문서 영상의 논리적인 구조 분석을 위한 문서 모델의 자동 생성

이경호°, 최윤철, 조성배, 고견
연세대학교 컴퓨터과학과, 청주대학교 컴퓨터정보공학과

An Automated Creation of Document Model for Logical Structure Analysis of Document Images

Kyong-Ho Lee, Yoon-Chul Choy, Sung-Bae Cho, Kyun-Koh
Dept. of Computer Science, Yonsei Univ., Dept. Computer and Information Engineering, Chongju Univ.

요 약

본 논문에서는 문서 영상으로부터 전자 문서를 자동 생성하기 위한 논리적인 구조 분석을 효율적으로 지원하기 위하여 문서 모델의 자동 생성과 점증적인 학습 기법을 제안한다. 이를 위하여 문서 유형의 논리적인 구조 정보와 기하적인 특성을 효과적으로 기술할 수 있는 문서 모델을 정의한다. 특히 제안된 방법은 문서 모델의 생성 결과로부터 SGML DTD와 DSSSL 스타일 시트를 생성하기 때문에 문서의 재 사용성과 호환성을 지원한다.

1. 서론

최근 들어 전자 문서의 활발한 보급에도 불구하고 종이 문서의 양도 급속도로 증가하고 있다. 이는 인간이 기본적으로 종이 형태의 문서를 선호한다는 것을 반영하는 것이다. 그러나 종이 문서는 문서 처리의 다양한 면에 있어서 전자 문서보다 비효율적이다. 이에 문서 영상으로부터 논리적인 구성 요소를 추출하여 전자 문서를 생성하는 논리적인 구조분석 방법의 개발이 절실히 요구된다.

특히 문서 영상으로부터 논리적인 계층 구조의 효과적인 추출을 위해서는 문서 유형의 기하적인 특성과 논리적인 계층 구조에 대한 다양한 정보를 표현할 수 있는 문서 모델과 실험 영상으로부터 문서 모델을 자동 생성할 수 있는 방법이 요구된다. 그러나 기존 연구의 대부분 [1] 은 단순한 수준의 문서 모델을 제공하며, 문서 모델의 자동 생성 방법을 지원하지 않는다.

본 논문에서는 문서 모델을 효율적으로 표현할 수 있는 언어인 DSDL (document structure description language)을 제안한다. 또한 본 논문은 문서 모델의 자동 생성과 더불어 기존 모델에 새로운 종류의 문서 구조를 반영할 수 있는 점증적인 학습(incremental learning) 방법을 제안한다. 특히 문서 모델에 포함된 논리적인 구조 정보와 기하적인 특성으로부터 각각 SGML DTD와 DSSSL(document style semantics and specification language) [2] 스타일 시트를 생성하여 문서의 재 사용성과 호환성을 지원한다.

본 논문의 구성은 다음과 같다. 2절에서는 문서 모델을 기술하기 위하여 정의된 DSDL을 설명한다. 3절에서는 문서 모델의 자동 생성 및 점증적인 학습 기법을 기술한다. 마지막으로 4절에서는 결론 및 향후 연구 방향을 기술한다.

2. 문서 모델

문서 모델을 표현하기 위하여 정의된 DSDL은 [그림 1]과 같이 문서 유형의 논리적인 구조 정보와 더불어 텍스트 영역의 기하적인 특성을 효율적으로 기술할 수 있다. 이에 대한 보다 자세한 기술은 본 연구팀이 제안한 논리적인 구조 분석 방법 [3] 에 자세히 기술하였다.

```
<!ELEMENT Document      (Title, Author, Affil, Abstract, Keyword, Sec-Body)>
<!ELEMENT Title         #( FUNTION_TYPE: HEADER
                           COLUMN_TYPE: SINGLE
                           MIN_LINE_HEIGHT: 31
                           MAX_LINE_HEIGHT: 42
                           MIN_LINE_NUMBER: 1
                           MAX_LINE_NUMBER: 3
                           MIN_SPACE_BEFORE: 105
                           MAX_SPACE_BEFORE: 120
                           MIN_SPACE_AFTER: 60
                           MAX_SPACE_AFTER: 70
                           MIN_BLACK_PIXEL_DENSITY: 0.307
                           MAX_BLACK_PIXEL_DENSITY: 0.525
                           JUSTIFY: CENTER )>
<!ELEMENT Author        #( ... )>
<!ELEMENT Affil         #( ... )>
<!ELEMENT Abstract      #( ... )>
<!ELEMENT Keyword       #( ... )>
<!ELEMENT Sec-Body      (Section*, Reference )>
<!ELEMENT Section       (Sec-Header, Paragraph*, Sub-Section*)>
<!ELEMENT Sub-Section   (Sub-Sec-Header, Paragraph*, Sub-Sub-Section*)>
<!ELEMENT Sub-Sub-Section (Sub-Sub-Sec-Header, Paragraph*)>
<!ELEMENT Reference     (Sec-Header, Ref-Item*)>
<!ELEMENT Sec-Header    #( ... )>
<!ELEMENT Sub-Sec-Header #( ... )>
<!ELEMENT Sub-Sub-Sec-Header #( ... )>
<!ELEMENT Paragraph     #( ... )>
<!ELEMENT Ref-Item      #( ... )>
```

[그림 1] 문서 모델의 예

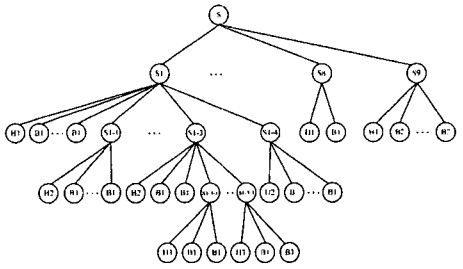
3. 문서 모델의 자동 생성

본 절에서는 실험 영상으로부터 문서 모델의 자동 생성과 점증적인 학습 방법을 기술한다. 제안된 방법은 기능 구조 트리의 생성, 기능 구조 트리의 일반화, 그리고 레이블링 (labeling)의 세 단계로 구성된다. 특히 기능 구조 트리의 생성은 제안된 논리적인 구조 분석 방법 [3]에서 자세히 기술되었다.

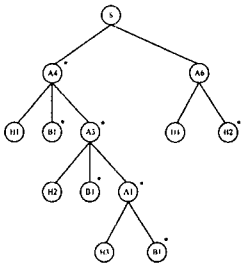
3.1 기능 구조 트리의 일반화

본 절에서는 [그림 2]와 같은 기능 구조 트리로부터 [그림 3]과 같은 일반화된 기능 구조 트리를 생성하는 방법을 기술한다. 이를 위하여 제안된 방법은 문서를 구성하는 구조의 반복적인 특성에 기반한다. 예를 들어, 과학기술 논문의 경우, 본문은 계층적으로 중첩된 다수의 절로 구성된다. 제안된 방법은 특정한 중간 노드가 포함하는 하부 구조를 일반화하기 위하여 반복하는 공통 구조를 식별하고 이를 Kleene *로 표현한다.

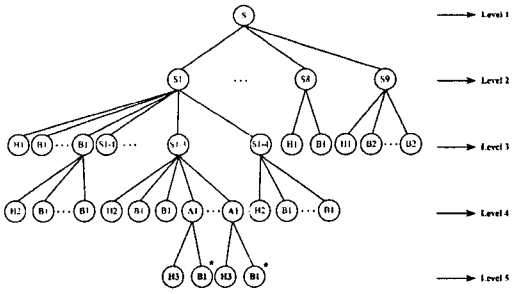
일반적으로 논리적으로 동일한 수준의 절 구조는 기능 구조 트리에서 동일한 레벨(level)에 위치한다. 예를 들어, [그림 2]의 기능 구조 트리에서 절 제목은 2번째 레벨에 위치한다. 따라서 제안된 방법은 기능 구조 트리를 상향식으로 각각의 레벨을 일반화하면서 상위 노드가 포함할 수 있는 하부 구조의 정규 수식을 추출한다.



[그림 2] 기능 구조 트리



[그림 3] 일반화된 기능 구조 트리



[그림 4] [그림 2]에서 레벨 5를 일반화한 결과

예를 들어, [그림 2]의 기능 구조 트리에서 최하위 레벨인 레벨 5를 일반화한 결과는 [그림 4]와 같다. [그림 2]에서 최하위 레벨에 위치하는 노드는 H3과 B1로 구성된 순차적인 집합에 해당한다. 본 논문은 반복적인 노드의 집합을 Kleene *로 표현한 결과를 일반화 패턴(generalized pattern)이라고 정의한다. 예를 들어, S1-2-1 = (H3, B1, B1, B1)로부터 반복적인 패턴 (B1, B1, B1)을 (B1*)로 대체하여 일반화 패턴 (H3, B1*)을 생성한다. 실제로 일반화 패턴은 상위 노드가 포함할 수 있는 자식 노드의 종류와 순서 그리고 빈도

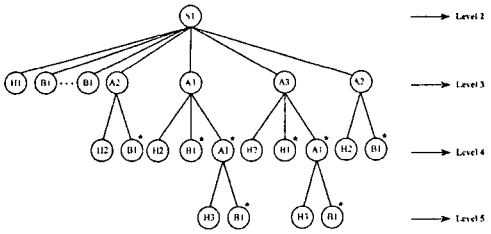
수에 대한 정규 수식에 해당한다.

제안된 방법은 일반화 패턴을 효과적으로 추출하기 위하여 자식 노드에 대한 동일한 일반화 패턴을 갖는 부모 노드는 동일한 이름의 보조 기호(auxiliary symbol)로 표현한다. 특히 본 논문은 동일한 헤더와 바디로 시작하는 두 패턴 (a, b)와 (a, b*)를 동일한 형태로 간주한다. 따라서 [그림 4]에서 일반화 패턴 (H3, B1*)와 (H3, B1)은 보조 기호 A1으로 대체된다. 또한 레벨 4에 속하는 노드의 집합으로부터 반복적인 패턴 (B1, ..., B1)과 (A1, ..., A1)을 식별하고, 이를 각각 (B1*)와 (A1*)로 대체하여 [그림 5]와 같은 일반화 패턴을 추출한다.

S1-1 = (H2, B1*), S1-2 = (H2, B1*, A1*), S1-3 = (H2, B1*, A1*), S1-4 = (H2, B1*)
S2-1 = (H2, B1, A1*), S2-2 = (H2, B1*, A1*), S2-3 = (H2, B1*)
S3-1 = (H2, B1*, A1*), S3-2 = (H2, B1*, A1*), S3-3 = (H2, B1)
S5-1 = (H2, B1*), S5-2 = (H2, B1*), S5-3 = (H2, B1), S5-4 = (H2, B1*, A1)

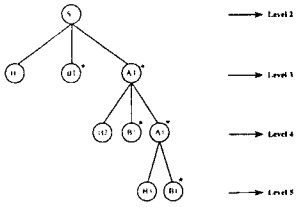
[그림 5] 레벨 4의 일반화 결과

마찬가지로 레벨 4의 결과를 바탕으로 레벨 3을 일반화한다. 예를 들어, S1 = (H1, B1, B1, B1, B1, B1, A2, A3, A3, A2)의 일반화 과정은 다음과 같다. 전술한 바와 같이 제안된 방법은 먼저 동일한 모양의 반복적인 패턴 (B1, ..., B1)과 (A3, ..., A3)를 식별하여 [그림 6]과 같은 정규 수식 S1 = (H1, B1*, A2, A3*, A2)을 생성한다.



[그림 6] S1을 부분적으로 일반화한 결과

일반적으로 문서를 구성하는 절은 하위 레벨의 절을 선택적으로 포함할 수 있다. 따라서 자식 노드의 패턴이 각각 (S)와 (S, C)인 두 노드 A와 B는 (B*)로 일반화한다. 여기서 S는 단일의 헤더 또는 헤더와 한 개 이상의 동일한 종류의 바디로 구성된 순차적인 집합이며 C는 보조 기호로 표현된 하위 레벨의 절 구조에 해당한다. 예를 들어, A2와 A3는 하부 구조에 해당하는 일반화 패턴으로써 모두 공통적인 접두사 A2를 포함한다. 따라서 제안된 방법은 [그림 7]과 같이 정규 수식 (A2, A3*, A2)를 A3*로 통합하여 S1의 일반화 패턴으로써 (H1 B1* A3*)를 추출한다. [그림 2]의 기능 구조 트리를 레벨 3까지 일반화한 결과는 [그림 8]와 같다.



[그림 7] S1의 일반화 결과

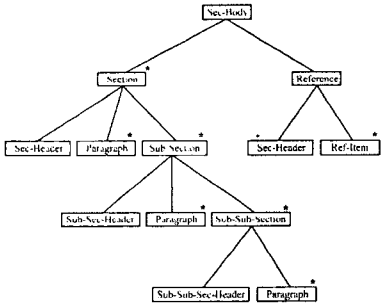
S1 = S2 + S3 = A4, S4 = A5, S5 = A4, S6 = S7 = S8 = A5, S9 = A6,
where A4 = (H1 B1* A3*), A5 = (H1 B1*), and A6 = (H1 B2*)

[그림 8] 레벨 3의 일반화 결과

마지막으로 노드 S의 자식 노드인 (S1, S2, S3, S4, S5, S6, S7, S8, S9)의 일반화 과정은 다음과 같다. 먼저 반복적인 패턴 (S1, S2, S3)과 (S6, S7, S8)을 식별하여 일반화 패턴 (A4* A1 A4 A1* A5)를 생성한다. 또한 제안된 방법을 적용하여 최종적으로 [그림 3]과 같이 일반화 패턴 (A4* A5)을 생성한다.

3.2 레이블링

본 단계는 일반화된 기능 트리의 노드에 논리적인 의미를 부여하는 단계이다. 이를 위하여 제안된 방법은 사용자로부터 각각의 노드에 대한 적절한 레이블을 입력 받는다. 예를 들어, [그림 3]에 레이블을 부여한 결과는 [그림 9]와 같다.



[그림 9] 레이블을 부여한 결과

3.3 문서 모델의 생성

제안된 방법은 각각의 중간 노드에 대하여 자식 노드에 대한 일반화 패턴 정보를 이용하여 문서 모델을 생성한다. 또한 각각 단말 노드가 포함하는 텍스트 영역의 기하적인 속성 값을 이용하여 해당 구조에 대한 기하적인 조건을 기술

한다. 제안된 방법이 레이블이 부여된 일반화된 기능 구조 트리에 너비 우선 탐색(breadth-first traversal)을 적용하여 추출한 문서 모델은 [그림 10]과 같다.

```
<!ELEMENT Sec-Body  (Section*, Reference)>
<!ELEMENT Section   (Sec-Header, Paragraph*, Sub-Section*)>
<!ELEMENT Sub-Section  (Sub-Sec-Header, Paragraph*, Sub-Sub-Section*)>
<!ELEMENT Sub-Sub-Section (Sub-Sub-Sec-Header, Paragraph*)>
<!ELEMENT Reference  (Sec-Header, Ref-Item*)>
```

[그림 10] [그림 9]에서 추출된 문서 모델

일반적으로 문서 모델은 다수의 학습 데이터로부터 생성된다. 한편 문서 유형에 속하는 새로운 문서의 구조를 반영하기 위하여 문서 모델에 대한 점증적인 학습 방법이 요구된다. 제안된 방법은 이러한 문제에 대한 효과적인 해결 방법을 제공한다.

먼저 다수의 실험 데이터로부터 문서 모델을 생성하기 위하여 제안된 방법은 먼저 중간 노드 각각에 대하여 가능한 문서 모델을 통합한다. 특히 보다 간결한 형태의 정규 수식을 추출하기 위하여 인수 분해(factorization) 기법을 적용한다. 예를 들어, 내용 모델 (a c), (a d), (b c), 그리고 (b d)에 인수 분해 과정을 적용한 결과는 ((a | b) | (c | d))에 해당한다. 본 논문은 이를 위하여 기존에 논리 함수 (boolean function)의 최적화 분야에서 사용하는 알고리즘 [4]를 적용한다. 마찬가지로 제안된 방법은 새로운 문서의 일반화 결과를 위와 같은 방법으로 기존 문법에 간단히 통합할 수 있기 때문에 문서 모델의 점증적인 학습을 지원한다.

```
<!ELEMENT Sec-Body          - -  (Section*, Reference)>
<!ELEMENT Section           - -  (Sec-Header, Paragraph*, Sub-Section*)>
<!ELEMENT Sub-Section       - -  (Sub-Sec-Header, Paragraph*, Sub-Sub-Section*)>
<!ELEMENT Sub-Sub-Section   - -  (Sub-Sub-Sec-Header, Paragraph*)>
<!ELEMENT Reference         - -  (Sec-Header, Ref-Item*)>
<!ELEMENT Sec-Header        - -  (#PCDATA)>
<!ELEMENT Sub-Sec-Header    - -  (#PCDATA)>
<!ELEMENT Sub-Sub-Sec-Header - - (#PCDATA)>
<!ELEMENT Paragraph         - -  (#PCDATA)>
<!ELEMENT Ref-Item          - -  (#PCDATA)>
```

[그림 11] 문서의 모델을 SGML DTD 로 표현한 결과

```
(element Sec-Header (make paragraph   font-size: 24pt
                                      font-weight: 'bold
                                      quadding: 'left
                                      line-spacing: 24pt
                                      space-before: 12pt
                                      space-after: 12pt ))
(element Paragraph  (make paragraph   font-size: 24pt
                                      font-weight: 'medium
                                      quadding: 'left
                                      first-line-start-indent: (if (first-sibling?)
                                                                   0pt
                                                                   Paragraph-Indent)
                                      space-before: 0pt
                                      space-after: 0pt
                                      color: (RGB-COLOR 0. 0. 0.)))
```

[그림 13] DSSSL 스타일 시트의 일부

제안된 문서 모델은 기하적인 속성 정보는 물론이고 논리적인 스키마를 유지한다. 본 논문은 전자 문서의 표준 포맷인 SGML과 DSSSL을 지원하기 위하여 문서 모델로부터 SGML DTD와 DSSSL 스타일 시트를 생성한다. 이를 위하여 제안된 방법은 문서 모델에 너비 우선 탐색 과정을 적용하여 SGML DTD를 생성한다.

또한 단말 노드에 해당하는 구성 요소가 포함하는 기하적인 특성으로부터 DSSSL 스타일 시트를 생성한다. 예를 들어, [그림 9]의 문서 모델에 포함된 논리적인 구조와 기하적인 특성을 SGML DTD와 DSSSL 스타일 시트로 표현한 결과는 각각 [그림 11]과 [그림 12]와 같다.

4. 결론 및 향후 연구 방향

본 논문에서는 다수의 문서 영상으로 구성된 복잡한 구조의 문서를 대상으로 효율적인 구조분석을 지원하기 위하여 문서 모델의 자동 생성과 더불어 기존 모델에 새로운 종류의 문서 구조를 반영할 수 있는 점증적인 학습 방법을 제안한다. 이를 위하여 문서 모델을 효율적으로 표현할 수 있는 언어인 DSDL을 제안한다. 또한 문서 모델에 포함된 논리적인 구조 정보와 기하적인 특성으로부터 각각 SGML DTD와 DSSSL 스타일 시트를 생성하기 때문에 문서의 재사용성과 호환성을 높인다.

향후 본 연구에서는 문서 모델의 자동 생성 기법을 평가하기 위하여 자동 생성된 문서 모델을 전문가의 수작업에 의하여 생성된 문서 모델과 비교하기 위한 체계적인 평가 기준을 개발한 계획이다.

참고 문헌

[1] M. Krishnamoorthy, G. Gagy, S. Seth, and M. Viswanathan, "Syntactic Segmentation and Labeling of Digitized Pages from Technical Journals," *IEEE Trans. Pattern Analysis ans Machine Intelligence*, Vol. 15, No. 7, pp. 737 ~ 747, Jul. 1993.

[2] International Organization for Standardization, Information Technology - Text and Office Systems - Document Style Semantics and Specification (DSSSL), *ISO/IEC 10179*, 1996.

[3] 이경호, 최윤철, 조성배, "구조화된 문서 생성을 위한 논리적인 구조 분석 기법: 구문론적인 접근 방식", 정보과학회 가을 학술발표논문집, 2000.

[4] A. R. R. Wang, *Algorithms for Multi-level Logic Optimization*. Ph.D. Thesis, The University of California, Berkeley, 1989.