation name : is published by

13  [Concept1:     Excelerator ][Concept2:    Sub Microsystems ]
    [Concept1: .    IconAuthor ][Concept2:              IBM PC ]
    relation name : runs on

14  [Concept1:    Grid Systems ][Concept2:               Tandy ]
    [Concpet1:        Kinetics ][Concept2:               Novell ]
    relation name : now owned by

15  [Concept1:          OS / 2 ][Concept2:           Microsoft ]
    [Concept1:            TOPS ][Concept2:    Sun Microsystems ]
    [Concept1:    RightWriter  ][Concept2:           RightSoft ]
    relation name : developed by

16  [Concept1:          FoxPro ][Concept2:               dBase ]
    relation name : replaces

17  [Concept1:      Microsoft ][Concept2:               OS / 2 ]
    [Concept1:          Pixar ][Concept2:           RenderMan ]
    relation name : will continue to develop

18  [Concept1:          OS / 2 ][Concept2:          MS - DOS ]
    [Concept1:Correct Grammar  ][Concept2:   Houghton Mifflin ]
    [Concept1:        DECWrite ][Concept2:          DECwindows ]
    relation name : is , as

19  [Concept1:         SunSoft ][Concept2:    Sun Microsystems ]
    [Concept1:     GlobalView  ][Concept2:           ViewPoint ]
    relation name : , the

20  [Concept1:              NCR ][Concept2:             Netware ]
    Relation name : Corp introduced
```

(그림 4) REX 시스템에서 추출된 관계의 예

〈표 5〉 추출된 관계의 이름에 대한 유용성에 대한 통계값

Number of extracted relations	Number of useful names	Average of usefulness
473	277	58.56%

Expectation of a useful relation	Average precision of clustering	Expectation of a correct cluster
55.75%	78.44%	74.69%

5. 결 론

인터넷의 발달로 인한 웹의 대중화는 정보의 공유 및 검색 등에 기계의 활발한 참여를 가져왔으나, 웹에 존재하는 대부분의 정보는 사람에 의한 이해를 목적으로 작성되었다. 이것은 기계가 사용자의 요청을 처리하는데 있어서 큰 장애로 작용하였다. 이를 해결하기 위한 방법으로 기계가 이해할 수 있는 형태로 정보를 표현하기 위한 시맨틱 웹이 제안되었다. 하지만 시맨틱 웹의 논리적 기반을 구성하는 온톨로지를 구축하고 유지하는 작업은 많은 시간과 비용을 요구한다. 따라서 본 연구에서는 온톨로지 구축 과정의 한 부분인 개념간의 관계를 문서 집합으로부터 추출하는 방법을 제안했다.

제안된 방법에서는 문장 상에 존재하는 두 개념들 사이의 내용은 두 개념을 설명하는 중요한 정보가 된다는 것을 가정한다. 이 가정을 기반으로 문장으로부터 두 개념 사이에 존재하는 내용을 추출하고, 이들 내용 중에서 중요한 개념쌍을 찾기 위해 연관 규칙을 찾는다. 그리고 연관 규칙으로 발견된 개념 쌍들의 내용의 군집화를 통해 개념간의 관계를 설명하는 패턴을 찾고, 다시 일반적인 패턴을 찾기 위해 패턴의 군집화를 수행한다. 최종적으로 이렇게 생성된 패턴과 내용을 사용하여 두 개념간의 관계를 위한 이름을 부여한다.

본 연구의 실험에서는 제안된 방법의 효용성을 검증하기 위해 실제 문서 집합과 개념들로부터 개념간의 관계와 이름을 추출하였으며, 그 결과로 추출된 관계의 이름 중 약 59%가 온톨로지의 구축과정에 있어서 사용자에게 도움을 줄 수 있다고 판단되었다. 이것은 제안된 방법이, 기존의 KAON 시스템의 직관적인 정보 제공의 부재와, SnowBall 시스템의 초기 입력으로 사용되는 개념 쌍(seed) 선정 과정의 어려움을 극복할 수 있는 충분한 대안이 될 수 있음을 보여준다.

참 고 문 헌

[1] Marti A. Hearst. "Automatic Acquisition of Hyponyms from Large Text Corpora" In *Proceedings of the 14th International Conference on Computational Linguistics*, 1992.

[2] William B. Frakes and Ricardo Baeza-Yates, editions. "Information Retrieval: Data Structure and Algorithms", Prentice-Hall, 1992.

[3] Thomas R. Gruber. "A Translation Approach to Portable Ontology Specifications" *Stanford Knowledge System Laboratory Technical Report KSL-92-71*, pp.1-2, 1993.

[4] Rakesh Agrawal and Ramakrishman Srikant. "Fast Algorithms for Mining Association Rules", In *Proceedings of the 20th International Conference on Very Large Databases (VLDB)*, September, 1994.

[5] Ramakrishnan Srikant and Rakesh Agrawal. "Mining Generalized Association Rules", In *Proceedings of the 21st VLDB Conference*, 1995.

[6] Mark Sanderson and Bruce Croft, "Deriving Concept Hierarchies from Text", In *Proceedings of the 22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.206-213, 1999.

[7] A. Maedche and S. Staab. "Semi-Automatic Engineering of Ontologies from Text", In *Proceedings of the 12th International Conference on Sw Engineering and Knowledge Engineering(SEKE'2000)*, 2000.

[8] Eugene Agichtein and Luis Gravano. "Snowball: Extracting Relations from Large Plain-Text Collections", In *Proceedings of the ACM International Conference on Digital Libraries(DL'00)*, 2000.

[9] Dawn Lawrie and W. Bruce Croft, "Discovering and Comparing Topic Hierarchies", In *Proceedings of RIAD2000 conference*, pp.314-330, 2000.

[10] T. Berners-Lee, J. Hendler, and O. Lassila. "The Semantic Web", *Scientific American*, pp.35-43, May, 2001.

[11] Natalya F. Noy and Deborah L. McGuinness. "Ontology Development 101: A Guide to Creating your First Ontology", *SMI Technical Report SMI-2001-0880*, pp.1-25, 2001.

[12] Dawn Lawrie, W. Bruce Croft, and Arnold Rosenberg, "Finding Topic Words for Hierarchical Summarization", In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.349-357, 2001.

[13] Dawn J. Lawrie and W. Bruce Croft, "Generating Hierarchical Summaries for Web Searches", In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.457- 458, 2003.

[14] Phillipp Cimiano, Steffen Staab, and Julien Tane. "Automatic Acquisition of Taxonomies from Text: FCA meets NLP", In *Proceedings of the GI Workshop Lehren-Lernen-Wissen Adaptivität(LLWA)*, 2003.

[15] Hee-soo Kim, Ikkyu Choi, and Minkoo Kim. "Refining Term Weights of Documents Using Term Dependencies", In *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 552-553, 2004.

[16] 김희수, 조용석, 최익규, "문서로부터 계층적 개념 트리 자동 구축", 2004년 추계정보과학회, pp.103-105, 2004.

김 희 수

e-mail : heemanz@ceai.ajou.ac.kr

2003년 아주대학교 정보 및 컴퓨터공학부 (학사)

2005년 아주대학교 정보통신 전문대학원 (석사)

2005년~현재 (주) 엔텔리아 프로토콜 1팀 연구원

관심분야 : 인공지능, 정보검색, 데이터마이닝, 모바일 네트워크

최 익 규

e-mail : ikchoi@ceai.ajou.ac.kr

1993년 아주대학교 공과대학 전자계산학과 (공학학사)

1995년 아주대학교 컴퓨터공학과(공학석사)

1995년~2000년 한국정보공학 재직

2003년 아주대학교 정보통신공학과 박사 수료

2004년~현재 아주대학교 프로그래밍 전문강사

관심분야 : 지능형 정보검색 시스템, 온톨로지

김 민 구

e-mail : minkoo@ajou.ac.kr

1977년 서울대학교 계산통계학과(이학사)

1979년 한국과학기술원 전산학과(공학석사)

1989년 Pennsylvania 주립대 전삭학과 (박사)

1999년~2000년 Louisiana대학 연구과학자

1981년~현재 아주대학교 컴퓨터공학과 교수

관심분야 : 지능형 정보검색 시스템, 지능형 교수 시스템, 지능형 캐릭터 에이전트