□특별기고□

지능형 분산 다국어 문헌검색을 위한 정보 아키텍춰 에이전트

최 기 선†

◈ 목 차 ◈

요 약

지능형 문헌[1] 검색은 문헌 (document)의 완전 이해를 가정한다. "지능"이라 함은 두 가지 측면을 갖는다. 하나는 사용자 친숙성이요, 다른 하나는 저자의 생각 그대로를 반영하는 문헌의 완전 이해 상태이다. 전자는 후자에 포함된다고 볼 수 있다. 이해 상태란 이해를 하기 위한 정보와 지식이 갖추어져 있음을 의미한다. 문헌의 저자가 자신이 쓴 글에 대하여 가지고 있는 완전한 이해 상태가 복원되어 있는 문헌을 "완전 문헌"이라고 한다. 그 문헌의 독자 (혹은 사용자)가 완전 문헌과 같은 이해 상태에 있다면, 그 독자의 요구에 맞는 올바른 정보가 정확히 수집될 수 있다. 전달된 문헌이 완전 문헌으로 복원되어야 할 각 수준의 지식의 전체적 형태를 "문헌 아키텍춰"라고 부른다. 저자의 이해 상태인 완전 문헌으로의 복원 과정은 "정보 아키텍춰"라는 처리 과정으로 설명한다. 복원을 위하여 참여하는 프로세스나 자원 (혹은 엔티티) 은 정보 아키텍춰 네트워크 상의 자율적 노드이다. "자율적 노드"라 함은 이미 정해진 스스로의 목적을 위하여 능동적 역할을 함을 의미한다. 이 노드 들은 네트워크 상에 "분산"되어 있으며, 한 가지 일을 다루기 위하여 서로 "경쟁"하기도 하지만, 상호 "보완"적으로 한 가지 목적을 이루기 위하여 동작하기도 한다. "다국어" 정보

† 정회원 : 한국과학기술원 전산학과 교수

1 "document"는 문헌 혹은 문서로 번역한다. "문헌"은 "이미 수집된 가치 있는" 이라는 뜻을 갖는다. 또, 문헌은 의미적 분류를 내재한다. 반면 "문서"는 수집의 대상이라는 뜻을 갖고 있지 않다고 본다. 문서는 그 하나로서의 존재 가치를 갖는다. 문서의 "글자 배열", 문서의 "형식"과 같은 표층적 표상에 더 비중이 간다 이 원고에서는 위의 두 가지 뜻을 모두 "문헌"(혹은 "document")로 표현하고자 한다

서비스도 정보 아키텍춰의 개념 하에 같은 방법으로 시행된다. 이와 같은 번역 방식을 특히 "탄력적 기계번역" (flexible machine translation)이라고 한다. 정보 아키텍춰의 본질은 자율적 진화와 자기 조직력에 있다. "자율적 진화"는 아키텍춰의 구성원의 자율적 진화를 의미한다. 아키텍춰의 구성원으로서의 노드는 스스로 입력 환경에 적응하여 진화한다. "자기 조직력"은 아키텍춰의 구성원 간의 협력 방법이 상황에 적응하여 그 구조가 효율적으로 바뀜을 의미한다. "탄력성" (flexibility)이라 함은 독자의 지식 수준 (즉, 문헌 아키텍춰의 어느 수준)에 따라 (저자가 쓴) 문헌의 "문헌 아키텍춰" 상의 어느 수준에서나 독자의 "문헌 아키텍춰"로 정보 전달이 일어날 수 있다는 데에 있다. 정보 전달은 "정보 아키텍춰" 네트워크 상의 노드가 참여하여 이루어진다. 이 때, 참여하는 노드는 중복 가능하다. 즉, 같은 역할을 하려는 처리기나 자원이 동시에 한 가지 대상에 적용될 수 있다. 이 의미에서 참여 노드들은 상호 경쟁적이며 상호 보완적인 "탄력성"을 갖는다. 결론적으로, 정보검색과 기계번역이 한 개의 패러다임인 "정보 아키텍춰" 하에서 설명과 구축이 가능하며, 이는 현실적인 "표준화"에 기여한다.

1. 서 문

문헌 검색 서비스란 한 문헌이 올바른 길을 경유하여 올바른 정보화가 되어 올바른 장소로 이동함을 가정한다. 주어진 장소에 맞는 (혹은 주어진 장소를 위한) 올바른 정보의 수집은 문헌 집합을 들춰봄 (브라우즈), 탐색 (찾아냄), 추출 (뽑아냄), 여과 (걸러냄) 등을 통하여 얻는다. 여기서, "장소"라 함은 정보를 요구하는 주체이다. 정보 요구 주체란 사용자, 에이전트, 또는 데이터베이스일 수도 있다. 한 가지의 "정보"는 여러 "문헌"에서 올 수 있다. 문헌 그 자체는 문헌의 내용인 문헌 정보의 표층적 표상이다. 표층적 표상은 따라서 아직 정보화하지 않은 데이터일 뿐이다. 예를 들면, "책"은 책이 전하고자 하는 정보를 담고 있으며, 인쇄된 책 그 자체는 정보를 표현하기 위한 표층적 표상이다. 한 가지 정보가 여러 문헌에서 추출될 수 있다는 것은 여러 문헌이 한 가지 정보를 여러 가지 "관점"으로 제시하였다는 것과 같다. 따라서, 한 가지 (혹은 한 단위의) 정보가 여러 문헌에서 "관점"이라는 "제약"을 통하여 변환된 것이라고 말할 수 있다. 정보는 정보를 보고자 하는 정보 요구 주체의 수준에 맞는 표현으로 바뀌어 문헌화 한다. 정보 요구 주체 (혹은 정보가 놓여진 "장소")의 수준은 정보를 받아 들이기 위한 문헌의 표층적 표상을 결정한다. 이 말을 더 능동적으로 표현하면, 정보 요구 주체의 수준을 결정하는 정보 요구 주체의 관점이 문헌의 표층적 표상을 결정한다고 말할 수 있다. 따라서, 한 단위의 정보가 "관점"의 제약하에 여러 문헌의 표층적 표상으로 "제시"된다. 정보 요구 주체는 문헌의 표층적 표상을 읽는 독자이기도 하고, 정보 요구 주체의 생각을 표층적 표상으로 쓰는 저자이기도 하다. 저자의 "관점"은 같은 정보라도 다른 방법으로 글을 쓴다. 다른 방법으로 쓴 글이란 결국 다른 표층적 표상으로 나타난 (혹은 제시된) 문헌이다.

문헌은 한 장소에서 다른 장소로 이동한다. 문헌을 만드는 "장소"에서 그 문헌을 필요로 하는 다른 "장소"로 이동한다. 이 때, 그 문헌을 필요로 하는 다른 장소에서는 문헌의 완전한 표상인 "완전 문헌" 중에서 필요한 수준 만을 간직하려고 한다. 만일 그 "장소"가 데이터베이스라면 그 데이터베이스의 형태에 따라 문헌의 변환이 선행된다. 만일 그 "장소"가 "정보"를 간직하는 곳이라면 문헌에서 그 장소의 "관점"에 따라 변환된

형태의 "정보"를 요구한다. 또, 그 "장소"가 사람이라면 그 사람의 수준에 맞도록 그 사람의 "관점"에 맞게끔 변환된 문헌의 표층표상을 요구한다. 그 "장소"에 따른 완전 문헌의 변환 정도는 다음과 같은 선행 사항이 있기 마련이다. 즉, 그 "장소"의 수준 파악과 그 수준에 맞는 올바른 변환이 필요하다. 그 변환의 중간 단계를 각기 한 장소에서 다른 장소로 이동하기 위한 "중간 장소"라 한다면, "정보의 올바른 통로"는 한 장소에서 다른 장소로 이동하기 위한 가장 효율적 혹은 가장 짧은 길을 찾는 것과 같다.

이와 같이, "정보"는 한 "장소"에서 다른 "장소"로 이동한다. "장소"란 문헌의 여러 표상 중의 하나이다. 여기서, 문헌의 표층 표상에서 각 문헌이 쓰여진 "관점"을 제거한 정보로의 이동을 "정보 추출"이라고 부른다. 또는, 문헌의 표층 표상의 입장에서 본다면, 이 과정은 "문헌 검색"이라고 부른다. "문헌 검색"은 결국 "장소"의 하나인 "사용자"가 요구한 "정보" 요구에 적합한 내용을 포함하는 표층 표상으로서의 문헌 들을 찾는 것이기 때문이다. 한편, "장소"의 하나인 사용자는 사람이며, 그 사람의 정보 요구의 형상화가 이루어지지 않았을 경우, 문헌 (혹은 문헌의 표층 표상)을 "들춰봄"으로서 그 사람의 필요 정보를 형상화함과 동시에 그 정보가 그 정보의 사용자로서의 그 사람의 관점에 맞도록 표층화된 문헌을 찾는다. 이와 같은 과정을 사용자의 입장에서 본다면 "정보 탐색"이라 부른다. 정보와 문헌의 입장에서 본다면, 두 가지 측면을 갖는다. 사용자의 정보 요구의 형상화 과정에서 본다면 "정보 공간의 항해"일 것이며, 사용자의 정보 요구에 맞으며 사용자의 관점에 맞는 문헌을 찾는다는 면을 본다면 "문헌 공간의 항해"라고 표현함이 옳다.

예를 들면, "유한 오토메타 이론"에 대한 문헌이 있다고 하자. 저자가 오토메타에 대한 설명을 수학적으로만 하였다고 하자. 이 때, 독자가 매우 수학적으로 숙달되어 있는 독자의 "관점" 하에 그 문헌을 저작하였다. 만일 그 저자가 오토메타 이론을 초등학교 학생에게 설명을 하려고 하였다면, 만화를 이용하여 예제 중심으로 설명을 하려고 하였을 것이다. 문헌의 표층 표상은 수학에 숙달된 독자에게는 수식과 그에 대한 설명문이고, 초등학교 학생들의 문헌에는 만화이다. 문헌의 표층 표상은 독자에게는 "데이터"일 뿐이다. 숙달된 독자의 관점은 수식과 그 수식의 설명에 있으며, 초등학교 학생의 관점은 수식에 있지 않고 그 학생 주변에서 일어 날 수 있는 사건에 의한 설명이다. 그러나, 그 관점이 부여되기 이전의 모습, 즉 독자를 생각하지 않은 표상 만을 본다면 "오토메타 이론"에 대한 "정보"와 "지식"이 있다. 오토메타의 定義와 관련 定理는 오토메타에 대한 "지식"이다. "정보"는 "지식"이 쓰여진 "상황"을 포함한다. 오토메타가 자연언어처리의 구문해석에 쓰여진 것이라면 여기서 오토메타 이론의 상황은 "구문해석"이다. 상황은 그 지식의 용도를 분명하게 한다. 오토메타의 노드는 구문해석기의 상태이며, 오토메타의 가지는 다음에 볼 입력 토큰이다. 그러나, 이 오토메타 이론이 "사전 구조"에 적용된다면 노드는 단어이며, 가지는 다음에 볼 글자이다. 이와 같이 상황이 부여된 지식을 정보라고 한다.

독자 (혹은 사용자)가 구문해석과 오토메타에 대한 정보를 얻고자 할 때, 최종적으로 얻는 것은 이 정보가 잘 표상화 된 문헌을 얻는다. 따라서, 정보는 지식과 문헌을 연결하는 형태라는 것을 알 수 있다. 즉, 지식과 데이터 (즉, 문헌) 간의 통로로서 정보는 존재한다. 통로의 표상은 "색인"일 수도 있고, "링크"일 수도 있는 구조이다. 효율적 구조는 최단거리의 탐색을 위함이다.

"정보 흐름"은 데이터의 최적 정보화 과정이다.

이 최적 정보화를 "정보 추출"이라고 한다. "정보 항해"란 위에서 언급한 바와 같이 독자의 정보 요구가 불명확 혹은 형상화가 안되었을 때, 문헌의 정보화 과정을 독자가 그 관점에서 하여 가면서 정보를 찾고자 하거나, 또는 정보의 효율적 구조화가 미흡하여 그 구조를 사용자의 머리 속에 있는 구조를 이용하여 재생하는 과정이다.

"지능형" 문헌 검색은 정보 흐름의 효율성을 얻기 위함이다. "지능"은 주어진 문헌의 "이해"의 어느 정도를 가정한다. 문헌 검색의 "지능화"는 문헌들에 대한 몰이해에서는 나오지 않는다. 문헌들에 대한 "이해"는 문헌 간의 구조, 정보 간의 구조, 문헌의 완전 문헌화 복구에 의존한다. 문헌이 그 문헌이 놓여진 "장소" (사람, 에이전트, 컴퓨터 등)에서 이해 가능하고 적절하다면 주어진 장소의 정보 요구에 맞는 올바른 정보와 적절한 표상화가 되었다고 말한다.

"이해" 상태는 문헌과 장소에 따라 다르다. 다시 말하면, "이해" 상태는 문헌에도 있고 장소에도 있다. 완전 문헌은 완벽한 이해 상태를 나타낸다. 장소에서의 이해 상태는 장소의 수준에 의존적이다. 따라서, 어느 문헌이 "이해 가능"이라면, 문헌의 완전 정도와 장소의 수준 정도와 맞물려 이해 가능 상태로 간다고 생각한다. 문헌에 쓰여진 개개 문장의 문법적 완전성, 문체의 제시 완전성, 정보 구조 완전성, 내용의 설명 방법 (혹은 시나리오)의 표현 완전성 등이 문헌의 완전 정도에 대한 척도이다. 한편, 장소의 수준 정도는 독자 (혹은 사용자)의 문헌 이해력이다. 문헌이 완전 문헌화 하면 할 수록, 독자의 문헌 이해력이 낮다 하더라도 전반적 이해 상태는 높아진다. 반대로 독자의 문헌 이해력이 높으면 문헌의 완전 문헌화 정도가 낮더라도 독자의 이해 상태는 높다. 여기서 독자의 문헌 이해력은 독자의 상황에 맞도록 지식을 이용할 수 있으며, 독자의 관점에 맞는 정보의 표층 표상 만을 다룰 수 있다. 이때, 한 장소에서 그 문헌을 이해하지 못하면, 그 장소에서는 저자의 장소로 더 많은 설명을 요구할 수 있다.

같은 문헌에 대한 이해 상태가 독자에 따라 다른 것은 두 독자 (혹은 장소)의 지식 수준이 다르다는 것을 의미한다. (여기서 "지식 수준"에 대한 정의는 내리지 않았다.) 다시 말하면, 두 개의 다른 장소 사이에 "지식 격차"가 있음을 의미한다.

이제 본래의 문제인 "효율적 문헌 검색"으로 돌아 가기로 한다. 효율적 문헌 검색을 위하여 문헌에 대한 "이해 상태"를 가정하였고, 그 이해 상태는 "완전 문헌"을 전제로 하였다. 완전 문헌은 문헌의 완전 문헌 정도와 장소의 이해력 정도를 합하여 보완적으로 완전 문헌을 이룰 수 있다고 하였다. 우리의 문제는 완전 문헌은 어떻게 구성하는가로 옮겨진다. 다음 절에서 설명할 "문헌 아키텍춰"는 문헌의 "이해"를 설명하기 위한 모델로 역할을 한다.

문헌 아키텍춰는 한 개의 문헌의 완전 이해 상태를 나타내기 위함이다. 위에서 언급한 바와 같이, 완전 문헌은 문헌과 장소의 보완적 통합에 의하여 가능하다. 완전문헌으로의 복구 과정에 여러 프로세스와 자원이 필요하다. 이들 역시 한 장소에만 국한하지 않고 여러 장소에서 협동적으로 혹은 경쟁적으로 복구 과정에 참여한다. 이를 "정보 아키텍춰"라고 한다.

"다국어" 처리는 문헌 아키텍춰의 "이해"의 정도로 설명한다. 또, 다국어 처리 (혹은 번역)은 정보 아키텍춰 상에서의 복원 과정과 같다.

마지막으로, 문헌 아키텍춰와 정보 아키텍춰는 관련 노드의 협동적이며 경쟁적 참여에 기반하므로, 아키텍춰 각 계층 간의 "정보 교환 형식"에 대하여 논의한다.

2. 지능형 문헌 검색을 위한 문헌 아키텍춰

2.1 문헌 아키텍춰와 문헌 이해

"문헌 아키텍춰"는 문헌의 완전한 이해를 위하여 필요한 이해 과정의 모든 내용을 담은 계층적 구조이다. 예를 들면, 문헌 아키텍춰는 다섯 개의 계층으로 이루어져 있다고 말할 수 있다. 즉, 문자 (또는 모든 객체 포함), 문헌 배치, 제시 데이터, 정보 구조와 지식 등이다. 첫 두 계층 (즉, 문자와 배치)는 문헌의 "표층" 표상이다. 제시 데이터 계층과 정보 구조는 문헌의 통사 구조와 같다. 정보 구조의 어느 측면과 지식은 문헌의 의미 부분이다.

한 개의 문헌은 글자 (혹은 그림 등) 와 배치 구조 (예, 제목, 단락 등)로 쓰여져 있다. 저자가 문헌을 쓸 때, 그는 글자와 배치 구조에 대하여 알고 있다. 저자가 컴퓨터 건반을 두드리면 그 입력은 표준화된 글자 코드로 저장된다. 이러한 것 들이 문헌의 표층 표상이다. 모든 문헌은 글자 (혹은 그림 등)로 표현되고 이 글자 들이 어떤 배치 하에 놓이게 된다. 이 때, 독자[2]는 이들 표층 표상을 인식한다는 것을 전제로 한다. 그렇지 않다면, 표층 표상을 이해 시키기 위한 다른 프로세스가 동작한다. 예를 들면, "다국어" 처리가 그 하나이다.

문헌의 이해는 단지 문자나 문자열의 배치에 대한 인식 만으로 끝나지 않는다. 문자나 문자열의 배치와 같은 데이터 인식 뿐만 아니라, 데이터 구조의 논리적 인식이 수반되어야 한다. 책을 읽을 때 인식되어야 할 논리적 구조는 문헌에 내재하는 구조적 연계로서, 일반적으로 텍스트 단위 관의 연계, 각주, 참고문헌, 그림 등에 대한 연계 구조이다. [3]

더 나아가서, 독자가 문헌을 읽으려면 문헌의 언어적 계층을 이해하여야 한다. 언어적 계층이란 형태, 통사, 의미론적 구조이다. 언어학적 이해 없이, 독자는 문헌의 문장을 이해할 수 없다. 마지막으로 완전 문헌은 전문용어 지식, 문헌이 쓰여져 있는 영역 지식을 포함한다.

문헌을 이해한다는 것은 독자의 장소에서 문헌 아키텍춰의 모든 면 인식에 필요한 능력을 보유한다는 것을 의미한다. 독자의 능력은 글자, 문헌 배치, 데이터 구조, 정보 구조 (언어 정보), 전문용어, 영역에 대한 지식을 포함한다. 문헌 검색이 문헌의 이해를 바탕으로 한다는 관점을 바탕으로 생각하여 보자. 문헌 검색이란 문헌의 저자와 많은 문헌 중에서 필요한 문헌을 찾아 읽으려는 독자와의 의사 소통이라고 생각하여 보자. 다시 말하면, 문헌 검색은 정보 생산자 (저자)와 정보 소비자 (독자) 사이의 "의사 소통"이다. 검색의 행위는 의사 소통의 프로세스를 포함한다. 의사 소통은 쌍방의 같은 지식 수준을 전제로 한다. 의사 소통은 이해와 동반하여 일어나기 때문이다. 잘못된 이해 상태에서 발생한 의사 소통은 결국 다른 지식[4] 수준에 기인한다. 이와 같은 지식 격차는 "의사 소통의 병목"을 유발한다. 저자와 독자 장소의 지식 상태에 대하여 다시 생각하여 보자.

문헌의 저자 (장소)는 그 저자가 쓴 문헌을 완전히 이해한다는 점에서 출발한다. 저자는 문헌 내의 모든 지식 수준을 가지고 있다. 저자는 문헌을 저작할 때, 그 문헌의 독자가 가지고 있어야 할 적정량의 지식 수준을 가정한다. 저자의 예상이 빗나갈 때, 의사 소통 병목 현상이 발생할 것이다.

---

2 여기서, "독자"란 "장소"를 의미한다. 장소는 반드시 그 장소의 역할 (혹은 과제)을 다하기 위한 만큼 문헌을 이해할 수 있는 능력을 보유하여야 한다.

3 하이퍼텍스트는 이 연계 구조를 물리적으로 연결한 결과이다.

4 "지식" 수준은 문헌 아키텍춰의 모든 계층을 이해하기 위한 정적인 지식과 다른 장소에 있는 지식을 찾아 이용하는 동적 지식, 그리고 유추하는 지식을 모두 포함한다.

이제 독자 측에서 생각하여 보자. 주어진 문헌에 대하여, 독자는 자신이 기억하고 있는 지역적 지식만을 이용하여 이해하려 들 것이다. 독자가 사람이면, 그 사람의 기억을 바탕으로 할 것이며, 독자가 컴퓨터이라면, 그 컴퓨터 프로그램은 자신의 지역 데이터베이스를 참조하여 처리하려 할 것이다. 독자가 완전히 이해를 못하는 경우, 다른 장소의 지식을 이용하려 할 것이다. 예를 들면, 사전이나 백과 사전을 참조한다. 독자가 컴퓨터라면, 그러한 공중 지식은 종이에 쓰여진 사람이 읽을 수 있는 형태가 아니고 기계가 이해할 수 있는 형태로 되어 있어야 한다. 그래도 이해가 안 될 경우, 독자는 저자에게 질문을 할 것이다. 저자는 적절한 대답을 준다. 이와 같은 대답은 독자의 문헌 아키텍춰의 비어져 있는 부분을 메우기 위한 것이다. 피드백 정보는 저자의 의도[5]를 이해하는데 도움을 준다.

2.2 완전 문헌과 문헌 교환 형식

"완전 문헌"은 문헌 아키텍춰의 모든 계층에 대한 완전한 기술이 필요하다. 의사 소통이 일어날 때, 문헌의 양측은 상대방이 저작한 문헌의 표층 구조에서 완전 문헌을 복원할 수 있다고 가정한다. 완전 문헌의 복구가 올바른 의사 소통의 전제이다. 이와 같은 완전 문헌의 사양을 "문헌 교환 형식"이라고 하자.

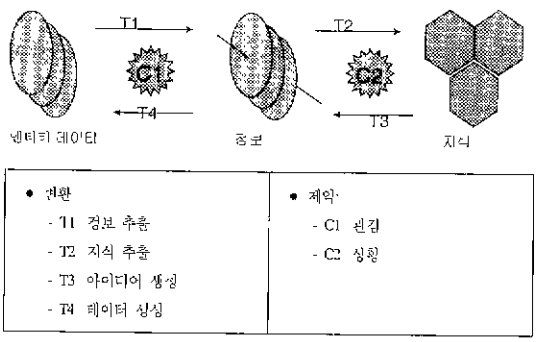
완전 문헌은 양측의 지식을 이용하여 복원된다. 양측은 독자 장소의 이해를 돕기 위하여 상호 협조적이기 때문에 양측의 지식은 서로 보완적이다. 한편, 양측은 완전성을 향하여 서로 경쟁적이기도 하다. 다음 절에서 "정보 아키텍춰"에 대하여 논의하고자 한다. 정보 아키텍춰는 의사 소통을 위하여 양측의 지식의 요소인 엔티티나 프로세스를 명확하게 하기 위함이다.

5 저자의 "의도"는 독자의 질문에 대한 대답에 반영한다. 그 대답이 의도에 대한 거울과 같은 것이다.

2.3 정보 아키텍춰

2.3.1 정보 아키텍춰의 정의

"정보 아키텍춰"는 문헌 아키텍춰를 구성하기 위한 과정을 면밀화 한다. 정보 아키텍춰 (그림 1)는 엔티티, 변환, 변환 간의 제약으로 이루어진다. 엔티티는 데이터, 정보, 지식의 세 종류이다. 엔티티 간 변환은 네 가지의 프로세스가 있다: 정보 추출, 지식 획득, 아이디어 생성, 데이터 제시 등이다. 제약은 두 종류로서 엔티티 간 변환에 구속력을 갖는다. 데이터와 정보 사이에는"관점"이 있다고 하고, 정보와 지식 사이에 "상황" 제약이 있다.



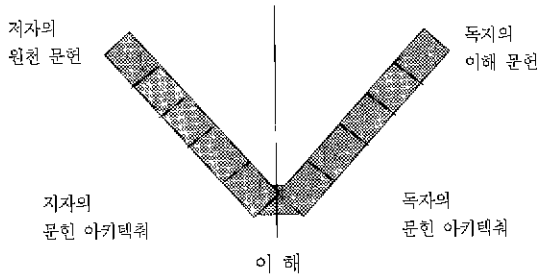
(그림 1) 정보 아키텍춰의 구성

정보 아키텍춰의 첫 엔티티는 "데이터"이다. 데이터는 문헌 아키텍춰의 표층 표상에 불과하다. 예로서 텍스트나 문서 작성기의 편집 결과 혹은 멀티미디어 문헌이 있다. 저자는 데이터의 형태로 문헌을 가꾼다. 저자는 정보의 형태로 문헌을 쓰지 않는다. 두 번째 엔티티는 "정보"이다. 정보는 데이터의 구조화의 결과이다. 문헌이 정보의 상태로 바뀌었을 때, 문헌의 표층 단위는 상호 참조 링크로 연결되어 있다. 언어 태그는 언어 단위를 주석한다. 정보의 예로서 다음을 생각한다. 구조화 데이터로서 하이퍼텍스트, 문장의 단위에 주석된 형태/통사 태그 - 가공된 구조화 데이터로서

이해 상태에 더 가까워져 있는 형태이다. 세 번째 엔티티는 "지식"이다. 지식은 각기 정규화[6] 표상을 지닌다. 이와 같은 지식은 용어 지식, 영역 지식 등을 포괄한다.

정보는 데이터 제시 관점을 없앰으로써 데이터에서 추출한다. "정보 추출"은 데이터에서 정보를 얻기 위한 변환이다. 여러 문헌의 내용은 한 단위의 정보로 요약할 수 있다. 요약은 정보 추출 변환의 한 종류이다. "관점"은 문헌의 독자가 데이터로서 볼 정보의 "옷"과 같다. 저자의 관점은 같은 정보라도 다른 제시 형태로 문헌을 쓰게 한다. 독자가 어린이라면 문헌은 매우 쉬운 제시 방법으로 쓰여지며 그 문헌은 많은 그림으로 제시된다. 전문가를 위한 문헌은 매우 기술적인 사실을 전달하기 위하여 형식화한 수식을 쓴다. 데이터의 제시는 "장소"의 "관점"에 의존적이다. 정보를 데이터화 하기 위한 "데이터 생성" 혹은 "데이터 제시" 과정은 독자의 수준에 맞는 잘 읽히는 문헌을 제작한다.

지식은 정보에서 "상황" 요소를 제거함으로써 정규화 과정을 거쳐 형식화한다. 이러한 과정을 "지식 추출"이라고 한다. 전통적으로, "학습" 혹은 "지식 획득"과 같은 용어가 쓰였다. 그러나, 이와 같은 전통적 용어는 데이터에서 직접 지식을 얻기 위한 변환에 쓰였던 용어이다. 지식 추출의 역과정인 "아이디어 생성"은 주어진 상황에 적합한 정보를 생성한다. 상황은 한 단위의 지식에 예증을 엮어, 지식의 예로서 정보를 생성한다. 한 단위의 정보는 주어진 상황에 의하여 생성된다. 이때, 한 묶음의 지식이 상황의 자로 재어져 엮어져 한 단위의 정보를 생성한다. 이와 같은 "아이디어 생성" 과정은 사용자로 하여금 아이디어를 생각해 내도록 한다. 다만, 그 아이디어란 데이터와 같은 마지막 단계의 제시 형태가 아니므로, 아이디어 표상은 데이터 입장에서 보면 다른 장소에 이해 가능한 것이 아닐 수도 있다.



(그림 2) 분산 정보 아키텍춰

2.3.2 정보 아키텍춰의 정규화 엔티티

정보 아키텍춰의 각 엔티티는 각각의 최적 사양을 갖도록 만들어질 수 있다. 각 엔티티는 여러 응용으로 구현된다. "문헌"은 데이터의 한 응용이다. 문헌의 최적 사양 중에는 주어진 사용자에게 가장 설득적인 서식이나 문장 전개 방식을 포함한다. 또, 주어진 장소의 관점 (혹은 수준)에 따라 최적의 제시 방법을 쓴다면 그것도 데이터로서의 문헌에 대한 최적의 사양의 일부가 된다. 이와 같은 임의의 사용자 (혹은 임의의 장소)에 대한 최적의 데이터 형태를 "최적 설득성 데이터"라고 정의하자. 최적의 설득력있는 문헌을 생성하기 위한 사양으로서 최적 설득성 데이터를 발견하는 것이 정보 아키텍춰 하에서 연구되어야 할 항목이다.

정보의 정규형은 "최적 구조 정보"라고 부른다. 예를 들면, 최적 연결 하이퍼텍스트는 꼬임이 없는 정규화 상태에 있다. 모든 정보 단위는 최적의 방법으로 탐색할 수 있다. "지식" 엔티티는 그 정규형을 "최적 형태 표상"이라고 한다. 정규 논리 형태가 그 한 예이다.

문헌 아키텍춰의 각 계층은 정보 아키텍춰의

6 지식의 정규화 표상은 장소(컴퓨터 혹은 사람)에 따라 다를 것인가에 대하여 명확하지 않다.

엔티티로 투영할 수 있다.[7] 다음 절에서 다시 분산 환경에서의 문헌 검색에 관한 논의를 계속하고자 한다.

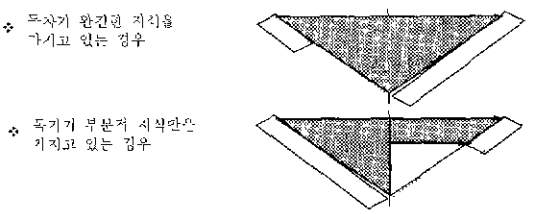
3. 정보 아키텍춰 네트워크와 지능형 분산 문헌 검색

앞 절에서 본 바와 같이, 저자는 저자 자신의 문헌을 이해할 충분한 완전한 지식을 갖추고 있지만, 저자는 완전 문헌을 만들어 내지는 않는다. 그러나, 완전 문헌은 지능형 문헌 검색을 하기 위한 전제 조건이다. 문헌 검색의 궁극적 목적은 저자와 독자 간의 효율적 의사 소통을 얻는 것이다. 문제는 어디에서 완전 문헌을 복구할 수 있을 것인가 이다. 이 논문에서 정보 아키텍춰에 의한 완전 문헌으로의 복원을 설명한다.

3.1 문헌 아키텍춰 복구와 정보 아키텍춰 네트워크

저자는 자신의 저작을 이해할 완전한 지식을 가지고 있다. 그렇지만 완전 문헌을 저자가 모두 만든다는 것과는 다르다. 완전 문헌을 만들기 위하여 메워야 할 부분이 저자의 장소에 있지 않고, 정보 아키텍춰의 해석 요소에서 공급한다. 즉, 그러한 지식이 있는 장소가 저자의 장소와는 다르다. 정보 아키텍춰의 모든 지식은 여러 가지 물리적 형태를 가진다. 그 장소가 컴퓨터 안이면, 그 물리적 형태는 기계가독형으로서, 컴퓨터 네트워크 혹은 저장 장치에 위치한다. 저자 측 정보 아키텍춰가 (그림 2)의 왼쪽과 같이 완전 문헌을 구성하고 복원한다. 한편, 독자의 이해는 (그림 2)의 오른 쪽과 같이 독자측의 완전 문헌에 기인한다.

7 문헌 교환 형식의 표준화는 정규화 엔티티의 개념하에 정의할 수 있다.



(그림 3) 정보 아키텍춰 네트워크의 경우
– 정보 아키텍춰의 역학

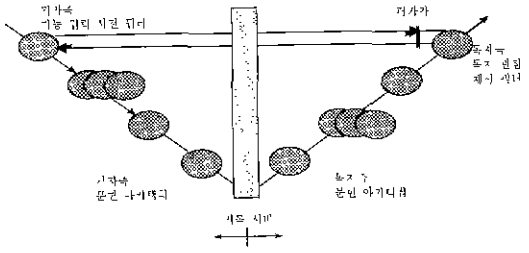
정보 아키텍춰 네트워크의 역학은 정적인 관점(그림 2)과는 거리가 있다. 독자와 저자의 문헌 아키텍춰는 보완형 객체 혹은 경쟁형 객체로 구성된다. (그림 3)에서 보는 바와 같이, 독자가 저자의 문헌을 이해할 완전한 지식을 가지고 있으면, 저자는 단지 문헌의 표층 형태 만을 보내면 된다. 왜냐하면, 문헌 아키텍춰의 독자측에서 그 자체의 지식 만으로 문헌 아키텍춰의 모든 계층을 복원할 수 있기 때문이다. 그러나, 독자측이 오직 부분적 지식만을 가지고 있다면, 저자 측은 독자의 불완전한 지식에 대하여 보완적이어야 한다.

3.2 정보 아키텍춰 네트워크의 구성과 정규화

완전 문헌 복원 지식은 (그림 4)에서 보는 바와 같이 정보 아키텍춰 네트워크의 노드에 분산되어 있다. 완전 문헌 복원에 필요한 지식이 불완전할 경우, 독자 (혹은 사용자)는 저자의 장소로 질문을 하고 저자는 이에 답을 준다. 이와 같은 피드백 과정과 내용은 중간 계층 노드에 기록된다. 이 중간 계층 노드는 네트워크에서 사실상 물리적인 기록 서버로 존재할 수도 있다. 정보 아키텍춰 네트워크는 그 스스로 진화한다. 이 때의 진화 방법론은 자기 조직 장치 등을 이용한 학습과 재구성 등의 방법을 쓴다. 기록 서버는 이와 같은 자발적 진화의 원천이 된다. 정보 아키텍춰는 정보 추출과 지식 추출을 위한 엔티티 간의 프로세

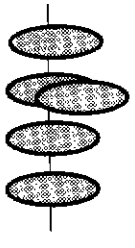
스를 가지고 있다고 하였다. 각 프로세스와 엔티티는 자발적이므로, 정보 아키텍춰에 참여하는 노드들은 자발적으로 진화한다. 프로세스는 중앙 집중적이 아닌 분산적이다. 같은 기능 (혹은 역할)에 대하여 참여하는 노드는 하나가 아닌 중복적이다. 여기서, 중복은 복제와는 다르다. 따라서, 이러한 노드는 상호 보완적 협력적이며 또한 경쟁적이다.

정보의 흐름은 단계적으로 증가한다.[8] 처음에는 문헌의 표층 표상이 그대로 독자에게 전달된다. 독자가 이해를 할 수 없다면, 독자측은 접수를 거부하며 저자의 장소로 반송한다. 이 때, 독자측은 문헌의 주어진 표상이 독자가 충분히 이해 가능한 것인가를 측정할 평가자를 가지고 있다.



(그림 4) 정보 아키텍춰 네트워크: 구성

(그림 4)에서는, 문헌 아키텍춰는 저자측과 독자측이 서로 다르고 분리되어 있다. 그러나, 문헌 교환 형식과 같은 표준화된 형태와 프로세스 프로토콜이 있다면, (그림 5)와 같이 하나로 묶여질 수 있다. 물리적 구축 비용도 따라서 줄고, 운용도 더 효율적일 것이다.



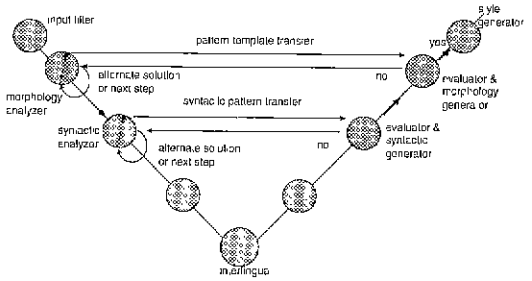
(그림 5) 문헌 교환 형식과 분산 정보 아키텍춰 네트워크

8 단계적인 증가 = incremental

4. 다국어 문헌 검색을 위한 탄력적 기계 번역

4.1 탄력적 기계번역의 구성과 정보 아키텍춰

번역의 과정을 생각하면, 번역의 과정에서 번역의 대상 문헌에 대한 이해가 전제가 되지 않음을 알 수 있다. 어떤 경우에는 번역 주형 모음 만으로도 충분히 번역할 수 있다. 예를 들면, 주식 기사를 번역하기 위해서 특수 상용구나 특별한 영역 표현만을 이용하면 된다. 이 과정에서 완전 문헌을 복원하기 위한 모든 프로세스가 일어나지는 않는다. 그 중 번역하고자 하는 한 프로세스가 실패를 하면, 문헌 아키텍춰의 더 깊은 수준의 다음 프로세스가 분석과 번역을 시작한다. 이 프로세스가 문헌 아키텍춰의 다음 수준을 메우고, 그 결과가 독자의 언어쪽으로 적절한 변환 지식에 의하여 전달된다. 완전 문헌을 복원하기 위한 프로세스 열은 요청에 따라 점진적으로 깨어난다. 프로세스 발진이 탄력적이라는 의미에서 이 과정이 탄력적이어 이와 같은 방식의 기계번역을 "탄력성 기계번역"이라고 부른다. 그림 6에서 보는 바와 같이, 첫 프로세스인 "형태소 해석기"가 실패를 하면, 그 프로세스는 다른 해를 제시하거나 다음 프로세스인 "구문 해석기"로 넘긴다. 구문 해석기의 결과는 구문 패턴 변환 지식에 의하여 변환된다. 독자측의 구문 생성기에 해당하는 노드 앞에 붙어 있는 평가자는 구문 변환 결과가 마지막 표층 형태로까지 생성될 수 있을 것인가를 사전 평가한다. 이와 같은 전과정이 독자측에서 거부 신호를 보낼 때마다 되풀이된다. 피드백 정보는 정보 아키텍춰 네트워크와 마찬가지로 기록한다 (그림 4). 결국, 계속적인 독자측 거부 신호가 오면, 끝에 가서는 이 프로세스는 중간 언어가 있는 곳까지 진행할 것이다. 반면에, 독자 측이 변환된 문헌을 받아들이면, 번역은 중간언어 지점까지 진행하지 않고 중간 단계에서 끝난다.

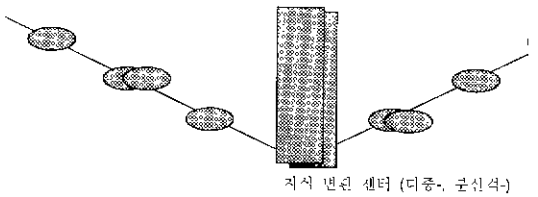


(그림 6) 탄력성 기계번역: 구성

4.2 분산 탄력성 기계번역

전 절에서 본 바와 같이, 탄력성 기계번역의 모든 모듈은 그 모듈의 결과가 나올 때마다 독자측의 목적 언어 생성에 연결된다. 모든 피드백은 기록이 되어 중간 노드에 저장되어 이에 따라 시스템은 자랄 수 있다. 탄력성 기계번역 시스템은 이와 같은 진화성 자기 조직형 네트워크의 일종이다.

탄력성 기계번역은 "분산"적 방법으로 구현한다. 모든 모듈은 네트워크의 노드이다. 이들은 고장 허용적[9]이다. 모든 수준이 같은 역할 (혹은 기능)을 하는 경쟁적 노드 들이기 때문이다. 그림 6과 같이 모든 노드는 경쟁적 혹은 보완적 협력적 프로세스나 엔티티이다.



(그림 7) 분산 탄력성 기계번역 운용
(DFMT: Distributed Flexible Machine Translation)

탄력성 기계번역 파라다임 하에서 기계번역 시스템을 개발하는 경우, 모든 단계에서 번역이 시도되고 가능하므로 처음부터 전 시스템이 가동될 수 있다. 변환 지식도 문헌의 일종이다. (저자측의) 원천 문장은 (독자측의) 목적 문장을 찾기 위한 질의어와 같다. 이 질의어는 목적 문장에 연결될 (혹은 변환될) 구성요소를 맞추기 위한 패턴을 찾기 위한 것이다. 이와 같은 과정은 문헌 검색의 프로세스와 같다.

5. 결 론

(지능형 분산 다국어 문헌 검색을 향하여)

균형잡힌 정보 흐름이 문헌 아키텍춰와 문헌 교환 형식의 표준화에 의하여 구현 가능임을 보였다. 전반적 모양으로서 정보 아키텍춰 파라다임을 제시하였다. 단일어 문헌 검색이나 다국어 번역 서비스도 정보 아키텍춰의 한 예이다. 문헌 검색의 "지능"은 완전 이해를 전제로 하고, 이것은 완전 문헌으로서 구현된다.

이 파라다임의 운용에 있어서, 표준화 문제가 성공적 의사소통의 실용적 목표의 하나로 제기되었다. 이 표준에 따라, 완전 문헌 복원 지식이 "분산" 네트워크에 자리잡을 수 있다. 여기서, 분산이라 함은 물리적 혹은 논리적으로 구현할 수 있다. 운용이 개발의 첫 단계에서부터 가능하다.

참고문헌

[1] Erbach, Gregor, "MULINEX: Multilingual Indexing, Navigation and Editing Extensions for the World-Wide Web," Cross-Lingual Text and Speech Retrieval (AAAI Spring Symposium), 25-31 (1997.3).

[2] Frederking, Rober, Teruko Mitamura, Eric Nyberg and Jaime Carbonell, "Translingual Information Access," Cross-Lingual Text and Speech Retrieval (AAAI Spring Symposium), 43-50 (1997.3).

[3] Jones, Karen Spark, Evaluating Natural Language Processing Systems, Springer-Verlag (1996)

9 fault-tolerant

최 기 선

1978년 서울대학교 수학과 (학사)
1980년 한국과학기술원 전산학 (석사)
1986년 한국과학기술원 전산학 (박사)
1987년 일본전기(주) 초빙연구원
1997년-현재 한국과학기술원 전산학과 교수
관심분야 : 자연언어처리, 한글공학, 정보검색, 지식획득

SOFT EXPO '97 개최

1. 일 시
전시회 '97 12. 10 ~ 12. 14 (5일간)
컨퍼런스 12. 9 ~ 12. 12 (4일간)
전 야 제 12. 9 (15:00)
폐 막 식 12. 14 (17:00)

2. 장 소
전 시 관 : 여의도 중소기업 종합전시장
컨퍼런스 : 여의도 중소기업회관
국제회의장, 세미나장

3. 추진기관
주 최 : 정보통신부
후 원 : 정보통신부 유관기관

4. 문 의
전화 : (02)583-6532, 3472-8683

※ 회원여러분의 많은 참석을 바라며 자세한 사항은 학회지 게시판을 참조바랍니다.