(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년03월20일
(11) 등록번호 10-1376863
(24) 등록일자 2014년03월14일

(51) 국제특허분류(Int. Cl.)
G06F 17/20 (2006.01) *G06K 9/72* (2006.01)
(21) 출원번호 10-2007-7030734
(22) 출원일자(국제) 2006년06월30일
심사청구일자 2011년06월02일
(85) 번역문제출일자 2007년12월28일
(65) 공개번호 10-2008-0026128
(43) 공개일자 2008년03월24일
(86) 국제출원번호 PCT/US2006/026140
(87) 국제공개번호 WO 2007/005937
국제공개일자 2007년01월11일
(30) 우선권주장
11/173,280 2005년07월01일 미국(US)
(56) 선행기술조사문헌
US05987171 A*
US20030130992 A1*
X.Chen et al., "Detecting and reading text in natural scenes." Proc. of the Computer Vision and Pattern Recognition 2004, 27 July 2004, vol. 2, pp. 366-373, ISSN : 1063-6919.*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
비올라, 폴 에이.
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
실맨, 마이클
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
(74) 대리인
제일특허법인

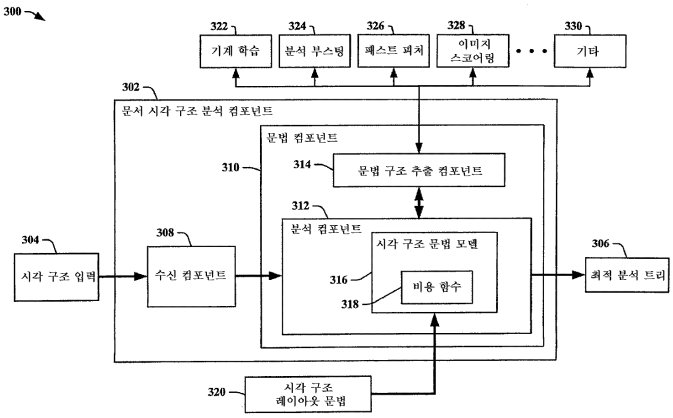
전체 청구항 수 : 총 19 항 심사관 : 임지환

(54) 발명의 명칭 **문서 시각 구조의 문법 분석**

(57) 요 약

문서의 2차원 표현이 문서의 인식을 돕는 계층 구조를 추출하는 데 이용된다. 문서의 시각 구조는 통계 분석 알고리즘의 2차원 적응을 이용하여 문법적으로 분석된다. 이것은 레이아웃 구조(예를 들어, 칼럼, 저자, 타이틀, 각주 등) 등의 인식을 가능하게 하여 문서의 구조적 성분들을 정확하게 해석할 수 있게 한다. 문서 레이아웃 인식을 돕기 위해 추가적인 기술들도 이용될 수 있다. 예를 들어, 기계 학습, 이미지 표현에 기초한 분석 스코어링, 부스팅 기술, 및/또는 "패스트 피처(fast feature)" 등을 이용하는 문법 분석 기술을 이용하여 문서 인식을 도울 수 있다.

대 표 도



특허청구의 범위

청구항 1

문서 레이아웃 구조의 인식을 돕는 시스템으로서,

프로세서를 포함하되, 상기 프로세서는,

문서의 시각 구조(visual structure)와 연관된 입력을 수신하는 수신 컴포넌트; 및

상기 문서의 시각 구조의 차별(discriminative) 문법 모델을 적어도 부분적으로 이용하여, 상기 문서의 시각 구조 내에서 식별되는 복수의 심벌 유형에 복수의 문법 규칙을 연관시킴으로써, 상기 입력의 문법적 분석(grammatically parsing)을 돕는 문법 컴포넌트

를 포함하는 컴퓨터 실행가능 컴포넌트들을 실행하고,

상기 문법 규칙은 자연 언어의 단락 및 상기 단락의 서브파트 간의 관계를 포함하고 또한 수학적 표현 및 상기 수학적 표현의 서브파트 간의 관계를 포함하고, 상기 차별 문법 모델은 상기 문서의 페이지, 섹션, 칼럼, 단락, 라인, 저자, 타이틀, 각주 또는 단어 중 적어도 하나와 연관된 계층 정보를 포함하며, 상기 입력의 문법적 분석은 적어도 부분적으로 기계 학습(machine learning) 기술을 통해 도출되는 문법 비용 함수(grammatical cost function)에 적어도 부분적으로 기초하되, 상기 기계 학습 기술은 상기 문서의 전역 검색으로부터 상기 문서의 최적 분석 트리를 결정하는 것을 돕고,

상기 입력의 문법적 분석은,

상기 문서의 시각 구조를 복수의 적분 이미지(integral images)로서 표현하는 것;

상기 최적 분석 트리를 결정하는 것의 분석 효율을 향상시키기 위해 상기 복수의 적분 이미지의 복수의 배열(constellations)을 이용하는 것 - 상기 복수의 배열은 화이트 스페이스 직사각형 내에 적어도 하나의 문서 피처(document feature)를 포함함 -; 및

상기 문법 비용 함수에 의한 상기 최적 분석 트리의 결정을 돕기 위해 상기 복수의 적분 이미지를 스코어링(scoring)하는 것을 포함하는,

인식을 돕는 시스템.

청구항 2

제1항에 있어서,

상기 문법 컴포넌트는 국지 피처(local features) 및 전역 피처(global features) 중 적어도 하나를 이용하여 상기 문서와 연관된 레이아웃 구조를 추출하는 문서 구조 추출 컴포넌트를 더 포함하는, 인식을 돕는 시스템.

청구항 3

제2항에 있어서,

상기 문서 구조 추출 컴포넌트는 상기 문서 레이아웃 구조의 추출을 돕기 위해 이미지 스코어링(image scoring), 분석 학습 부스팅(parse learning boosting) 및 패스트 피처(fast features) 중 적어도 하나를 이용하는, 인식을 돕는 시스템.

청구항 4

제3항에 있어서,

상기 문법 컴포넌트는 상기 전역 검색으로부터 상기 최적 분석 트리를 결정하는 것을 돕기 위해 적어도 하나의 분류자(classifier)를 이용하는 분석 컴포넌트(parsing component)를 더 포함하는, 인식을 돕는 시스템.

청구항 5

제4항에 있어서,

상기 분석 컴포넌트는 문법 비용 함수의 결정을 돕기 위해 상기 분류자를 이용하는, 인식을 돕는 시스템.

청구항 6

제5항에 있어서,

상기 분류자는 통상의 기계 학습 기술을 통해 훈련된 분류자를 포함하는, 인식을 돕는 시스템.

청구항 7

제6항에 있어서,

상기 기계 학습 기술은 퍼셉트론 기반 기술(perceptron-based technique)을 적어도 부분적으로 포함하는, 인식을 돕는 시스템.

청구항 8

제2항에 있어서,

상기 문서 구조 추출 컴포넌트는 상기 문서 레이아웃 구조를 추출하는 것을 돕기 위해 기계 학습을 이용하는, 인식을 돕는 시스템.

청구항 9

제1항에 있어서,

상기 문법 컴포넌트는 동적 프로그래밍을 적어도 부분적으로 이용하여 상기 입력에 대한 전역적으로 최적인 분석 트리를 결정하는, 인식을 돕는 시스템.

청구항 10

컴퓨터 실행가능 명령어를 포함하는 컴퓨터 판독가능 저장 매체로서,

상기 컴퓨터 실행가능 명령어는 컴퓨터 상에서 실행되는 경우 문서 레이아웃 구조의 인식을 돕는 것을 포함하는 액트를 수행하되, 상기 문서 레이아웃 구조의 인식을 돕는 것은,

문서의 시각 구조와 연관된 입력을 수신하는 것; 및

문법 분석 프로세스를 상기 문서의 시각 구조의 추론에 적용하는 것을 포함하고,

상기 문서의 시각 구조의 추론은 상기 문서의 추론되는 페이지, 섹션, 칼럼, 단락, 라인 또는 단어 중 적어도 하나를 포함하며, 상기 문법 분석 프로세스는 문법 규칙에 기초하되, 상기 문법 규칙은 자연 언어의 단락 및 상기 자연 언어의 단락의 서브파트 간의 관계를 포함하고 또한 수학적 표현 및 상기 수학적 표현의 서브파트 간의 관계를 포함하고,

상기 문법 분석 프로세스는,

적어도 부분적으로 기계 학습(machine learning) 기술 - 상기 기계 학습 기술은 상기 문서의 전역 검색으로부터 상기 문서의 최적 분석 트리를 결정하는 것을 도움 - 을 통해 도출되는 문법 비용 함수에 적어도 부분적으로 기초하여 상기 입력을 분석하는 것;

상기 문서의 시각 구조를 복수의 적분 이미지들로서 표현하는 것;

상기 최적 분석 트리를 결정하는 것의 분석 효율을 향상시키기 위해 상기 복수의 적분 이미지들의 복수의 배열들(constellations)을 이용하는 것 - 상기 복수의 배열들은 화이트 스페이스 직사각형 내에 적어도 하나의 문서 피처를 포함함 -; 및

상기 문법 비용 함수에 의한 상기 최적 분석 트리의 결정을 돕기 위해 상기 복수의 적분 이미지들을 스코어링하는 것을 포함하는,

컴퓨터 판독가능 저장 매체.

청구항 11

제10항에 있어서,

상기 문법 레이아웃 구조의 인식을 돕는 것은 상기 문서의 시각 구조의 차별 문법 모델을 적어도 부분적으로 이용하여 상기 문서의 시각 구조의 입력의 문법적 분석을 돕는 것을 더 포함하되, 상기 차별 문법 모델은 상기 문서의 페이지, 섹션, 칼럼, 단락, 라인 또는 단어 중 적어도 하나와 연관된 계층 정보를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 12

제10항에 있어서,

상기 문법 분석 프로세스는 차별 문법 모델에 기초하는, 컴퓨터 판독가능 저장 매체.

청구항 13

컴퓨터로 구현되는(computer-implemented) 문서 레이아웃 구조 인식 방법으로서,

프로세서 및 메모리에 연결된 입력 장치로부터 상기 메모리에 저장된 문서의 시각 구조와 연관된 입력을 수신하는 단계; 및

상기 메모리에 저장된 상기 문서의 시각 구조의 문법 모델을 적어도 부분적으로 이용하여 상기 입력의 문법적 분석(grammatically parsing)을 돕는 단계 - 상기 문법 모델은 상기 문서의 페이지, 섹션, 칼럼, 단락, 라인 또는 단어 중 적어도 하나와 연관된 계층 정보를 포함하되, 문법 규칙은 자연 언어의 단락 및 상기 자연 언어의 단락의 서브파트 간의 관계를 포함하고 또한 수학적 표현 및 상기 수학적 표현의 서브파트 간의 관계를 포함하고, 상기 입력의 문법적 분석은 적어도 부분적으로 기계 학습 기술을 통해 도출되는 문법 비용 함수에 적어도 부분적으로 기초하되, 상기 기계 학습 기술은 상기 문서의 전역 검색으로부터 상기 문서의 분석 트리를 결정하는 것을 도움 - ;

상기 문서의 시각 구조를 복수의 적분 이미지로서 표현하는 단계; 및

상기 분석 트리를 결정하는 것의 분석 효율을 향상시키기 위해 상기 복수의 적분 이미지의 복수의 배열을 이용하는 단계 - 상기 복수의 배열은 상기 문서 내 화이트 스페이스 직사각형 내에 적어도 하나의 문서 피처를 포함함 - 를 포함하는,

컴퓨터로 구현되는 문서 레이아웃 구조 인식 방법.

청구항 14

제13항에 있어서,

기계 학습 기술을 통하여 훈련된 적어도 하나의 분류자를 이용하여 상기 입력을 분석하는 단계를 더 포함하는,

컴퓨터로 구현되는 문서 레이아웃 구조 인식 방법.

청구항 15

제10항에 있어서,

상기 문법 레이아웃 구조의 인식을 돕는 것은 적어도 하나의 문서 피처의 상기 적분 이미지들 중 적어도 하나 또는 상기 복수의 적분 이미지들의 상기 배열들 중 적어도 하나를 계산하여 상기 입력을 분석하는 것을 돕는 것을 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 16

제10항에 있어서,

상기 문법 레이아웃 구조의 인식을 돕는 것은 상기 입력의 분석을 돕기 위해 에이다부스트(AdaBoost)를 이용하는 것을 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 17

문서 레이아웃 구조의 인식을 돕는 시스템으로서,

프로세서; 및

상기 프로세서에 통신 가능하게 연결된 메모리를 포함하되,

상기 메모리는,

문서의 시각 구조와 연관된 입력을 수신하는 수신 컴포넌트 - 상기 시각 구조는 상기 문서의 픽처, 단락, 칼럼, 섹션, 수학식, 저자, 타이틀, 텍스트의 배향, 스페이싱 또는 포맷팅 중 적어도 하나와 연관됨 - ;

상기 문서의 시각 구조의 차별 문법 모델을 적어도 부분적으로 이용하여, 상기 문서의 시각 구조 내에서 식별되는 복수의 심벌 유형에 복수의 문법 규칙을 연관시킴으로써, 상기 입력의 문법적 분석을 돕는 문법 컴포넌트 - 각 심벌 유형은 말단(terminal)을 기술하는 연관된 문법 규칙을 가지되, 상기 말단은 텍스트의 심벌, 숫자 또는 문자를 포함하고, 상기 문법 규칙은 자연 언어의 단락 및 상기 자연 언어의 단락의 서브파트 간의 관계를 포함하고 또한 수학적 표현 및 상기 수학적 표현의 서브파트 간의 관계를 포함하고, 상기 차별 문법 모델은 상기 문서의 페이지, 섹션, 칼럼, 단락, 라인 또는 단어 중 적어도 하나와 연관된 계층 정보를 포함하며, 상기 입력의 문법적 분석은 적어도 부분적으로 기계 학습(machine learning) 기술을 통해 도출되는 문법 비용 함수에 적어도 부분적으로 기초하되, 상기 기계 학습 기술은 상기 문서의 전역 검색으로부터 상기 문서의 분석 트리를 결정하는 것을 도움 - ;

국지 피처 또는 전역 피처 중 적어도 하나를 이용하여 상기 문서와 연관된 레이아웃 구조를 추출하는 문서 구조 추출 컴포넌트 - 상기 문서 구조 추출 컴포넌트는 상기 문서 레이아웃 구조를 추출하는 것을 돕기 위해 기계 학습을 이용함 -; 및

상기 문서의 시각 구조의 요소에 대응하는 복수의 적분 이미지를 계산하고, 상기 분석 트리를 결정하는 것의 분석 효율을 향상시키기 위해 상기 복수의 적분 이미지의 복수의 배열을 이용하는 패스트 피처 메카니즘 - 상기 복수의 배열은 상기 문서 내 화이트 스페이스 직사각형 내에 적어도 하나의 문서 피처를 포함함 -

을 포함하여 상기 문서 레이아웃 구조의 인식 시스템을 구현하도록 구성된 컴퓨터 실행가능 명령어가 저장되어 있는,

시스템.

청구항 18

삭제

청구항 19

제10항의 컴퓨터 판독가능 저장 매체를 이용하는 장치로서,

상기 장치는 컴퓨터, 서버 또는 핸드헬드 전자 장치를 포함하는, 장치.

청구항 20

문서 시각 구조의 임포팅(importing) 및 익스포팅(exporting) 중 적어도 하나를 돕기 위해 제1항의 시스템을 이용하는, 인식 시스템.

명 세 서

배 경 기 술

[0001] 시간이 지남에 따라, 사람들은 일 및 여가 활동 양쪽을 지원하기 위해 컴퓨터에 더욱 의존하고 있다. 그러나 컴퓨터는 정보를 처리하기 위해 개별 상태들이 식별되어야 하는 디지털 도메인에서 동작한다. 이것은 사건들이 완전히 흑 또는 백이 아니라 회색 음영들 사이에 있는 뚜렷한 아날로그 방식으로 동작하는 사람들과는 대조적이다. 따라서, 디지털과 아날로그 사이의 중요한 차이는 디지털이 시간에 따라 분리되는 개별 상태들(예를 들어, 개별 레벨들)을 필요로 하는 반면, 아날로그는 시간에 따라 연속적이라는 점이다. 사람들은 본질적으로 아날로그 방식으로 동작하므로, 컴퓨팅 기술은 전술한 시간적 차이에 의해 발생하는 사람과 컴퓨터 간의 인터페이싱(예를 들어, 디지털 컴퓨팅 인터페이스)과 연관된 곤란성을 완화하도록 발전해왔다.

[0002] 기술은 먼저 기존의 타이핑 또는 식자된 정보의 컴퓨터로의 입력의 시도에 집중하였다. 처음에는 픽처들의 "디

지털화"(예를 들어, 이미지의 컴퓨터 시스템으로의 입력)를 위해 스캐너 또는 광학 이미저가 사용되었다. 이미지가 컴퓨터 시스템 안으로 디지털화될 수 있다면, 이에 따라 인쇄 또는 식자된 자료도 디지털화될 수 있어야 한다. 그러나 스캐닝된 페이지의 이미지는 컴퓨팅 시스템 내로 입력된 후에는 텍스트 또는 심벌로서 조작될 수 없는데, 이는 이러한 이미지가 컴퓨터에 의해 "인식"되지 못하기 때문, 즉 시스템이 페이지를 이해하지 못하기 때문이다. 문자 및 단어는 실제로 편집할 수 있는 텍스트 또는 심벌이 아니라 "픽처"이다. 텍스트에 대한 이러한 제한을 극복하기 위하여, 스캐닝 기술을 이용하여 텍스트를 편집 가능한 페이지로 디지털화하는 광학 문자 인식(OCR) 기술이 개발되었다. 이 기술은 OCR 소프트웨어가 스캐닝된 이미지를 편집 가능한 텍스트로 변환하는 것을 가능하게 하는 특정 텍스트 폰트를 사용한 경우에 상당히 잘 동작하였다.

[0003] 텍스트가 컴퓨팅 시스템에 의해 "인식"되었더라도, 그러한 프로세스에 의해 중요한 정보가 손실되었다. 이 정보는 텍스트의 포맷팅, 텍스트의 스페이싱, 텍스트의 배향, 및 일반적인 페이지 레이아웃 등과 같은 것을 포함하였다. 따라서, 페이지가 우상 코너에서 픽처와 함께 이중 칼럼화된 경우, OCR 스캐닝된 이미지는 워드 프로세서에서 이중 칼럼 및 픽처 없이 텍스트의 그룹핑이 되었을 것이다. 또는, 픽처가 포함된 경우, 일반적으로 픽처는 텍스트들 사이의 소정의 임의 포인트에 삽입되었다. 이것은 상이한 문서 작성 표준들이 사용될 때 훨씬 더 큰 문제가 된다. 통상적인 OCR 기술은 일반적으로 다른 문서 표준으로부터 구조를 "변환"하거나 적절히 인식할 수 없다. 대신에, 결과적인 인식은 인식된 부분들을 그의 관련 표준으로 한정하거나 강제하려고 시도한다. 이것이 발생할 때, OCR 프로세스는 통상적으로 물음표와 같은 "미지"의 마커를 인식된 부분들 내에 입력하여, 문서의 이들 성분을 처리할 수 없음을 표시한다.

[0004] <발명의 요약>

[0005] 다음은 본 발명의 실시예들의 몇몇 양태의 기본적 이해를 제공하기 위해 본 발명의 간단한 요약을 제공한다. 이 요약은 본 발명의 포괄적인 개요는 아니다. 이것은 실시예들의 주요/임계 요소들을 식별하거나 본 발명의 범위를 기술하는 것을 의도하지 않는다. 그 유일한 목적은 후술하는 상세한 설명에 대한 서론으로서 본 발명의 몇몇 개념을 간단한 형태로 제공하는 것이다.

[0006] 문법 분석을 이용하여 문서 구조의 인식을 돕는 시스템 및 방법이 제공된다. 문서의 2차원 표현이 문서의 인식을 돕는 계층 구조를 추출하는 데 이용된다. 문서의 시각 구조는 통계 분석 알고리즘의 2차원 적응을 이용하여 문법적으로 분석된다. 이것은 레이아웃 구조(예를 들어, 칼럼, 저자, 타이틀, 각주 등) 등의 인식을 가능하게 하여 문서의 구조적 성분들을 정확하게 해석할 수 있게 한다. 문서 레이아웃 인식을 돕기 위해 추가적인 기술들도 이용될 수 있다. 예를 들어, 기계 학습, 이미지 표현에 기초한 분석 스코어링, 부스팅 기술, 및/또는 "패스트 피처(fast feature)" 등을 이용하는 문법 분석 기술을 이용하여 문서 인식을 도울 수 있다. 이것은 상당히 정확도가 개선된 효율적인 문서 인식을 제공한다.

[0007] 상기 및 관련 목적의 달성을 위해, 본 명세서에서 실시예들의 소정의 예시적인 양태들이 아래의 설명 및 첨부 도면들과 관련하여 기술된다. 그러나 이들 양태는 본 발명의 원리가 이용될 수 있는 다양한 방법 중 일부를 나타내며, 본 발명은 그러한 모든 양태 및 그 균등물을 포함하는 것을 의도한다. 본 발명의 다른 이점 및 신규한 특징은 도면들과 관련하여 고려할 때 아래의 상세한 설명으로부터 명백해질 수 있다.

발명의 상세한 설명

[0018] 이제, 도면들을 참조하여 본 발명을 설명하는데, 도면 전반에서 동일한 참조 번호는 동일한 요소를 지칭하는 데 사용된다. 아래의 설명에서는 설명의 목적으로, 본 발명의 충분한 이해를 제공하기 위해 다양한 특정 상세가 설명된다. 그러나 본 발명의 실시예들은 이러한 특정 상세들 없이도 실시될 수 있음이 명백할 수 있다. 다른 예들에서, 실시예들의 설명을 돕기 위해 공지 구조 및 장치는 블록도 형태로 도시된다.

[0019] 본 출원에서 사용되는 "컴포넌트"라는 용어는 컴퓨터 관련 엔티티, 즉 하드웨어, 하드웨어와 소프트웨어의 조합, 소프트웨어 또는 실행중인 소프트웨어를 지칭하는 것을 의도한다. 예를 들어, 컴포넌트는 프로세서 상에서 실행되는 프로세스, 프로세서, 개체, 실행 파일, 실행 스레드, 프로그램 및/또는 컴퓨터일 수 있지만 이에 제한되는 것은 아니다. 예를 들어, 서버 상에서 실행되는 애플리케이션 및 서버 양자는 컴퓨터 컴포넌트일 수 있다. 하나 이상의 컴포넌트가 프로세스 및/또는 실행 스레드 내에 위치할 수 있으며, 하나의 컴포넌트는 하나의 컴퓨터 상에 국한되고, 그리고/또는 둘 이상의 컴퓨터 사이에 분산될 수 있다. "스레드"는 운영 체제 커널이 실행을 위해 스케쥴링하는 프로세스 내의 엔티티이다. 이 분야에 공지되어 있듯이, 각각의 스레드는 스레드의 실행과 연관된 휘발성 데이터인 관련 "문맥"을 갖는다. 스레드의 문맥은 시스템 레지스터의 내용 및 스레드의 프로세스에 속하는 가상 어드레스를 포함한다. 따라서, 스레드의 문맥을 포함하는 실제 데이터는 스레드가

실행됨에 따라 변한다.

[0020] 시각 구조의 이용을 통해 문서의 인식을 돕는 시스템 및 방법이 제공된다. 문서의 고유 계층 구조(예를 들어, 문서->페이지->섹션->칼럼->단락 등)는 문법 기반 기술을 이용하는 2차원 분석 메카니즘을 이용하여 인식된다. 문법 분석 메카니즘과 함께 기계 학습 프로세스를 추가로 이용함으로써, 높은 정확도를 계속 제공하면서 문서 인식 효율이 크게 개선될 수 있다. 분석 속도 및 효율의 향상을 돕기 위해 이미지 스코어링 기술도 이용될 수 있다. 문서의 패스트 피처의 선택은 물론 분석 학습을 위한 부스팅 기술도 시스템 및 방법의 생산성을 향상시키는 데 이용될 수 있다.

[0021] 문법 분석은 컴퓨터 언어 및 자연 언어를 처리하는 데 이용된다. 컴퓨터 언어의 경우, 문법은 명백하며, 입력이 주어지면, 하나의 유효한 분석만이 존재한다. 자연 언어의 경우, 문법은 불명료하며, 입력 시퀀스가 주어지면, 매우 많은 수의 잠재적인 분석이 존재한다. 통계적 자연 언어 분석에서의 소망은 기계 학습을 이용하여, 정확한 분석에 최고의 스코어를 부여하는 스코어링 함수를 산출하는 것이다. 본 명세서에 제공되는 시스템 및 방법에서는, 가시 구조 레이아웃이 문법으로서 모델링되며, 최적 분석을 위한 전역 검색이 문법 비용 함수에 기초하여 수행된다. 이어서, 기계 학습을 이용하여, 피처들을 차별적으로 선택하고, 다양한 시각 구조 레이아웃에 적응하는 문법 분석 프로세스의 모든 파라미터를 설정할 수 있다.

[0022] 도 1에는, 일 실시 양태에 따른 문서 시각 구조 분석 시스템(100)의 블록도가 도시되어 있다. 문서 시각 구조 분석 시스템(100)은 입력(104)을 수신하고 출력(106)을 제공하는 문서 시각 구조 분석 컴포넌트(102)를 포함한다. 문서 시각 구조 분석 컴포넌트(102)는 문서의 시각 구조 레이아웃의 비생성 문법 모델을 이용하여 시각 구조 레이아웃에 대한 최적의 분석 트리의 결정을 돕는다. 입력(104)은 예를 들어 문서의 페이지의 시각 레이아웃을 포함한다. 문서 시각 구조 분석 컴포넌트(102)는 문서의 시각 구조를 분석하는 문법 분석 프로세스를 이용하여 입력(104)을 분석하여 출력(106)을 제공한다. 출력(106)은 예를 들어 문서 시각 구조 레이아웃에 대한 최적의 분석 트리를 포함할 수 있다. 추가적인 문법 학습을 필요로 하지 않고 상이한 태스크들에 대한 분석 솔루션들을 제공하는 전역적으로 학습된 "참조" 문법도 확보될 수 있다.

[0023] 도 2에는 일 실시 양태에 따른 문서 시각 구조 분석 시스템(200)의 다른 블록도가 도시되어 있다. 문서 시각 구조 분석 시스템(200)은 시각 구조 입력(204)을 수신하고 최적의 분석 트리(206)를 제공하는 문서 시각 구조 분석 컴포넌트(202)를 포함한다. 문서 시각 구조 분석 컴포넌트(202)는 문서 시각 구조 레이아웃의 차별 문법 모델을 이용한다. 문서 시각 구조 분석 컴포넌트(202)는 수신 컴포넌트(208) 및 문법 컴포넌트(210)를 포함한다. 수신 컴포넌트(208)는 시각 구조 입력(204)을 수신하고 이 입력(204)을 문법 컴포넌트(210)로 중계한다. 다른 예에서, 수신 컴포넌트(208)의 기능은 문법 컴포넌트(210)에 포함되어, 문법 컴포넌트(210)가 시각 구조 입력(204)을 직접 수신하게 할 수 있다. 문법 컴포넌트(210)는 또한 기본 구조 레이아웃 문법(212)을 수신한다. 기본 구조 레이아웃 문법(212)은 문서 레이아웃에 대한 초기 시각 구조 문법 프레임워크를 제공한다. 문법 컴포넌트(210)는 시각 구조 입력(204)을 분석하여 최적의 분석 트리(206)를 얻는다. 문법 컴포넌트(210)는 문서의 시각 구조를 분석하는 문법 분석 프로세스의 이용을 통해 이를 달성한다. 문법 컴포넌트(210)는 동적 프로그래밍 프로세스를 이용하여 전역적으로 최적인 분석 트리를 결정한다. 이것은 최적의 분석 트리(206)가 국지적으로만 평가되는 것을 방지함으로써 개선된 전역적 결과를 산출한다.

[0024] 도 3에는 일 실시 양태에 따른 문서 시각 구조 분석 시스템(300)의 또 다른 블록도가 도시되어 있다. 문서 시각 구조 분석 시스템(300)은 시각 구조 입력(304)을 수신하고 최적의 분석 트리(306)를 제공하는 문서 시각 구조 분석 컴포넌트(302)를 포함한다. 문서 시각 구조 분석 컴포넌트(302)는 분석을 위해 문서 시각 구조 레이아웃의 차별 문법 모델을 이용한다. 문서 시각 구조 분석 컴포넌트(302)는 수신 컴포넌트(308) 및 문법 컴포넌트(310)를 포함한다. 문법 컴포넌트(310)는 분석 컴포넌트(312) 및 문서 구조 추출 컴포넌트(314)를 포함한다. 분석 컴포넌트(312)는 문법 비용 함수(318)를 갖는 시각 구조 문법 모델(316)을 포함한다. 시각 구조 입력(304)은 예를 들어 문서 페이지의 시각 레이아웃을 포함한다. 수신 컴포넌트(308)는 시각 구조 입력(304)을 수신하여, 이 입력(304)을 분석 컴포넌트(312)로 중계한다. 다른 예에서, 수신 컴포넌트(308)의 기능은 분석 컴포넌트(312)에 포함되어, 분석 컴포넌트(312)가 시각 구조 입력(304)을 직접 수신하게 할 수 있다. 분석 컴포넌트(312)는 초기에 시각 구조 레이아웃 문법(320)에 기초하여 시각 구조 입력(304)으로부터 문서 시각 구조를 분석한다. 분석 컴포넌트(312)는 문서 구조 추출 컴포넌트(314)와 상호작용하여, 시각 구조 입력(304)으로부터 시각 구조 정보를 추출하는 것을 특별히 돕는다.

[0025] 문서 구조 추출 컴포넌트(314)는 복합 국지 및/또는 전역 피처들을 이용하여, 분석 컴포넌트(312)가 시각 구조 입력(304)을 분석하는 것을 돕는다. 이 컴포넌트(314)는 기계 학습(322), 분석 부스팅(324), 패스트 피처

(326), 이미지 스코어링(328) 및/또는 기타(330) 등등을 포함하지만 이에 제한되지 않는 다양한 옵션 메카니즘을 이용하여 분석 컴포넌트(312)에 의한 시각 구조 레이아웃 분석을 보강한다. 기타(330)는 분석 컴포넌트(312)의 촉진 및/또는 강화를 돕는 추가적인 효율 및/또는 시각 지향 메카니즘을 나타낸다.

[0026] 예를 들어, 기계 학습(322)은 차트를 생성하도록 분석 컴포넌트(312)를 돕기 위해 문서 구조 추출 컴포넌트(314)에 의해 제공될 수 있다. 이어서, 분석 컴포넌트(312)는 차트를 분류 프로세스로 중계되는 라벨링된 예들의 후속 세트로 변환한다. 분류 프로세스는 라벨링된 예들의 후속 세트를 기계 학습과 함께 이용하여 분류자들의 세트를 훈련시킨다. 이어서, 분류 프로세스는 포지티브 예와 네가티브 예 사이의 식별 특성들을 결정한다. 식별 특성들은 분류자들이 정확한 분석 및/또는 부정확한 분석에 대한 적절한 비용의 할당을 돕는 것을 가능하게 한다. 이어서, 분석 컴포넌트(312)는 시각 구조 문법 모델(316)의 문법 비용 함수(318)에서 분류자들의 세트를 이용하여, 라벨링된 예들의 후속 세트에 대한 세부 분석(subparse)들의 스코어링을 돕는다. 이러한 방식으로, 프로세스는 최적의 분석 트리(306)가 얻어질 때까지(예를 들어, 더 높은 스코어링 분석 트리가 얻어지지 않거나, 더 낮은 비용의 분석 트리가 얻어지지 않을 때까지) 반복적으로 계속된다.

[0027] 마찬가지로, 분석 부스팅 메카니즘(324)이 정확한 분석들을 더 효율적으로 학습하는 것을 돕기 위해 분석 컴포넌트(312)에 제공될 수 있다. 문서 피처들의 적분 이미지들의 계산 및/또는 적분 이미지들의 배열들의 이용을 통해 분석 이미지들을 계산하여 분석 효율을 향상시키기 위해 패스트 피처 메카니즘(326)이 제공될 수 있다. 이미지 스코어링 메카니즘(328)은 문법 비용 함수(318)에 대해 분석된 이미지들의 스코어들을 제공함으로써 분석을 도울 수 있다. 이들 메카니즘(322-330)은 옵션이며, 시각 구조 입력(304)의 분석에 필수적이지 않다.

[0028] 문서의 전체 페이지에 대해 단일의 적분 이미지가 아니라 적분 이미지들의 배열들을 이용할 때, 페이지의 각 요소(예를 들어, 문자, 단어, 및/또는 적절한 경우에 라인 등)에 대해 적분 이미지가 계산된다. 피처 계산에 중요한 문자들만을 포함시킴으로써 주의가 집중될 수 있다. 본 시스템 및 방법은 또한 문서 피처들의 계산된 적분 이미지들도 이용할 수 있다. 예를 들어, 큰 화이트 스페이스 직사각형들, 테두리 박스들의 수직 정렬들, 및/또는 텍스트 라인들의 수평 정렬들과 같은 문서 피처들이 사용될 수 있다.

[0029] 따라서, 적분 이미지를 이용함으로써, 이미지 직사각형 내의 화이트 및/또는 블랙 픽셀들의 수를 빠르게 계산할 수 있다. 하나의 이미지에 대한 적분 이미지의 계산은 비용이 많이 들지만, 일단 계산되면, 직사각형 합계들이 빠르게 계산될 수 있다. 하나의 이미지 내에 있거나 있지 않을 수 있는 개체들의 세트가 주어질 때, 이미지로부터 렌더링될 수 있는 기하급수적인 수의 이미지들(거듭제곱 세트 P(N))이 존재하게 된다. 이러한 이미지들의 렌더링 및 렌더링된 각각의 이미지에 대한 직사각형 합계들의 계산은 엄청난 비용이 든다. 따라서, 그 대신에, 개체들 각각에 대해 적분 이미지가 렌더링되며, "적분 이미지 배열"로서 나타내어 진다. 따라서, 이미지들의 임의의 서브세트에 대한 직사각형 합계는 배열들로부터의 직사각형 합계들의 합계이다.

[0030] 이차원 분석

[0031] 다수의 경쟁적인 분석 알고리즘이 존재하지만, 간단하지만 일반적인 하나의 프레임워크는 "차트 분석"이다(M. Kay, "Algorithm schemata and data structures in syntactic processing," pp. 35-70, 1986 참조). 차트 분석은 차트 C(A,R)의 엔트리들을 채우려고 시도한다. 각각의 엔트리는 비말단(non-terminal) A의 최상의 스코어를 말단들의 서브 시퀀스(R)의 해석으로서 저장한다. 임의의 비말단의 비용은 아래의 점화식으로 표현될 수 있다.

수학식 1

[0032]
$$C(\mathrm{A},R_0)=\min_{\substack{\mathrm{A}\rightarrow\mathrm{BC}\\ R_1\cap R_2=\varnothing\\ R_1\cup R_2=R_0}} C(\mathrm{B},R_1)+C(\mathrm{C},R_2)+l(\mathrm{A}\rightarrow\mathrm{BC})$$

[0033] 여기서, {BC}의 범위는 A 내의 모든 생성물(production)을 포괄하며, R_0는 말단들의 서브 시퀀스("영역"으로 표시)이고, R_1 및 R_2는 서로 소이고 그의 합집합이 R_0인 서브 시퀀스들이다(즉, 이들은 "파티션"을 형성한다). 본질적으로, 이 점화식은 말단들의 2개의 서로 소인 세트로의 저비용 분해를 발견함으로써 A에 대한 스코어가 계산됨을 말한다. 각각의 생성에는 테이블에서 비용(또는 손실 또는 네가티브 로그 확률) l(A->BC)이 할당된다. 차트 내의 엔트리들(때로는 에지라고 함)은 임의의 순서로, 하향식 또는 상향식으로 채워질 수 있다. 분석 프로세스의 복잡성은 채워져야 하는 차트 엔트리들의 수 및 각각의 엔트리를 채우는 데 필요한 작업으로부터 발생한다. P개의 비말단을 포함하는 문법을 이용하여 N개의 말단의 선형 시퀀스를 분석하는 동안 구성된 차트는

$O(PN^2)$개의 엔트리를 갖는다($\frac{1}{2}\binom{N}{2} \in O(N^2)$ 개의 연속 서브 시퀀스들 {i,j}이 존재하는데, $0 \le i \le j$이고, $j<N$이다). 각각의 엔트리를 채우는 데 필요한 작업은 O(N)이므로, 전체 복잡성은 $O(PN^3)$이다.

[0034] 불행하게도, 말단들의 2차원 배열에 대한 차트 분석의 직접 적용은 기하급수적인 시간을 필요로 한다. 중요한 문제는 말단들이 더 이상 선형적인 순차적 순서를 갖지 않는다는 점이다. 수학식 1로 돌아가면, 이제 영역 R_0는 서브세트이고, R_1 및 R_2는 서로 소이고 그의 합집합이 R_0인 서브세트들이다(즉, 이들은 파티션을 형성한다). 차트의 크기가 분석될 수 있는데, 이는 O(P|P(N)|)이며, 여기서 P(N)은 N개 말단의 모든 서브세트의 세트이다. 기하급수적인 수의 서브세트들이 존재하므로, 알고리즘은 지수적이다.

[0035] 헐은 비용의 기하학적 성분이 너무 큰 경우에 검색을 간결하게 하는 기하학적 기준을 도입하였다(J.F.Hull, "Recognition of mathematics using a two-dimensional trainable context-free grammar," Maseter's thesis, MIT, June 1996 참조). 밀러 및 비올라는 $chull(R_1) \cap R_2 = \Phi$ 또는 $chull(R_2) \cap R_1 = \Phi$를 위반하는 영역들($R_1$, R_2)을 거절하는 볼록 다각형(convex hull)들에 기초하는 발견적 방법을 도입하였다(E.G.Miller and P.A.Viola, "Ambiguity and constraint in mathematical expression recognition," in Proceedings of the National Conference of Artificial Intelligence, American Association of Artificial Intelligence, 1998 참조). 이제, 각각의 세트가 페이지의 볼록 영역 내에 있으므로, 이들 세트를 영역이라 지칭하는 것이 적절하다. 말단들이 라인을 따라 위치하는 경우(따라서, 엄격한 선형적인 순서를 갖는 경우), 볼록 다각형 기준은 $O(N^2)$개의 영역을 산출하고, 통상의 분석에서 사용되는 선형 시퀀스와 동등하다는 점에 유의할 가치가 있다.

[0036] 볼록 다각형 제한은 물론, 다른 기하학적 제한을 이용함으로써, 분석 동안 고려되는 서브세트들의 세트가 크게 감소될 수 있다. 이들 제한은 대다수 유형의 인쇄 문서들에 대해 거의 $O(N^3)$의 복잡성을 산출하도록 결합된다.

[0037] 문서 레이아웃 분석

[0038] 문서 레이아웃 분석의 하나의 목적은 스캐닝된 문서를, 예를 들어 LaTeX 및/또는 워드 프로세서 등과 같은 문서 준비 프로그램용의 완전 편집 가능한 입력 파일로 변환하는 데 필요한 정보를 결정하는 것이다. 스캐닝된 파일 내의 텍스트는 OCR을 이용하여 쉽게 추출될 수 있지만, 이 정보는 쉽게 편집할 수 있는 파일을 생성하기에 충분하지 않다. 단락 경계, 칼럼, 조정, 및 보다 중요하게는 판독 흐름과 같은 추가 정보가 또한 필요하다. 이러한 문서 구조 정보는 또한 종종 휴대형 문서 파일(PDF) 및 포스트스크립트 파일로부터 누락된다. 스캔, PDF 및/또는 포스트스크립트에 대한 것인지에 관계없이, 문서 구조 정보의 추가는 재 페이지화, 재 포맷 및/또는 편집 등이 될 수 있는 살아 있는 문서를 생성한다. 따라서, 이러한 능력을 갖는 것은 문서의 유용성을 크게 향상시킨다.

[0039] 문서 준비 프로그램들은 종종 인쇄 페이지를 섹션들로 분할한다. 각각의 섹션은 소정 수의 칼럼을 가지며, 각각의 칼럼은 소정 수의 단락을 갖는다. 이러한 순환 구조는 아래의 테이블 1에서 문법으로서 표현된다. 이러한 구조에 대한 지식은 스캐닝된 문서로부터 편집 가능한 파일을 정확히 생성하기에 충분하다.

[0040] 테이블 1: 인쇄 페이지들을 기술하는 데 사용될 수 있는 문법 예

```
(Page → ParList)
(ParList → Par ParList)
(ParList → Par)
(Par → LineList)
(LineList → Line LineList)
(LineList → Line)
(Line → WordList)
(WordList → Word WordList)
(WordList → Word) (Word → terminal)
```

[0041]

[0042] UWIII 문서 이미지 데이터베이스를 이용하여 실험을 수행하였다(I.Philips, S.Chen, and R.Haralick, "Cd-rom document database standard," in Proceedings of 2nd International Conference on Document Analysis and Recognition, 1993 참조). 이 데이터베이스는 라인, 단락, 영역 및 판독 순서에 대한 그라운드 트루스(ground

truth)를 갖는 스캐닝된 문서를 포함한다. 도 4에는 UWIII 데이터베이스로부터의 예시적인 페이지(400)가 도시되어 있다. 분석 알고리즘에 대한 입력은 라인들의 테두리 박스들(예를 들어, 테두리 단락 박스(402) 및 테두리 라인 박스(404))이다. 출력은 섹션/칼럼/단락으로의 계층적 분해이다. 대다수의 문서에 대해, 그라운드 트루스 라벨들은 위의 문법으로 쉽게 변환된다. 훈련 및 평가는 연구 논문, 서적 및 잡지로부터의 페이지들을 포함하는 60개 문서의 세트를 이용하여 수행되었다.

[0043] 인쇄된 수학 해석

[0044] 학술 연구 단체에서, 거의 모든 새로운 논문은 PDF 또는 포스트스크립트 포맷으로 이용 가능하게 되어 있다. 이들 포맷은 인쇄에는 편리하지만, 쉬운 재사용 또는 재 포맷팅을 지원하지 않는다. 하나의 분명한 예는 쉽게 추출, 편집 또는 검색될 수 없는 포함된 방정식들이다. 다른 예는 테이블, 각주, 및 서지 사항 등을 포함한다. 과학적 공개를 위한 사실상의 표준은 LaTeX인데, 이는 부분적으로는 LaTeX가 강력하고 고품질의 수학 레이아웃을 제공하기 때문이다. PDF 또는 포스트스크립트 문서 어느 것도 원본을 생성하는 데 사용된 LaTeX 방정식들을 재구성하는 데 필요한 정보를 제공하지 못한다.

[0045] 훈련 LaTeX 문서들의 세트가 주어지면, LaTeX 매크로들의 세트가 문서 렌더링 프로세스를 "도구화(instrument)"하는 데 사용될 수 있다. 그 결과는 페이지 상의 문자들의 테두리 박스들 및 대응 LaTeX 표현식을 추출하기 위해 처리될 수 있는 도구화된 장치 독립(DVI) 파일들의 세트이다. 이들 매크로는 ArXiv 사전 인쇄 서버로부터 이용될 수 있는 LaTeX 파일들의 세트에 적용된다(도 5에 도시된, 수학적 표현 인식기를 훈련시키는 데 사용되는 예시적인 방정식(500) 참조).

[0046] 후처리 후에, 훈련 데이터는 각각이 규칙에 맞게 구성된 말단들의 구문 트리인 표현식들의 집합이다. 이들 트리는 문법을 직접 유도할 수 있는 기회를 제공하는데, 이는 문법의 생성이 입력 트리들로부터 직접 관측될 수 있기 때문이다(이러한 문법은 종종 "트리-뱅크" 문법이라 한다). 유도된 문법이 아래의 테이블 2에 나타나 있다. 테이블 2에 문법의 말단들은 포함되지 않으며, 말단들은 비말단인 RawItem에 의해 참조된다. RawItem들의 세트는 수학적 표현을 구성하는 데 사용되는 문자, 숫자, 및 심벌이다. 문법의 말단들은 검은색(black ink)의 프리미티브(primitive)가 연결된 형태의 요소들이다.

[0047] 테이블 2: 수학적 표현들에 대한 문법

(Expr → Row)
(Row → Row Item)
(Row → Item)
(Item → SubItem)
(Item → FracItem)
(Item → RawItem)
(Item → SupItem)
(FracItem → Row FracItem1)
(FracItem1 → BAR Row)
(SubItem → SupItem Row)
(SubItem → RawItem Row)
(SupItem → RawItem Row)

[0048]

[0049] 수학적 분석에 관한 다른 작업과 달리, 말단들은 해석이 시작되기 전에 분할되고 인식된 것으로 가정하지 않는다. 말단들의 인식은 분석 프로세스의 필수 과정이다. 모든 심벌 유형은 말단들의 생성을 기술하는 관련 문법 규칙을 갖는다. 예를 들어, (RawItem->EQUALS) 및 (EQUALS->CC1 CC2)는 "등가 표시(equals sign)"가 한 쌍의 연결된 성분들로 이루어진다는 것을 나타낸다. EQUALS의 생성과 연관된 비용 함수는 "="로 보이는 한 쌍의 연결 성분들에 낮은 비용을 할당하도록 학습되어야 한다. 이러한 문제의 전반적인 해결은 기계적으로 간단하다. 문법은 예시적인 LaTeX 파일들로부터 생성되며, 피처들은 아래에 정의되는 일반적으로 중요하다고 여겨지는 많은 세트의 피처들로부터 자동으로 선택된다.

[0050] 피처들

[0051] 생성 스코어링 함수들을 학습하는 데 사용되는 몇몇 피처들은 광범위한 태스크들에 대해 일반적으로 적용 가능하며 유용하다. 기하학적 테두리 박스 피처(geometric bounding box features)들의 세트는 성분(component)들의 정렬(alignment)을 측정하는 데 유용한 것으로 입증되었다. 제1 유형의 피처 세트는 테두리 박스들 R_0, R_1, 및 R_2의 세트들과 관련된다. 이들은 페이지 좌표에 있어서의 박스의 코너들의 위치(X_i,Y_i) 및 박스의 크기(W, H)를 측정한다. {m_j(R)}로서 참조되는 총 360개의 측정 피처들이 존재한다. 제2 유형의 피처 세트는 조합된 것으로, 박스 측정 피처들의 모든 쌍 g(m_j(R_a),m_j(R_b))와 관련되는데, 여기서 a 및 b는 {0,1,2}이며, 함수 g는 가산, 감산, 승산, 제산, 최소 또는 최대일 수 있다. 제3 유형의 피처 세트는 영역들에 포함된 말단들의 테두리 박스들의 특성들을 측정한다. 이것은 모든 영역 말단에 대해 평가된 소정의 측정 피처의 최소, 최대, 평균, 표준 편차 및 중앙 값의 측정을 포함한다.

[0052] 또한, 시각적 외관(visual apperance)에 기초하여 영역들을 구별하도록 설계된 패턴 인식 피처들에 대한 많은 세트가 존재한다. 이러한 피처들은 영역들 내의 말단들의 렌더링된 이미지들 상에 작용한다. 도 6에는 수학적 표현(602)의 예(600)가 도시되어 있다. 파싱(parsing)이 진행되는 동안, 표현 Z_0(604)이 접해지게 되며 이를 해석해야 한다. 생성 스코어링 프로세스에 대한 입력으로서 사용되는 4개의 렌더링된 이미지(606)가 예시되어 있다. 말단들 자체가 외관에 기초하여 인식되어야 하는 경우, 시각 피처들이 필요하다. 비올라 및 존스에 의해 제안된 직사각형 피처들이 채택된다(P.Viola and M.Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001 참조). 이들은 계산적으로 효율적이며, 광범위한 태스크들에 대해 유효한 것으로 입증되었다. 각각의 입력 이미지는 위치 및 스케일에서 균일하게 샘플링된 121개의 단일 직사각형 피처에 의해 표현된다. 보다 어려운 이미지 인식 태스크들에 대해서는 훨씬 더 많은 세트가 사용되었지만, 121개의 단일 직사각형 피처에 의한 표현은 앞선 예와 같은 태스크에 대해서는 충분한 것으로 입증되었다.

[0053] 기하학적 정규화는 이미지 분류 함수들을 구성할 때 중요한 문제이다. 이 경우, R_0의 크기 및 위치를 정규화하는 기준 프레임이 선택된다. 목표는 R_0에 대해 시각 이미지의 80%를 채우는 것이다. R_1 및 R_2의 말단들은 이러한 좌표 프레임에서 렌더링된다. 이것은 R_1 및 R_2의 상대 위치들에 대한 정보를 포함하는 입력 이미지를 갖는 이미지 피처들을 제공한다. 따라서, 예를 들어, R_2가 서브스크립트인 경우, 그의 렌더링된 성분들의 위치는 기준 프레임의 하부를 향할 것이다. 마지막으로, 전체 문서로부터의 말단들은 R_0의 기준 프레임에서 렌더링되지만, 훨씬 더 작은 스케일로 렌더링된다. 이 이미지는 문서 "문맥"을 인코딩하며, 소정 유형의 국지적 명확화를 수행하는 데 사용될 수 있다.

[0054] 분석 동안, 모든 잠재적 영역 및 서브 영역은 한 세트의 이미지들로서 인코딩된다. 많은 영역이 존재할 때, 이미지 재 스케일링을 수반하는 이미지 인코딩 프로세스는 지나치게 많은 양의 계산을 초래할 것이다. 이러한 계산을 피하기 위해, 비올라 및 존스에 의해 도입된 적분 이미지 표현을 이용하여, 추가 비용 없이 임의의 스케일로 직사각형 필터들을 계산한다.

[0055] 예들

[0056] 전술한 피처들을 이용하여 두 세트의 실험들이 수행되었다. 문법 파라미터들을 학습하기 위한 전체 프로세스는 아래의 테이블 3에 기술되어 있다. 각각의 학습 라운드에서, 결정 스텀프들(decision stumps) 상의 AdaBoost가 사용된다. 이것은 복잡성 제어(초기 중지)를 위한 매우 간단한 메카니즘을 제공한다. 이것은 또한 피처 선택을 위한 메카니즘을 제공하는데, 이는 각각의 부스팅 라운드가 단일 피처와 또한 연관되는 단일 스텀프를 선택하기 때문이다.

[0057] 테이블 3: 훈련 알고리즘을 위한 의사 코드

```
0) Initialize weights to zero for all productions
1) Parse a set of training examples using current parameters
2) For each production in the grammar
2a)    Collect all examples from all charts.
          Examples from the true parse are TRUE.
          All others are FALSE.
2b)    Train a classifier on these examples.
2c)    Update production weights.
          New weights are the cumulative sum.
3) Repeat Step 1.
```

[0058]

[0059] 훈련의 초기 라운드들은 최종 분배를 나타내지 않는 예들을 만날 가능성이 높으므로, AdaBoost는 복잡성을 증가시키는 스케쥴로 실행된다. 제1 부스팅 라운드는 2개의 약한 분류자를 선택한다. 제2 및 제3 라운드는 4개 및 8개의 분류자를 각각 선택한다. 이후, 각각의 분석 라운드에서 8개의 분류자(따라서 8개의 피처)가 선택된다.

[0060] 분석 결과들의 평가는 다소 기교적인 면이 있다. 어떠한 시스템도 완벽하지 않으므로, 대개는 정확한 분석의 품질을 정량화하는 기준을 정하는 것이 중요하다. 하나의 계획은 각 유형의 생성에 대한 리콜 및 정확도를 측정하는 것이다. 그라운드 트루스는 각 생성의 많은 예를 포함한다. 각 생성이 정확하게 식별되는 횟수의 퍼센트가 리콜이다. 학습된 문법은 각각의 입력 예에 대한 분석을 산출한다. 이들 생성이 정확한 분석에 대응하는 횟수의 퍼센트가 정확도이다.

[0061] UWIII 문서 데이터베이스는 3개의 상호 검증 라운드 내에 80-20 분할된 57개의 파일을 포함한다(테이블 4 참조-평균은 모든 생성에 대한 평균 성능을 나타낸다. 가중 평균은 만난 예들의 수에 기초하여 평균에서의 가중치를 할당한다.). 훈련 세트에 대한 성능은 거의 완벽한 반면, 테스트 세트에 대한 성능은 양호하지만, 완벽과는 거리가 멀다. 보다 큰 훈련 세트 및/또는 피처 표현의 변화가 보편화를 개선할 수 있다. 문서 및 수학식 도메인 양자에 대해, 80개의 말단을 가진 일반적인 입력을 1GB의 RAM을 갖춘 1.7GHz 펜티엄 4에서 분석하는 데 약 30초가 걸린다.

[0062] 테이블 4: UWIII 문서 구조 추출 태스크에 대한 결과

[0063]

	F1	정확도	리콜
훈련:			
평균	0.96	0.97	0.96
가중	0.95	0.95	0.95
테스트:			
평균	0.85	0.86	0.84
가중	0.89	0.89	0.88

[0064] 방정식 데이터베이스는 180개의 표현, 및 λ 및 δ와 같은 51개의 상이한 수학 심벌을 갖춘 문법을 포함한다. 결과들이 아래 테이블 5에 나타나 있다.

[0065] 테이블 5: 수학적 표현 인식 태스크에 대한 결과

[0066]

	F1	정확도	리콜
훈련:			
가중	1	1	1
테스트:			
가중	0.942	0.947	0.936

[0067] 시스템 및 방법의 예들은 인쇄된 문서들의 성분들을 동시에 분할하고 인식하도록 학습할 수 있는 분석 프레임워크를 제공한다. 이 프레임워크는 분석 프로세스의 모든 파라미터가 훈련 예들의 데이터베이스를 이용하여 설정

된다는 점에서 아주 일반적이다. 이 프레임워크의 유효성 및 보편성은 2개의 응용, 즉 페이지 레이아웃 구조 추출 및 수학적 표현 인식을 제시함으로써 입증되었다. 첫 번째의 경우, 알고리즘에 대한 입력은 페이지 상의 라인들의 집합이며, 출력은 섹션, 칼럼 및 단락 구조이다. 두 번째의 경우, 입력은 페이지 상의 연결 성분들의 집합이며, 출력은 한 세트의 인식된 수학적 심벌들 및 입력을 재생하는 데 필요한 LaTeX 코드이다. 최종 시스템들은 아주 다르지만, 정확한 인식 시스템을 생성하기 위해 학습 및 분석 프로세스에 대한 매우 적은 수정이 필요하다.

[0068] 위에 지시되고 설명된 예시적인 시스템들에 비추어, 실시예들에 따라 구현될 수 있는 방법들이 도 7 및 8의 흐름도를 참조하여 보다 잘 이해될 것이다. 설명의 간략화를 위해 방법들은 일련의 블록으로서 지시되고 설명되지만, 몇몇 블록들은 일 실시예에 따라 상이한 순서로, 그리고/또는 여기에 지시되고 설명되는 것과 다른 블록들과 동시에 발생할 수 있으므로, 실시예들은 블록들의 순서에 의해 한정되지 않는다는 것을 이해하고 인식해야 한다. 더욱이, 실시예들에 따르면 모든 예시된 블록이 방법들을 구현하는 데 필요한 것은 아닐 수도 있다.

[0069] 실시예들은 하나 이상의 컴포넌트에 의해 실행되는 프로그램 모듈과 같은 컴퓨터 실행가능 명령들과 일반적으로 관련하여 설명될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 개체, 데이터 구조 등을 포함한다. 일반적으로, 프로그램 모듈의 기능은 다양한 실시예에서 필요에 따라 조합 또는 분산될 수 있다.

[0070] 도 7에는 일 실시 양태에 따른 문서 시각 구조 분석을 돕는 방법(700)의 흐름도가 도시되어 있다. 방법(700)은 702에서 시작하여, 문서의 시각 구조와 연관된 입력을 수신한다(704). 이어서, 문법 분석 프로세스가 문서 시각 구조의 추론에 적용되며(706), 흐름은 708에서 종료된다. 문법 분석 프로세스는 기계 학습 등을 이용하여 문법 비용 함수를 돕는 분류자를 구성하는 프로세스를 포함할 수 있지만, 이에 제한되는 것은 아니다. 기계 학습은 예를 들어 퍼셉트론 기반 기술 등과 같은 통상의 기계 학습 기술을 포함하지만 이에 제한되는 것은 아니다.

[0071] 도 8을 참조하면, 일 실시 양태에 따른 문서 시각 구조 분석을 돕는 방법(800)의 다른 흐름도가 도시되어 있다. 방법(800)은 802에서 시작하여, 문서의 시각 구조와 연관된 입력을 수신한다(804). 이어서, 문서의 시각 구조가 복합 국지 및/또는 전역 피처들을 이용하여 입력으로부터 추출되며(806), 흐름은 808에서 종료된다. 기계 학습, 분석 부스팅, 패스트 피처, 및/또는 이미지 스코어링 등을 포함하지만 이에 제한되지 않는 다양한 옵션 메카니즘이 시각 구조 추출을 개선하는 데 이용될 수 있다. 예를 들어, 기계 학습은 차트를 생성하기 위해 분석을 도울 수 있다. 이어서, 차트는 분류 프로세스로 중계되는 라벨링된 예들의 후속 세트로 변환될 수 있다. 분류 프로세스는 라벨링된 예들의 후속 세트를 기계 학습과 함께 이용하여 분류자들의 세트를 훈련시킬 수 있다. 이어서, 분류 프로세스는 포지티브 예 및 네가티브 예 사이의 식별 특성들을 결정할 수 있다. 식별 특성들은 분류자들이 정확한 분석 및/또는 부정확한 분석에 대한 적절한 비용의 할당을 돕는 것을 가능하게 한다.

[0072] 유사하게, 정확한 분석들을 보다 효율적으로 학습하는 것을 돕기 위해 분석 부스팅이 분석 프로세스에 제공될 수 있다. 분석 효율을 개선하도록 문서 피처들의 적분 이미지들의 계산 및/또는 적분 이미지들의 배열들의 이용을 통해 분석 이미지들을 계산하기 위해 패스트 피처 프로세스가 제공될 수 있다. 이미지 스코어링 프로세스가 분석에 이용되는 비용 함수에 대해 분석된 이미지들의 스코어를 제공함으로써 분석을 도울 수 있다.

[0073] 다양한 실시 양태를 구현하기 위한 추가적인 배경을 제공하기 위해, 도 9 및 아래의 설명은 다양한 실시 양태가 구현될 수 있는 적절한 컴퓨팅 환경(900)에 대한 간단하고 일반적인 설명을 제공하는 것을 의도한다. 실시예들은 로컬 컴퓨터 및/또는 원격 컴퓨터 상에 실행되는 컴퓨터 프로그램의 컴퓨터 실행가능 명령들과 일반적으로 관련하여 전술되었지만, 다른 프로그램 모듈들과 조합하여 구현될 수도 있다는 것을 이 분야의 전문가들은 인식할 것이다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하고, 그리고/또는 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조 등을 포함한다. 더욱이, 이 분야의 전문가들은, 본 발명의 방법들이 각기 하나 이상의 관련 장치와 유효하게 통신할 수 있는 퍼스널 컴퓨터, 핸드-헬드 컴퓨팅 장치, 마이크로프로세서 기반 및/또는 프로그램가능한 가전제품 등은 물론, 단일 프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니 컴퓨터, 메인프레임 컴퓨터를 포함하는 다른 컴퓨터 시스템 구성을 이용하여 실시될 수 있음을 이해할 것이다. 예시된 실시 양태들은 또한 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 소정의 태스크들이 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 그러나 실시 양태들 모두는 아니더라도, 그 일부 양태는 독립형 컴퓨터 상에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈들은 로컬 및/또는 원격 메모리 저장 장치 둘다에 위치할 수 있다.

[0074] 본 출원에서 사용되는 "컴포넌트"라는 용어는 컴퓨터 관련 엔티티, 즉 하드웨어, 하드웨어와 소프트웨어의

조합, 소프트웨어 또는 실행 중인 소프트웨어를 지칭하는 것을 의도한다. 예를 들어, 컴포넌트는 프로세서 상에서 실행되는 프로세스, 프로세서, 개체, 실행 파일, 실행 스레드, 프로그램 및 컴퓨터일 수 있지만, 이에 제한되는 것은 아니다. 예를 들어, 서버 상에서 실행되는 애플리케이션 및/또는 서버는 컴포넌트일 수 있다. 또한, 컴포넌트는 하나 이상의 서브 컴포넌트를 포함할 수 있다.

[0075] 도 9를 참조하면, 실시예들의 다양한 양태를 구현하기 위한 예시적인 시스템 환경(900)은 처리 장치(904), 시스템 메모리(906), 및 시스템 메모리를 비롯한 각종 시스템 컴포넌트를 처리 장치(904)에 결합시키는 시스템 버스(908)를 포함하는 통상의 컴퓨터(902)를 포함한다. 처리 장치(904)는 상업적으로 입수 가능하거나 독점적인 임의의 프로세서일 수 있다. 또한, 처리 장치는 예를 들어 병렬로 접속될 수 있는 둘 이상의 프로세서로 구성되는 멀티 프로세서로서 구현될 수 있다.

[0076] 시스템 버스(908)는 메모리 버스 또는 메모리 컨트롤러, 주변 버스, 및 예를 들어 PCI, VESA, 마이크로채널, ISA 및 EISA와 같은 각종 통상의 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스를 비롯한 몇몇 유형의 버스 구조 중 어느 것이라도 될 수 있다. 시스템 메모리(906)는 ROM(910) 및 RAM(912)을 포함한다. 예를 들어 시동 중과 같은 때에 컴퓨터(902) 내의 구성요소들 사이의 정보 전송을 돕는 기본 루틴들을 포함하는 기본 입/출력 시스템(BIOS)(914)은 ROM(910)에 저장된다.

[0077] 컴퓨터(902)는 또한 예를 들어 하드 디스크 드라이브(916), 예를 들어 이동식 디스크(920)에 기록을 하거나 그로부터 판독을 하는 자기 디스크 드라이브(918), 및 예를 들어 CD-ROM 디스크(924) 또는 다른 광학 매체에 기록을 하거나 그로부터 판독을 하는 광 디스크 드라이브(922)를 포함할 수 있다. 하드 디스크 드라이브(916), 자기 디스크 드라이브(918) 및 광 디스크 드라이브(922)는 하드 디스크 드라이브 인터페이스(926), 자기 디스크 드라이브 인터페이스(928) 및 광 드라이브 인터페이스(930)에 의해 각각 시스템 버스(908)에 접속된다. 드라이브들(916-922) 및 이들과 관련된 컴퓨터 판독가능 매체는 컴퓨터(902)에 대한 데이터, 데이터 구조, 컴퓨터 실행가능 명령 등의 비휘발성 저장을 제공한다. 위의 컴퓨터 판독가능 매체의 설명은 하드 디스크, 이동식 자기 디스크 및 CD와 연관되지만, 자기 카세트, 플래시 메모리 카드, 디지털 비디오 디스크, 베르누이 카트리지 등과 같이 컴퓨터에 의해 판독될 수 있는 다른 유형의 매체도 예시적인 운영 환경(900)에서 사용될 수 있으며, 또한 이러한 임의의 매체는 실시예들의 방법들을 수행하기 위한 컴퓨터 실행가능 명령들을 포함할 수 있다는 것을 이 분야의 전문가들은 이해해야 한다.

[0078] 운영 체제(932), 하나 이상의 애플리케이션 프로그램(934), 다른 프로그램 모듈(936) 및 프로그램 데이터(938)를 포함하는 다수의 프로그램 모듈이 드라이브들(916-922)에 저장될 수 있다. 운영 체제(932)는 임의의 적절한 운영 체제 또는 운영 체제들의 조합일 수 있다. 예를 들어, 애플리케이션 프로그램(934) 및 프로그램 모듈(936)은 일 실시 양태에 따른 인식 스킴을 포함할 수 있다.

[0079] 사용자는 키보드(940) 및 포인팅 장치(예를 들어, 마우스 942)와 같은 하나 이상의 사용자 인터페이스를 통해 컴퓨터(902)에 명령 및 정보를 입력할 수 있다. 다른 입력 장치(도시되지 않음)는 마이크, 조이스틱, 게임 패드, 위성 안테나, 무선 리모컨, 스캐너 등을 포함할 수 있다. 이들 및 다른 입력 장치는 종종 시스템 버스(908)에 결합된 직렬 포트 인터페이스(944)를 통해 처리 장치(904)에 접속되지만, 병렬 포트, 게임 포트 또는 USB(universal serial bus) 등의 다른 인터페이스에 의해 접속될 수도 있다. 모니터(946) 또는 다른 유형의 디스플레이 장치도 비디오 어댑터(948) 등의 인터페이스를 통해 시스템 버스(908)에 접속된다. 모니터(946) 외에, 컴퓨터(902)는 스피커 및 프린터 등의 기타 주변 출력 장치(도시되지 않음)를 포함할 수 있다.

[0080] 컴퓨터(902)는 하나 이상의 원격 컴퓨터(960)에 대한 논리적 접속을 사용하여 네트워크화된 환경에서 동작할 수 있다는 것을 이해해야 한다. 원격 컴퓨터(960)는 워크스테이션, 서버 컴퓨터, 라우터, 피어 장치 또는 다른 공통 네트워크 노드일 수 있으며, 도 9에는 간략화를 위해 메모리 저장 장치(962)만이 도시되어 있지만, 일반적으로 컴퓨터(902)와 관련하여 설명된 요소들의 대부분 또는 모두를 포함한다. 도 9에 도시된 논리 접속들은 LAN(964) 및 WAN(966)을 포함할 수 있다. 이러한 네트워킹 환경은 사무실, 전사적 컴퓨터 네트워크(enterprise-wide computer network), 인트라넷 및 인터넷에서 일반적인 것이다.

[0081] 예를 들어, LAN 네트워킹 환경에서 사용될 때, 컴퓨터(902)는 네트워크 인터페이스 또는 어댑터(968)를 통해 LAN((964)에 접속된다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(902)는 통상적으로 모뎀(예를 들어, 전화, DSL, 케이블 등; 970)을 포함하거나, LAN 상의 통신 서버에 접속되거나, 인터넷과 같은 WAN을 통해 통신을 설정하기 위한 다른 수단을 구비한다. 컴퓨터(902)에 관하여 내장형 또는 외장형일 수 있는 모뎀(970)은 직렬 포트 인터페이스(944)를 통해 시스템 버스(908)에 접속된다. 네트워크화된 환경에서, 프로그램 모듈들(애플리케이션 프로그램 포함; 934) 및/또는 프로그램 데이터(938)는 원격 메모리 저장 장치(962)에 저장될 수 있다. 도시된

네트워크 접속들은 예시적이며, 일 실시 양태를 수행할 때, 컴퓨터들(902, 906) 사이에 통신 링크를 설정하는 다른 수단(예를 들어, 유선 또는 무선)이 사용될 수 있다는 것을 이해할 것이다.

[0082] 컴퓨터 프로그래밍 분야의 전문가들의 관례에 따라, 실시예들은 달리 지시되지 않는 한 컴퓨터(902) 또는 원격 컴퓨터(960)와 같은 컴퓨터에 의해 수행되는 행위들 또는 동작들의 심벌 표현과 관련하여 설명되었다. 이러한 행위 또는 동작은 때때로 컴퓨터에 의해 실행되는 것으로서 지칭된다. 행위 또는 심벌로 표현된 동작은 전기 신호 표현의 결과적인 변환 또는 축소를 유발하는 데이터 비트들을 나타내는 전기 신호들의 처리 장치(904)에 의한 조작, 및 메모리 시스템(시스템 메모리(906), 하드 드라이브(916), 플로피 디스크(920), CD-ROM(924) 및 원격 메모리(962)를 포함함) 내의 메모리 위치들에서의 데이터 비트들의 유지를 포함하며, 이에 따라 컴퓨터 시스템의 동작은 물론 신호들의 다른 처리를 재구성하거나, 변경한다. 이러한 데이터 비트들이 유지되는 메모리 위치들은 데이터 비트들에 대응하는 특정 전기, 자기 또는 광학 특성을 갖는 물리적 위치들이다.

[0083] 도 10은 실시예들이 상호작용할 수 있는 샘플 컴퓨팅 환경(1000)의 다른 블록도이다. 시스템(1000)은 하나 이상의 클라이언트(1002)를 포함하는 시스템을 더 도시하고 있다. 클라이언트(1002)는 하드웨어 및/또는 소프트웨어(예를 들어, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 시스템(1000)은 또한 하나 이상의 서버(1004)를 포함한다. 서버(1004)는 또한 하드웨어 및/또는 소프트웨어(예를 들어, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 클라이언트(1002)와 서버(1004) 간의 하나의 가능한 통신은 둘 이상의 컴퓨터 프로세스 사이에 전송하기에 적합한 데이터 패킷의 형태일 수 있다. 시스템(1000)은 클라이언트(1002)와 서버(1004) 사이의 통신을 돕는 데 사용될 수 있는 통신 프레임워크(1008)를 포함한다. 클라이언트(1002)는 클라이언트(1002)에 국한된 정보를 저장하는 데 사용될 수 있는 하나 이상의 클라이언트 데이터 스토어(1010)에 접속된다. 마찬가지로, 서버(1004)는 서버(1004)에 국한된 정보를 저장하는 데 사용될 수 있는 하나 이상의 서버 데이터 스토어(1006)에 접속된다.

[0084] 실시예들의 시스템 및/또는 방법은 인식 지원 컴퓨터 컴포넌트들 및 비 컴퓨터 관련 컴포넌트들에서 똑같이 이용될 수 있다는 것을 이해해야 한다. 또한, 실시예들의 시스템 및/또는 방법은 컴퓨터, 서버 및/또는 핸드헬드 전자 장치 등을 포함하지만 이에 제한되지 않는 거대 배열의 전자 관련 기술들에서 이용될 수 있다는 것을 이 분야의 전문가들은 이해할 것이다.

[0085] 전술한 내용은 실시예들을 포함한다. 물론, 실시예들의 설명을 위해 컴포넌트들 또는 방법들의 모든 구상 가능한 조합을 설명할 수 없지만, 실시예들의 많은 추가 조합 및 교환이 가능하다는 것을 이 분야의 전문가는 이해할 수 있다. 따라서, 본 발명은 첨부된 청구범위의 사상 및 범위 내에 있는 모든 그러한 변경, 수정 및 변형을 포함하는 것을 의도한다. 또한, "구비"라는 용어가 상세한 설명 또는 청구범위에서 사용되는 한도까지, 이 용어는 "포함"이라는 용어가 청구범위에서 전이구로 사용될 때 해석되는 것과 유사한 방식으로 포괄적임을 의도한다.

도면의 간단한 설명

[0008] 도 1은 일 실시 양태에 따른 문서 시각 구조 분석 시스템의 블록도이다.

[0009] 도 2는 일 실시 양태에 따른 문서 시각 구조 분석 시스템의 다른 블록도이다.

[0010] 도 3은 일 실시 양태에 따른 문서 시각 구조 분석 시스템의 또 다른 블록도이다.

[0011] 도 4는 일 실시 양태에 따른 UWIII 데이터베이스의 페이지 예를 나타내는 도면이다.

[0012] 도 5는 일 실시 양태에 따라 수학적 표현 인식기를 훈련시키는 데 사용되는 방정식 예를 나타내는 도면이다.

[0013] 도 6은 일 실시 양태에 따른 수학적 표현을 나타내는 도면이다.

[0014] 도 7은 일 실시 양태에 따른 문서 시각 구조 분석을 돕는 방법의 흐름도이다.

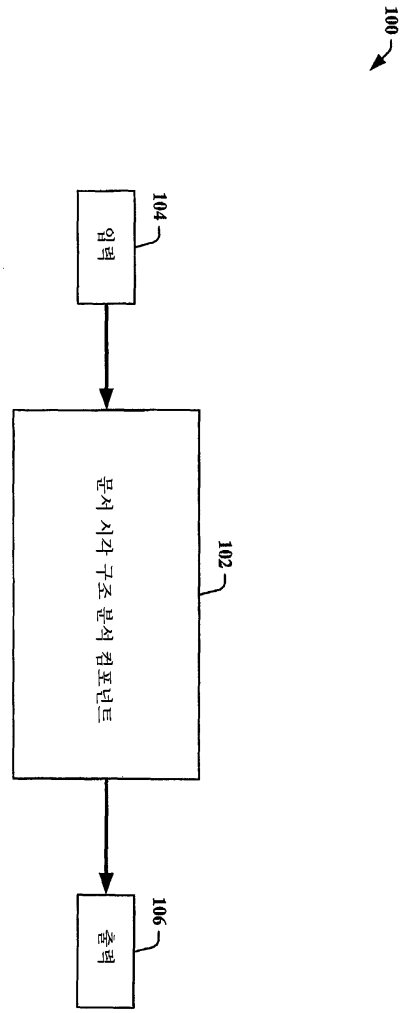
[0015] 도 8은 일 실시 양태에 따른 문서 시각 구조 분석을 돕는 방법의 다른 흐름도이다.

[0016] 도 9는 일 실시예가 동작할 수 있는 운영 환경의 일례를 나타내는 도면이다.

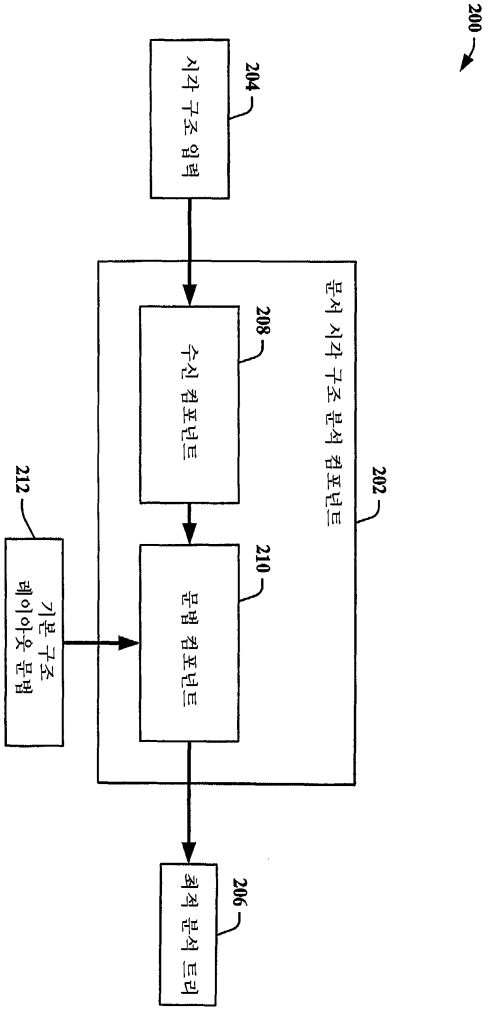
[0017] 도 10은 일 실시예가 동작할 수 있는 운영 환경의 다른 예를 나타내는 도면이다.

도면

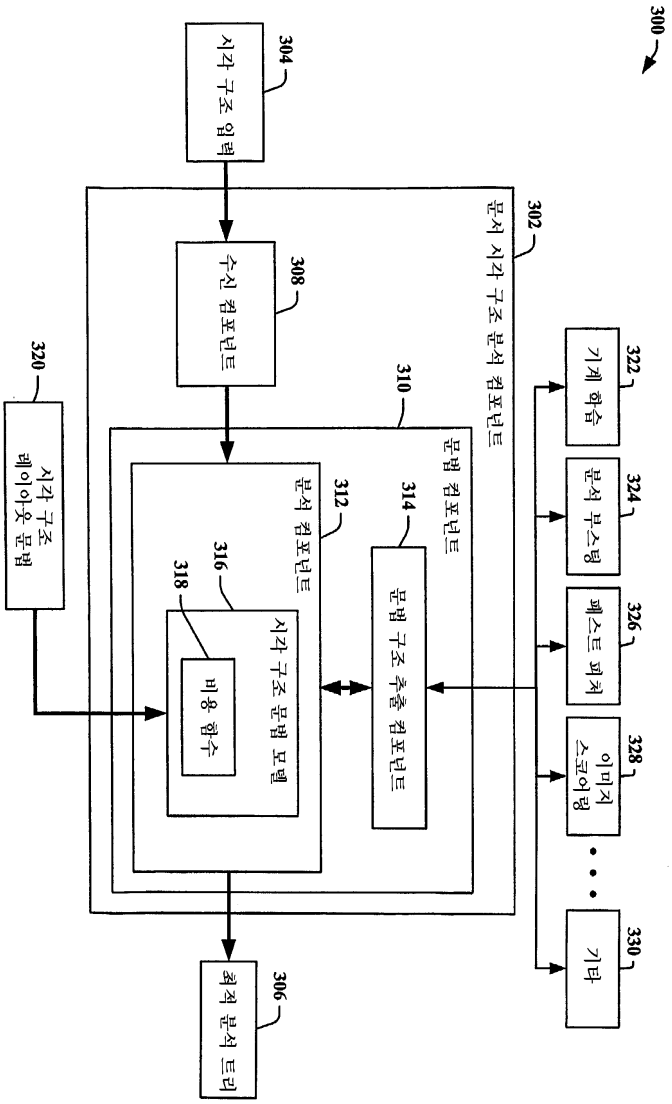
도면1



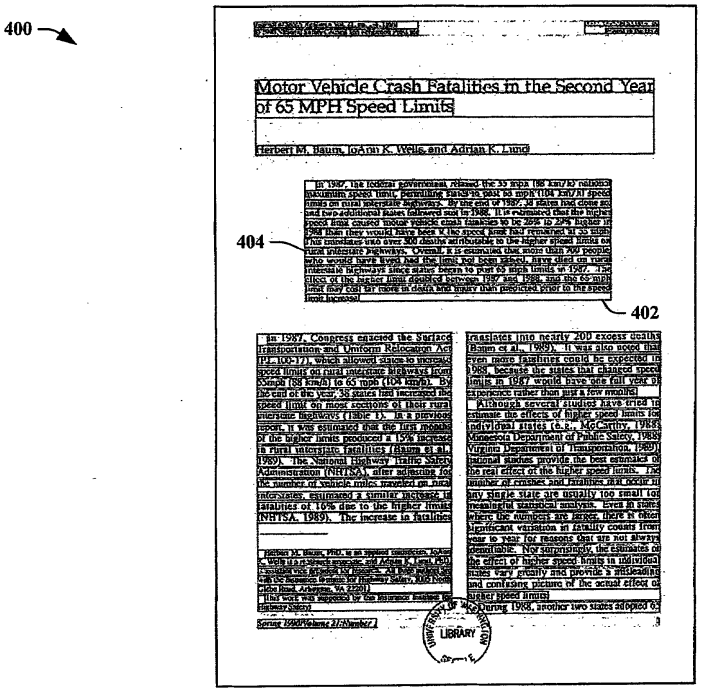
도면2


200
시각 구조 입력
204
문서 시각 구조 분석 컴포넌트
202
수신 컴포넌트
208
문법 컴포넌트
210
기본 구조 레이아웃 문법
212
최적 분석 트리
206

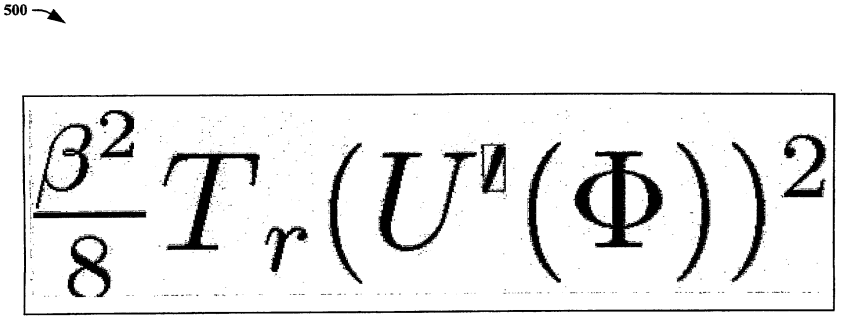
도면3


300
304
시각 구조 입력
302
문서 시각 구조 분석 컴포넌트
308
수신 컴포넌트
310
문법 컴포넌트
312
분석 컴포넌트
314
문법 구조 추출 컴포넌트
316
시각 구조 문법 모델
318
비용 함수
320
시각 구조 레이아웃 문법
306
최적 분석 트리
322
기계 학습
324
분석 부스팅
326
페스트 피처
328
이미지 스코어링
330
기타

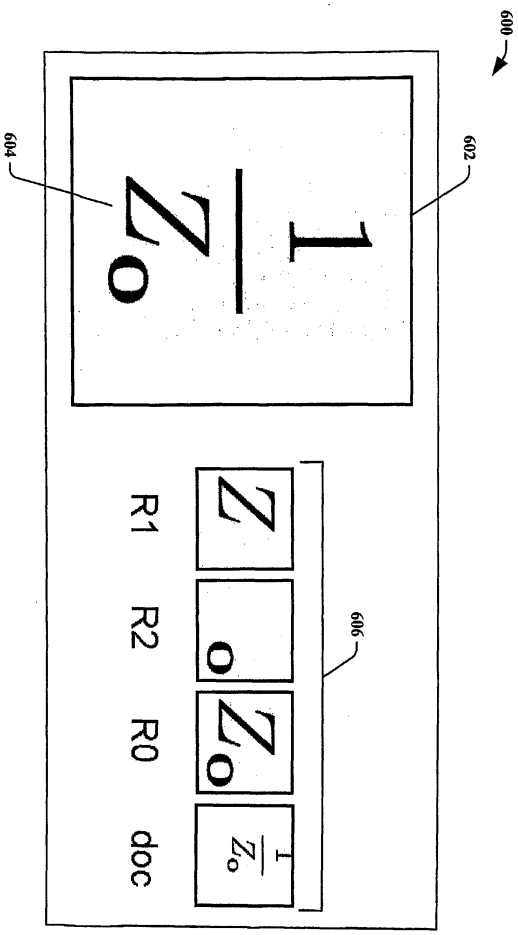
도면4


400
Motor Vehicle Crash Fatalities in the Second Year of 65 MPH Speed Limits
404
402
LIBRARY

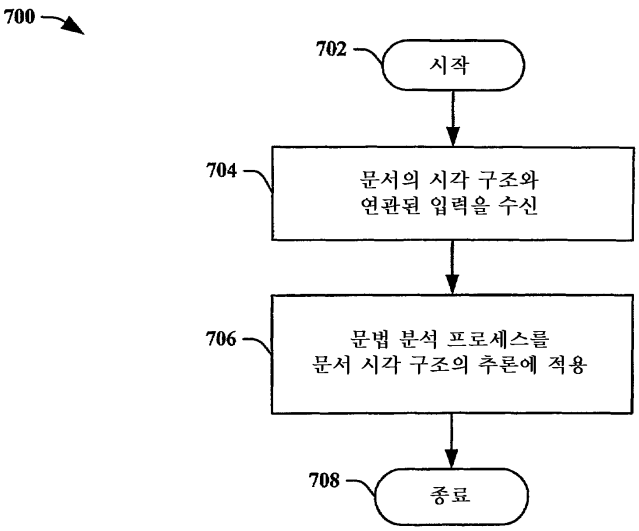
도면5


500
$\frac{\beta^2}{8} Tr\left(U'(\Phi)\right)^2$

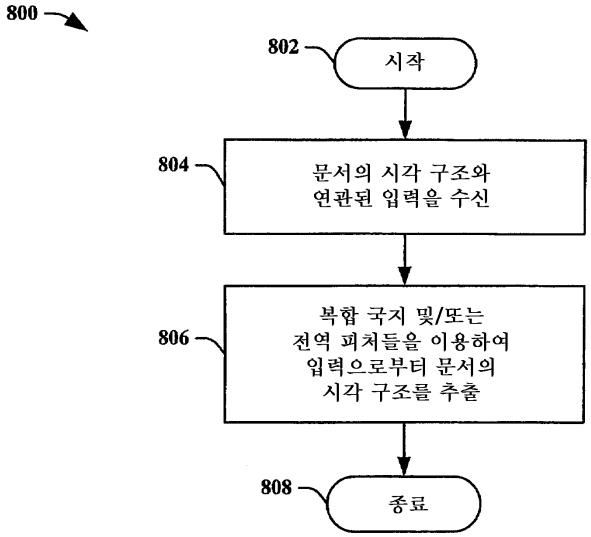
도면6


600
604
602
1/Zo
606
R1
Z
R2
0
R0
Zo
doc
1/Zo

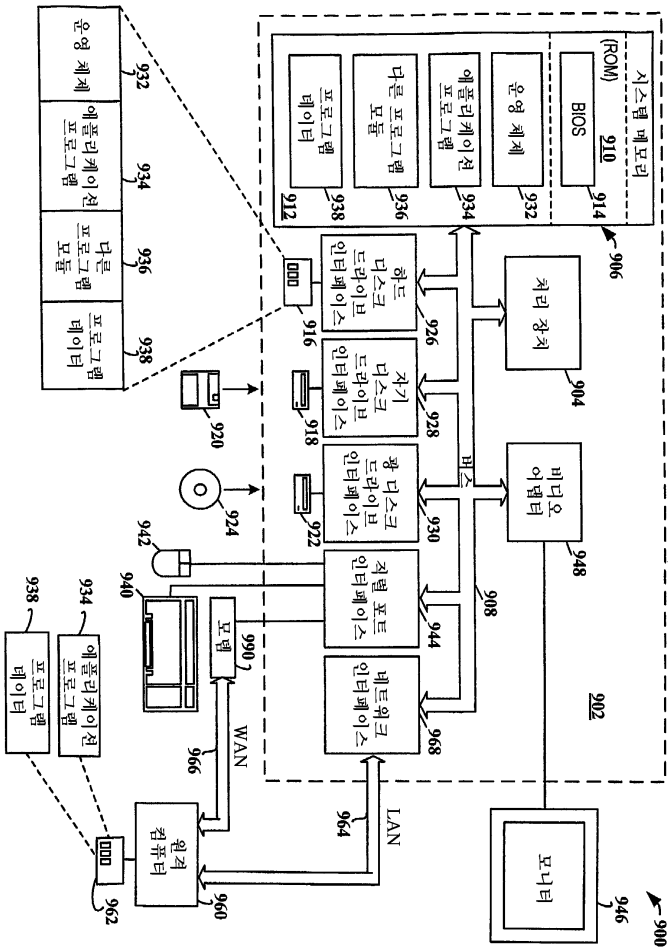
도면7


700
702
시작
704
문서의 시각 구조와
연관된 입력을 수신
706
문법 분석 프로세스를
문서 시각 구조의 추론에 적용
708
종료

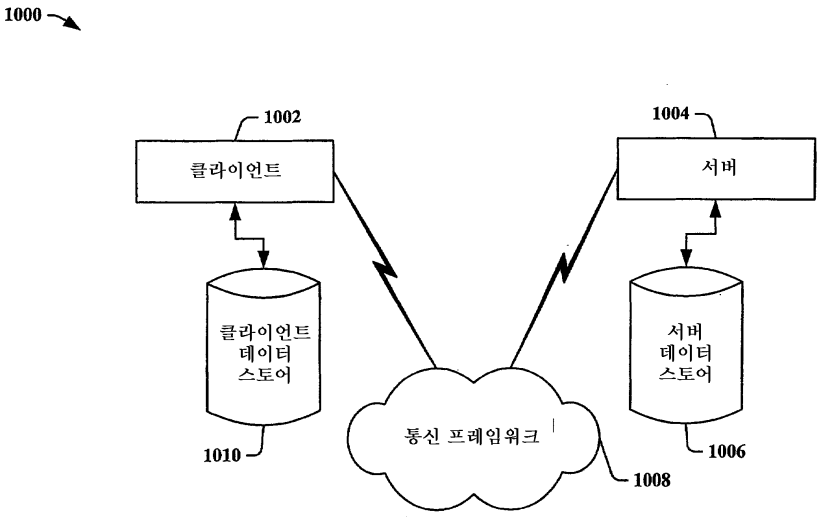
도면8


800
802
시작
804
문서의 시각 구조와
연관된 입력을 수신
806
복합 국지 및/또는
전역 피쳐들을 이용하여
입력으로부터 문서의
시각 구조를 추출
808
종료

도면9


900
902
904
906
908
910
912
914
916
918
920
922
924
926
928
930
932
934
936
938
940
942
944
946
948
960
962
964
966
968
990
시스템 메모리
(ROM)
BIOS
운영 체제
애플리케이션
프로그램
다른 프로그램
모듈
프로그램
데이터
하드
디스크
드라이브
인터페이스
자기
디스크
드라이브
인터페이스
광 디스크
드라이브
인터페이스
직렬 포트
인터페이스
네트워크
인터페이스
처리 장치
비디오
어댑터
버스
모니터
모뎀
WAN
LAN
원격
컴퓨터

도면10


1000
1002
클라이언트
1004
서버
클라이언트 데이터 스토어
서버 데이터 스토어
통신 프레임워크
1010
1006
1008