엔지니어링 문서의 문장 자동 계층정의 방법론

The methodology of automatic hierarchical definition for the sentences in engineering documents

박 상 일* • 김 봉 근** • 김 경 환*** • 이 상 호†
Park, Sang Il • Kim, Bong-Geun • Kim, Kyeong-Hwan • Lee, Sang-Ho

요 약

본 연구에서는 토목분야의 엔지니어링 문서를 대상으로 자동으로 문장의 계층을 정의하여 비 구조화 문서를 준 구조화 문서로 변환하는 방법론을 제시하고, 이를 시범적으로 구현하였다. 이는 기본적으로 문장의 머리기호가 그 문서에서의 계층을 표현한다는 점을 바탕으로 연구가 이루어졌다. 이를 위하여 본 연구에서는 첫째, 문장을 머리기호와 내용으로 분리하는 방법을 개발하였고, 둘째, 분리한 머리기호를 바탕으로 머리기호간 상대적인 위치를 파악하여 문장의 계층을 정의하는 프로세스를 제안하였다. 또한 제안한 방법을 이용하여 구현한 시범 모듈을 실무에서 사용하는 엔지니어링 문서를 대상으로 성능을 평가하여 개발한 모듈의 효용성을 검증하였다.

Keywords: 문서 계층정의, 엔지니어링 문서, 준 구조화된 정보, XML

1. 서 론

전자적으로 기록된 정보는 크게 구조화된 정보(structured information), 준 구조화된 정보(semi-structured information) 및 비 구조화된 정보(unstructured information)의 3가지 타입으로 나뉠 수 있다. 구조화된 정보란 정형화된 정보모델에 따라 기록된 정보를 말하며, 준 구조화된 정보란 정형화된 스키마는 없지만 데이터의 내용과 더불어 형식이나 의미를 지니는 부가적인 정보가 포함된 정보로서 웹문서나 스키마가 없는 XML 문서가 이에 속한다. 그리고 비 구조화된 정보란 특정한 형식이 없이 저장된 정보를 의미한다. 구조 및 수리계산서, 공사시방서 등 현재 건설 실무에서 엔지니어링 문서로 활용되고 있는 대부분의 문서들은 비 구조화된 정보형식에 속하며, 이들이 건설 프로젝트 수행과정에서 생산되는 정보의 상당부분을 차지하고 있다(Caldas and Soibelman, 2003). 한편, Liu *et al.* (2006)의 연구결과는 엔지니어링 분야의 경우 동일한 특징을 가지는 여러 문서의 내용을 제공받기 보다는 여러 비슷한 문서에서 일부분에 대한 정보를 필요로 하는 경우가 많음을 말하고 있는데, 이것은 빠른 정보의 검색, 추출 및 제공 등이 가능해야 실현될 수 있는 것으로 구조화 혹은 준 구조화된 정보의 형식에서는 손쉬운 구현이 가능한 반면, 비 구조화된 정보형식의 경우에는 그 실현이 매우 힘들다.

본 연구에서는 비 구조화 정보 형태인 엔지니어링 문서를 대상으로 준 구조화된 정보 형태로의 자동 변환 방법론을 제시하고, 웹을 기반으로 하는 시범적 모듈 구현 및 모듈의 성능 평가를 실시하였다.

* 학생회원 • 연세대학교 토목공학과 박사과정 si@csem.yonsei.ac.kr
** 정회원 • 연세대학교 토목공학과 박사과정 bgkim@csem.yonsei.ac.kr
*** 학생회원 • 연세대학교 토목공학과 석사과정 kay@csem.yonsei.ac.kr
† 정회원 • 연세대학교 사회환경시스템공학부 교수 lee@yonsei.ac.kr

2. 문장의 자동 계층정의 방법론

머리기호(heading symbol)는 문서의 작성자가 활용자에게 문서의 계층정보를 제공하기 위해 임의로 작성하는 것이지만, 엔지니어링 문서와 같이 일반적 서술이 많지 않고, 대규모의 종방향 문서 구조를 가지고 있는 경우 체계적인 문서를 작성하기 위해 필수적으로 사용한다. 따라서 문서의 머리기호를 구조화된 문서로 변환하는데 활용하는 것은 매우 효과적일 수 있으며, 본 연구에서도 이를 기본으로 활용한다. 그림 1은 텍스트 형식의 엔지니어링 문서를 준 구조화된 XML 문서로 변환하는 과정을 나타낸 것으로, 본 연구에서는 문장의 분리 프로세스와 문장의 계층정의 프로세스를 포함한다.

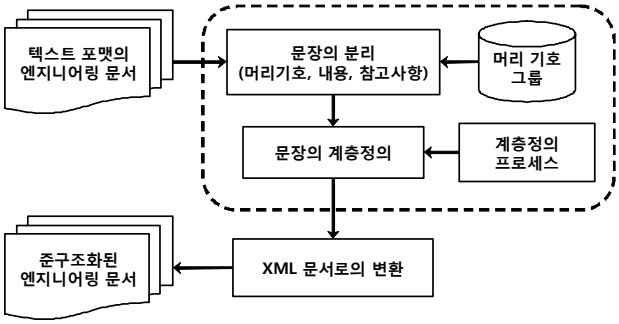

그림 1 문서의 자동 계층정의의 전체적인 흐름도

2.1 문장의 분리

앞서 설명한대로, 본 연구에서는 문서의 계층을 자동으로 정의하기 위해 머리기호를 사용했는데, 이를 위해서는 문장의 구성요소 파악이 선행되어야 한다. 식 1은 엔지니어링 문서의 문장을 표현한 식으로 문장은 문자열의 집합 M^n으로 표현 가능하다.

$$M^n ::= H^n C^n | H^n C^n R^n | C^n | C^n R^n \quad (1)$$

여기서, H^n은 머리기호에 대한 문자열 집합, C^n은 내용에 대한 문자열 집합을 나타내고, R^n은 참고사항 등 참조에 관한 문자열 집합을 나타낸다. 즉, 본 연구에서는 하나의 문장은 머리기호와 내용, 머리기호와 내용과 참조, 내용, 내용과 참조의 네 가지 경우로 구성되는 것으로 간주하였다. 이렇게 한 가지에서 세 가지의 문자열로 구성된 문장은 머리기호와 분리가 필요한데, 이를 위해서 본 연구에서는 그림 1에 나타나 있는 것처럼 머리기호 그룹을 활용하였다. 머리기호 그룹은 엔지니어링 문서나 일반 문서에 자주 나오는 머리기호들을 저장하고 있는 데이터 테이블로, 아라비아 숫자, 로마 숫자, 알파벳, 한글 자음, 원문자, 괄호문자, 기타 특수문자 등이 포함되어 있으며, 각 그룹마다 고유의 그룹 id를 가지고 있다. 문장에서 처음에 나타나는 문자를 머리기호 그룹과 비교하여, 머리기호 그룹에 존재하는 문자이면, 이를 머리기호로 식별하고 해당 그룹 id를 부여하며, 그렇지 않으면 내용 또는 내용과 참조로 식별하는 방식을 활용하여 머리기호를 구분해 낸다. 참조 문자열도 머리기호를 분류하는 방식과 비슷하게 미리 저장된 참조 구분자 테이블을 활용하여 분리해 내는데, 참조 문자열은 문장에서 내용을 정확히 식별하기 위해 사용하는 것으로 문서의 계층을 정의하는데 영향을 주지는 않는다.

2.2 문장의 계층정의

2.1절에서 제시한 방법을 활용하여 분리한 머리기호는 그 자체만으로는 문서 내에서 문장의 계층을 식별할 수는 없다. 일반적으로 문서를 작성할 때 로마 숫자(Ⅰ, Ⅱ, …)나 아라비아 숫자(1., 2., …)를 제일 처음에 사용하여 이것을 포함한 문장을 제일 상위계층 혹은 두 번째 계층이라고 간주할 수도 있지만, 문서 작성자의 취향에 따라서 얼마든지 바뀔 수 있다. 이는 머리기호를 사용하여 문장의 계층을 식별하기 위해서는 머리기호간 상대적인 구조위치를 파악해야함을 의미하는 것이다. 그림 2는 트리 노드의 순서 정렬 방식을 나타낸 그림으로, 그림 2의 좌측에 해당하는 깊이우선방식(depth-first

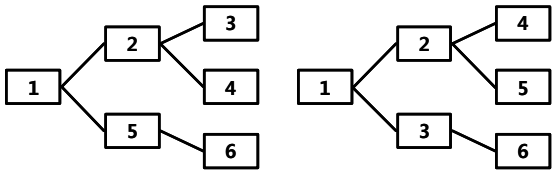

그림 2 트리 노드의 선형 순서 정렬 방식: 깊이우선방식(좌)과 너비우선방식

order)은 대부분의 문서에서 사용하고 있는 순서 정렬방식으로, 본 연구에서 계층을 정의하는 방식 역시 깊이우선방식과 유사하다. 즉, 제일 처음에 나오는 문장(1)을 가장 상위 계층으로 간주하고, 하위 계층인 2로 가서 계층을 정의하고, 또 하위인 3으로 가서 계층을 정의해 가다가, 더 이상 하위 계층이 없으면, 같은 구조적 위치에 있는 문장(4)을 찾아 계층을 정의하는 식이다.

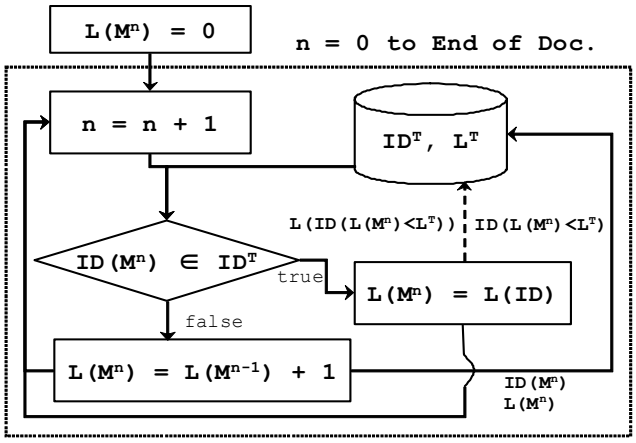

그림 3 자동 계층정의 프로세스

그림 3은 그림 2의 깊이우선방식을 응용하여 정의한 계층정의 프로세스를 나타낸다. 그림 3에서 L(X)는 X의 계층 등급을 나타내는 것으로, 제일 상위계층의 등급은 0이다. 그리고 ID(X)는 X의 머리기호 그룹 id를 의미하며, ID^T는 id 데이터 테이블을, L^T는 등급 데이터의 테이블을 의미한다. 이 프로세스에는 머리기호를 가지고 있지 않은 문장은 접근할 수 없으며, 첫문장(M^0)부터 읽어가면서 순차적으로 문장의 계층을 정의하고 해당 id와 등급을 저장해 나가다가, 그림 2의 4와 같은 경우처럼, 같은 계층 등급이 존재하면, 그 문장의 계층은 이미 존재하는 등급과 같은 계층으로 정의된다. 또한 이때에는 해당등급보다 더 낮은 등급의 id와 이에 해당하는 등급이 삭제가 되어, 이후에는 새로운 그룹 id로 계층이 정의된다. 그림 3에서 점선으로 된 직선이 데이터 삭제의 흐름을 나타내는 것으로, $ID(L(M^n)<L^T)$와 $L(ID(L(M^n)<L^T))$가 각각 삭제될 데이터의 id와 등급을 의미한다.

이러한 방법으로 자동 계층정의 된 문서는 Lee *et al.* (2006)이 제안한 방법에 따라 XML 문서형식으로 자동 변환되어, 다른 응용기술에 활용가능하게 된다.

3. 엔지니어링 문서의 자동 계층정의 모듈의 구현 및 성능 평가

3.1 문서의 자동 계층정의 모듈의 구현

2장에서 제시한 방법을 활용하여 3장에서는 텍스트 형식의 구조계산서를 준 구조화된 형식인 XML 문서로 변환하는 웹기반 응용모듈을 개발하였다. 그림 4는 개발한 모듈의 동작 사례를 나타낸 것으로, 텍스트 문서의 업로드를 통해 자동으로 문장의 계층을 정의하고, 그것을 이용하여 문서가 XML 형식으로 변환되어 트리구조로 나타나는 것을 확인할 수 있다.

3.2 구현 모듈의 성능 평가

본 연구에서 개발한 모듈의 성능에 영향을 미치는 요인은 2장에 서술된 것처럼 크게 문장의 분리부분과 문장의 계층정의 부분이 있다.

문장의 분리는 전체 문서에 있어 머리기호를 포함하고 있는 문장에 대해 머리기호와 내용을 얼마나 제대로 분리하였는지가 평가 요소가 될 수 있으며, 이는 정밀도(precision)와 정확도(accuracy) 및 recall을 이용하여 정량적인 수치로 나타낼 수 있다. 정밀도는 머리기호가 맞는 항목을 맞게 분리한 경우(TP, True Positive)와 머리기호가 아닌데 분리한 경우(FP, False Positive)대비 TP의 값으로 산출 가능하며, 정확도는 문서의 전체 줄 수 대비 TP와 머리기호가 아니라 분리하지 않은 경우(TN, True Negative)의 합으로 산출 가능하고, recall은 TP와 머리기호가 맞지만 추출하지 못한 경우(FN, False Negative)대비 TP의 값으로 산출 가능하다. 본 연구에서 엔지니어링 문서 18개를 대상으로 한 실험에서 정밀도는 평균 약 98.8%를 나타내

었으며, 정확도의 경우에는 평균 약 99.6%, recall의 경우는 평균 약 97.8%의 높은 수치를 기록했다.

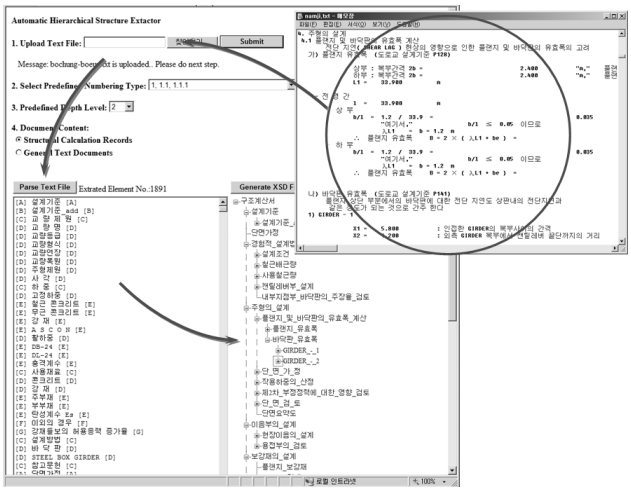

그림 4 개발 모듈의 웹기반 사용자 인터페이스

문장의 계층정의는 앞서 설명한대로 머리기호간의 상대적 차이로 구분하는 것으로, 계층의 정의가 상대적으로 올바르게 수행되었느냐(GAH, Generalized Accuracy for Hierarchy labeler)와 계층의 정의가 절대적인 측면에서도 올바른가(PAH, Precise Accuracy for Hierarchy labeler)의 두 가지로 나누어 성능을 평가 할 수 있으며 이는 모두 TP값을 기준으로 한다. 본 연구 결과에서는 GAH가 평균 약 99.0%, PAH가 평균 약 92.6%로 나타나 이 역시 높은 값을 보였다.

개발한 모듈의 전체적인 성능은 문장의 분리부분과 계층정의가 같이 평가되어야 하며, 이는 TP와 FN대비 계층정의가 상대적으로 올바르게 수행되었느냐(GAA, Generalized Accuracy for Application module)와 절대적으로 올바르게 수행되었느냐(PAA, Precise Accuracy for Application module)로 평가 할 수 있으며, 이들의 값은 각각 평균 약 96.3%와 90.6%의 값을 나타내었다.

4. 결 론

본 연구에서는 비 구조화된 텍스트 형식의 엔지니어링 문서를 준 구조화된 XML 문서로 변환하는 방법론에 대하여 제시하고, 이를 활용한 시험적 모듈의 구현과 함께 실무 엔지니어링 문서를 통한 성능 평가를 실시하였다. 그 결과 최소 정확도 90%가 넘는 수치를 보여 준 구조문서화에 역할을 할 수 있을 것으로 기대된다.

감사의 글

본 연구는 건설교통부에서 실시한 건설핵심기술연구개발사업(교량설계핵심기술연구단)의 연구비 지원에 의해 수행되었으며, 2008년도 교육인적자원부 BK21 사업의 일환인 연세대학교 사화환경시스템공학부 미래사회기반시설 산학연공동사업단의 부분적인 지원을 받았음.

참고문헌

Lee, S.-H., Kim, B.-G., Kim, D.-H. and Jeong, Y.-S. (2006) Development of standardized semantic model for structural calculation documents of bridges and XML schema matching technique, *Proceedings of the 3rd International Conference on Bridge Maintenance Safety and Management (IABMAS)*, pp. 633~634.

Caldas, C.H. and Soibelman, L. (2003) Automating hierarchical document classification for construction management information systems, *Automation in Construction*, Vol. 12, No. 4, pp. 395~406.

Liu, S., McMahon, C.A., Darlington, M.J., Culley, S.J., and Wild, P.J. (2006) A computational framework for retrieval of document fragments based on decomposition schemes in engineering information management, *Advanced Engineering Informatics*, Vol. 20, pp. 401~413.