(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년05월20일
(11) 등록번호 10-1522049
(24) 등록일자 2015년05월14일

(51) 국제특허분류(Int. Cl.)
G06F 17/20 (2006.01) *G06F 17/27* (2006.01)
(21) 출원번호 10-2010-7006475
(22) 출원일자(국제) 2008년08월29일
심사청구일자 2013년08월01일
(85) 번역문제출일자 2010년03월24일
(65) 공개번호 10-2010-0075451
(43) 공개일자 2010년07월02일
(86) 국제출원번호 PCT/US2008/074935
(87) 국제공개번호 WO 2009/029903
국제공개일자 2009년03월05일
(30) 우선권주장
12/200,962 2008년08월29일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
US20050108630 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
밴 덴 버그, 마틴
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내
크로츠, 리차드
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내
(뒷면에 계속)
(74) 대리인
제일특허법인

전체 청구항 수 : 총 16 항 심사관 : 이복현

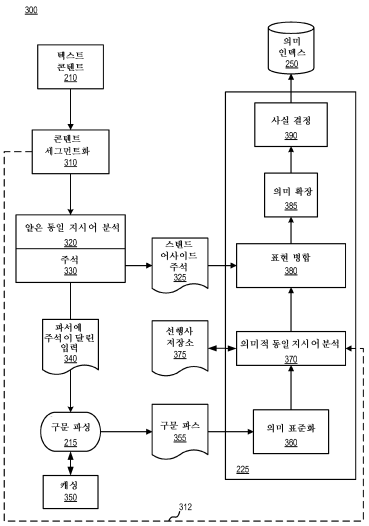
(54) 발명의 명칭 **모호성 민감 자연 언어 처리 시스템에서의 동일 지시어 분석**

(57) 요 약

모호성 민감 자연 언어 처리 시스템에서의 동일 지시어 분석을 위한 기술들이 여기에 설명된다. 자연 언어 처리 시스템에 지시 분석을 통합하기 위한 기법들은 정보 검색 및 검색 시스템 내에서 인덱싱될 문서들을 처리할 수 있다. 모호성 분석 기능뿐만 아니라 모호성 인식 특징들이 동일 지시어 분석과 협조하여 동작할 수 있다. 모호

(뒷면에 계속)

대 표 도 - 도3



한 해석들뿐만 아니라 동일 지시 엔티티들의 주석이 텍스트 콘텐트 내의 인라인 마크업에 의해 또는 외부 엔티티 맵들에 의해 지원될 수 있다. 문서들 내에서 표현된 정보는 사실들, 즉 텍스트 내의 엔티티들 사이의 관계들에 의하여 형식적으로 조직될 수 있다. 확장은 다수의 별칭들, 또는 모호성들을 인덱싱되고 있는 엔티티에 적용하는 것을 지원함으로써, 그 엔티티에 대한 모든 가능한 지시들 또는 해석들이 인덱스 내에 캡처되도록 할 수 있다. 대안적인 저장된 서술들은 본래의 서술에 의해 또는 동일 지시적인 서술에 의해 사실의 검색을 지원할 수 있다.

(72) 발명자

살베티, 프랜코

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

티오네, 지오바니 로렌조

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

안, 데이비드

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

(30) 우선권주장

60/969,426 2007년08월31일 미국(US)

60/969,483 2007년08월31일 미국(US)

명 세 서

청구범위

청구항 1

동일 지시어 분석(coreference resolution) 메커니즘들을 통합하는 방법으로서,

서버의 자연 언어 엔진을 이용하여, 텍스트의 부분을 검색하는 단계;

상기 서버의 상기 자연 언어 엔진을 이용하여, 상기 텍스트의 부분 내에서 동일 지시어(coreference)를 식별하는 단계;

상기 서버의 상기 자연 언어 엔진을 이용하여, 상기 텍스트의 부분으로부터 사실(fact)을 추출하는 단계 - 상기 사실은 의미(meaning)를 가짐 - ; 및

상기 서버의 상기 자연 언어 엔진을 이용하여, 상기 텍스트의 부분 내에서 모호성(ambiguity)을 식별하는 단계;

상기 서버의 상기 자연 언어 엔진을 이용하여, 상기 사실을 확장된 사실로 확장하는 단계 - 상기 확장된 사실은, 상기 의미와 상이하며 상기 식별된 동일 지시어에 기초한 동일 지시어 의미와, 상기 식별된 모호성에 기초한 모호한 의미(ambiguous meaning)를 포함함 -

를 포함하는 방법.

청구항 2

제1항에 있어서, 상기 텍스트의 부분 내에서 상기 동일 지시어를 식별하는 단계는 구문 파싱(syntactic parsing)을 적어도 일부 이용하여 상기 텍스트의 부분 내에서 상기 동일 지시어를 식별하는 단계를 포함하는 방법.

청구항 3

제1항에 있어서, 상기 텍스트의 부분 내에서 상기 동일 지시어를 식별하는 단계는 의미 매핑(semantic mapping)을 적어도 일부 이용하여 상기 텍스트의 부분 내에서 상기 동일 지시어를 식별하는 단계를 포함하는 방법.

청구항 4

제1항에 있어서, 상기 동일 지시어를 식별하는 단계는 모호성을 가지는 동일 지시어를 식별하는 단계를 포함하는 방법.

청구항 5

삭제

청구항 6

삭제

청구항 7

제1항에 있어서, 상기 서버의 상기 자연 언어 엔진을 이용하여, 상기 확장된 사실을 정보 검색을 지원하도록 동작 가능한 인덱스 내에 저장하는 단계를 더 포함하는 방법.

청구항 8

제7항에 있어서, 상기 서버의 상기 자연 언어 엔진을 이용하여, 검색 쿼리(search query)에 응답하여 상기 인덱스로부터 상기 확장된 사실을 검색하는 단계를 더 포함하는 방법.

청구항 9

제1항에 있어서, 상기 서버의 상기 자연 언어 엔진을 이용하여, 상기 텍스트의 부분 내에서 식별된 동일 지시어들에 주석을 다는(annotate) 단계를 더 포함하는 방법.

청구항 10

제2항에 있어서, 상기 서버의 상기 자연 언어 엔진을 이용하여, 상기 구문 파싱으로부터의 정보를 캐싱하는 단계를 더 포함하는 방법.

청구항 11

컴퓨터 실행가능 명령어들이 저장된 컴퓨터 저장 매체로서, 상기 명령어들은 컴퓨터에 의해 실행될 때, 상기 컴퓨터로 하여금,

텍스트의 부분을 검색하고,

상기 텍스트의 부분 내에서 동일 지시어를 식별하고,

상기 텍스트의 부분으로부터 사실을 추출하되, 상기 사실은 의미를 가지고,

상기 텍스트의 부분 내에서 모호성을 식별하고,

상기 식별된 동일 지시어에 기초하여 상기 사실과는 다른 동일 지시어 의미를 가지고, 상기 식별된 모호성에 기초하여 모호한 의미를 가지도록 상기 사실을 확장하게 하는 컴퓨터 저장 매체.

청구항 12

제11항에 있어서, 상기 동일 지시어를 식별하는 것은 구문 파싱을 적어도 일부 이용하여 상기 텍스트의 부분 내에서 상기 동일 지시어를 식별하는 것을 포함하는 컴퓨터 저장 매체.

청구항 13

제11항에 있어서, 상기 동일 지시어를 식별하는 것은 의미 매핑(semantic mapping)을 적어도 일부 이용하여 상기 텍스트의 부분 내에서 상기 동일 지시어를 식별하는 것을 포함하는 컴퓨터 저장 매체.

청구항 14

제11항에 있어서, 상기 동일 지시어를 식별하는 것은 모호성을 가지는 동일 지시어를 식별하는 것을 포함하는 것인 컴퓨터 저장 매체.

청구항 15

삭제

청구항 16

삭제

청구항 17

제11항에 있어서, 상기 컴퓨터로 하여금 상기 확장된 사실을 정보 검색을 지원하도록 동작 가능한 인덱스 내에 저장하게 하는 명령어를 더 포함하는 컴퓨터 저장 매체.

청구항 18

제17항에 있어서, 상기 컴퓨터로 하여금 검색 쿼리에 응답하여 상기 인덱스로부터 상기 확장된 사실을 검색하게 하는 명령어를 더 포함하는 컴퓨터 저장 매체.

청구항 19

제11항에 있어서, 상기 컴퓨터로 하여금 상기 텍스트의 부분 내에서 식별된 동일 지시어들에 주석을 달게 하는 명령어를 더 포함하는 컴퓨터 저장 매체.

청구항 20

동일 지시어 분석 메커니즘들을 통합하는 방법으로서,

서버 컴퓨터의 자연 언어 엔진을 이용하여, 텍스트의 부분을 검색하는 단계;

상기 서버 컴퓨터의 상기 자연 언어 엔진을 이용하여, 상기 텍스트의 부분 내에서 동일 지시어를 식별하는 단계;

상기 서버 컴퓨터의 상기 자연 언어 엔진을 이용하여, 상기 텍스트의 부분 내에서 모호성을 식별하는 단계;

상기 서버 컴퓨터의 상기 자연 언어 엔진을 이용하여, 상기 텍스트의 부분으로부터 사실을 추출하는 단계 - 상기 사실은 의미를 가짐 - ;

상기 서버 컴퓨터의 상기 자연 언어 엔진을 이용하여, 상기 사실을 확장하는 단계 - 상기 확장된 사실은, 상기 의미와 상이하며 상기 식별된 동일 지시어에 기초한 동일 지시어 의미와, 상기 식별된 모호성에 기초한 모호한 의미(ambiguous meaning)를 포함함 - ;

상기 확장된 사실을 정보 검색을 지원하도록 동작 가능한 인덱스 내에 저장하는 단계; 및

검색 쿼리에 응답하여 상기 인덱스로부터 상기 확장된 사실을 검색하는 단계

를 포함하는 방법.

발명의 설명

기 술 분 야

[0001] 자연 언어에서, 엔티티(entity)들을 상이한 서술들에 의해 지시하는 것은 드문 일이 아니다. 예를 들면, 명사들을 대신하기 위해 일반적으로 대명사들이 이용된다. 또한, 한 엔티티를 지시하기 위해 지시(reference)의 다양한 다른 서술들, 또는 상이한 형태들이 이용될 수 있다. 예로서 텍스트의 다음 부분들을 생각해보자:

[0002] "Pablo Picasso was born in Malaga."

[0003] "The Spanish painter became famous for his varied styles."

[0004] "Among his paintings is the large-scale Guernica."

[0005] "He painted this disturbing masterpiece during the Spanish Civil War."

[0006] "Picasso died in 1973."

[0007] 다양한 언어 변화에 직면한다. 예를 들면, "Pablo Picasso" 및 "Picasso"라는 2개의 상이한 이름이 사용된다. 한정하는 서술인 "the Spanish painter" 및 2개의 대명사 "his" 및 "he"는 모두 Picasso를 지시하기 위해 사용된다. 그림(painting)을 지시하기 위해 2개의 상이한 표현이 사용된다: 작품의 이름인 "Guernica" 및 지시 서술(demonstrative description)인 "this disturbing masterpiece."

[0008] 2개의 언어 표현들은 그것들이 동일한 지시 대상(referent)을 갖는다면 동일 지시적이라고 할 수 있다. 바꾸어 말하면, 그것들이 동일한 엔티티를 지시한다는 가정이다. 두 번째 어구는 첫 번째 어구에 조응적인(anaphoric) 전방조응사(anaphor)일 수 있다. 그러므로, 첫 번째 어구는 두 번째 어구의 선행사(antecedent)이다. 전방조응사의 지시 대상을 판정하기 위해 선행사의 지시 대상에 대한 지식이 필요할 수 있다. 문서 내에서 동일 지시적인 표현들, 전방조응사들, 및 그들의 선행사들을 찾아내는 일반적인 작업은 동일 지시어 분석(coreference resolution)이라고 불릴 수 있다. 동일 지시어 분석은 2개의 표현들이 동일한 지시 대상을 지시하는 것을 확립하는 프로세스이고, 반드시 그 지시 대상이 무엇인지를 확립하는 것은 아니다. 지시 분석(reference resolution)은 그 지시 대상이 무엇인지를 확립하는 프로세스이다.

[0009] 동일 지시적인 표현들의 집단(cluster)들에 대하여, 그들의 조응적 관계들에 관계없이, 그 표현들은 서로의 별칭(alias)들이라고 불릴 수 있다. 상기 예에 따르면, 표현들 "Pablo Picasso", "the Spanish painter", "his", "he", 및 "Picasso"는 Picasso를 지시하는 별칭 집단(alias cluster)을 형성한다.

[0010] 자연 언어 표현들은 종종 모호성(ambiguity)을 나타낸다. 모호성은 한 표현이 2개 이상의 의미로 해석될 수 있을 때 일어난다. 예를 들면, 문장 "The duck is ready to eat(오리는 먹을 준비가 되어 있다)"는 오리가 적당

하게 요리되어 있다는 것 또는 오리가 배고파서 모이를 줄 필요가 있다는 것을 주장하는 것으로 해석될 수 있다.

[0011] 동일 지시어 분석 및 모호성 분석은 인간 사용자들에 의해 일반적으로 표현되는 언어를 기계적으로 지원하는 데 이용될 수 있는 자연 언어 처리 동작들의 2가지 예들이다. 정보 검색을 지원하는 텍스트 인덱싱 및 쿼링(querying)과 같은 정보 처리 시스템들은 자연 언어 처리 시스템들의 증가된 적용으로부터 이익을 얻을 수 있다.

[0012] 여기에 작성된 명세서가 제시되는 것은 이러한 사정들 및 다른 사정들에 관련한 것이다.

발명의 내용

[0013] 모호성 민감 자연 언어 처리 시스템에서의 동일 지시어 분석을 위한 기술들이 여기에 설명된다. 특히, 정보 검색 및 검색 시스템 내에서 인덱싱될 문서들을 처리하기 위한 시스템에 동일 지시어 분석 기능을 통합하기 위한 기법들이 설명된다. 이 통합은 자연 언어 문서들 내의, 동일 지시어 분석, 및 모호한 의미를 지원하는 정보에 의한 인덱싱을 향상시킬 수 있다.

[0014] 여기에 제시된 일 양태에 따르면, 동일 지시어 분석 시스템에 의해 제공되는 정보가 자연 언어 처리 시스템에 통합되어 자연 언어 처리 시스템의 성능을 개선할 수 있다. 그러한 시스템의 일례는 문서 인덱싱 및 검색 시스템이다.

[0015] 여기에 제시된 다른 양태에 따르면, 모호성 분석 기능뿐만 아니라 모호성 인식 특징들이 자연 언어 처리 시스템 내의 동일 지시어 분석과 협조하여 동작할 수 있다. 모호한 해석뿐만 아니라 동일 지시 엔티티들의 주석이 텍스트 표현들 내의 인라인 마크업(in-line markup)에 의해 또는 대안적으로는 외부 엔티티 맵들(external entity maps)에 의해 지원될 수 있다.

[0016] 여기에 제시된 또 다른 양태에 따르면, 인덱싱될 텍스트로부터 사실(fact)들이 추출될 수 있다. 텍스트 내에 표현된 정보는 사실들에 의하여 형식적으로 조직될 수 있다. 이러한 의미에서 사용되는 경우, 사실은 텍스트에 포함된 임의의 정보일 수 있고, 반드시 진실일 필요는 없다. 사실은 엔티티들 사이의 관계로서 표현될 수 있다. 사실은 의미 인덱스(semantic index) 내에 저장된 엔티티들 사이의 관계로서 상기 의미 인덱스에 저장될 수 있다. 사실 기반 검색 시스템에서, 문서는 그것이 쿼리의 분석을 통하여 판정된 사실과 부합하는 사실을 포함한다면 검색될 수 있다.

[0017] 여기에 제시된 또 다른 양태에 따르면, 확장의 프로세스가 다수의 별칭들, 또는 모호성들을 인덱싱되고 있는 엔티티에 적용하는 것을 지원할 수 있다. 그러한 확장은 의미 인덱스에 캡처되고 있는 주어진 엔티티에 대하여, 추가적인 가능한 지시들, 또는 해석들을 지원할 수 있다. 대안적인 저장된 서술들은 본래의 서술에 의해 또는 동일 지시적인 서술에 의해 사실의 검색을 지원할 수 있다.

[0018] 전술한 청구 대상은 또한 컴퓨터 제어되는 장치, 컴퓨터 프로세스, 컴퓨팅 시스템, 또는 컴퓨터 판독 가능한 매체와 같은 제조물로서 구현될 수 있다는 것을 이해해야 한다. 이들 및 다양한 다른 특징들은 다음의 상세한 설명을 읽고 관련 도면들을 검토하는 것으로부터 명백할 것이다.

[0019] 본 요약은 아래 상세한 설명에서 더 설명되는 개념들 중 선택된 것을 단순화된 형태로 소개하기 위해 제공된다. 본 요약은 청구된 내용의 중요한 특징들 또는 본질적인 특징들을 일치시키려 의도된 것이 아니며, 또한 본 요약은 청구된 내용의 범위를 제한하는 데 이용되도록 의도된 것도 아니다. 또한, 청구된 내용은 이 명세서의 임의의 부분에서 지적된 임의의 또는 모든 불리점들을 해결하는 구현들에 제한되지 않는다.

도면의 간단한 설명

[0020] 도 1은 본 명세서에 제시된 실시예의 양태들에 따른 정보 검색 시스템을 예시하는 네트워크 아키텍처 도이다.

도 2는 본 명세서에 제시된 실시예의 양태들에 따른 자연 언어 인덱스 및 쿼리 시스템의 다양한 컴포넌트들을 예시하는 기능 블록도이다.

도 3은 본 명세서에 제시된 실시예의 양태들에 따른 자연 언어 처리 시스템 내의 동일 지시어 분석 및 모호성 분석을 예시하는 기능 블록도이다.

도 4는 본 명세서에 제시된 실시예의 양태들에 따른 동일 지시어 분석에 의한 모호성 민감 인덱싱을 위한 프로

세스들의 양태들을 예시하는 논리 흐름도이다.

도 5는 본 명세서에 제시된 실시예의 양태들을 구현할 수 있는 컴퓨팅 시스템에 대한 예시적인 컴퓨터 하드웨어 및 소프트웨어 아키텍처를 보여주는 컴퓨터 아키텍처 도이다.

발명을 실시하기 위한 구체적인 내용

[0021] 다음의 상세한 설명은 모호성 민감 자연 언어 처리 시스템에서의 동일 지시어 분석을 위한 기술들에 관한 것이다. 여기에 제시된 기술들 및 개념들의 이용을 통하여, 정보 검색 및 검색 시스템에서 사용하기 위해 인덱싱될 문서들을 처리하는 자연 언어 처리 시스템에 동일 지시어 분석 기능이 통합될 수 있다. 이 통합은 인덱싱되고 있는 자연 언어 문서들에 대한 동일 지시어 분석을 지원하는 정보에 의한 인덱스를 향상시킬 수 있다.

[0022] 여기에 설명된 내용은 컴퓨터 시스템 상의 운영 체제 및 애플리케이션 프로그램들의 실행과 관련하여 실행하는 프로그램 모듈들의 일반적인 컨텍스트에서 제시되지만, 숙련된 당업자들은 다른 유형의 프로그램 모듈들과 함께 다른 구현들이 수행될 수 있다는 것을 인지할 것이다. 일반적으로, 프로그램 모듈들은 특정한 태스크들을 수행하거나 특정한 추상 데이터 유형들을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조, 및 기타 유형의 구조를 포함한다. 또한, 숙련된 당업자들은 여기에 제시된 내용은 핸드헬드 장치, 마이크로프로세서 시스템, 마이크로프로세서 기반 또는 프로그램 가능한 소비자 전자 장치, 미니컴퓨터, 메인프레임 컴퓨터 등을 포함하는 다른 컴퓨터 시스템 구성들과 함께 실시될 수 있다는 것을 알 것이다.

[0023] 다음의 상세한 설명에서는, 본 명세서의 일부를 형성하고, 특정한 실시예들 또는 예들이 예시로서 도시되어 있는 첨부 도면들이 참조된다. 이제, 몇몇 도면들을 통하여 유사한 참조번호들이 유사한 엘리먼트들을 나타내는 도면들을 참조하여, 모호성 민감 자연 언어 처리 시스템에서의 동일 지시어 분석을 위한 컴퓨팅 시스템 및 방법의 양태들을 설명한다.

[0024] 이제 도 1을 참조하여, 여기에 제시된 구현들에 대한 예시적인 동작 환경에 관하여 상세히 설명한다. 특히, 네트워크 아키텍처 다이어그램(100)은 여기에 제시된 실시예의 양태들에 따른 정보 검색 시스템을 예시한다. 클라이언트 컴퓨터들(110A-110D)은 자연 언어 엔진(130)과 관련된 정보를 얻기 위해 네트워크(140)를 통하여 서버(120)에 인터페이스할 수 있다. 4개의 클라이언트 컴퓨터들(110A-110D)이 예시되어 있지만, 임의의 수의 클라이언트 컴퓨터들(110A-110D)이 이용될 수 있다는 것을 이해해야 한다. 클라이언트 컴퓨터들(110A-110D)은 네트워크(140)에 걸쳐서 지리적으로 분산되거나, 한 곳에 배치되거나, 또는 그의 임의의 조합으로 될 수 있다. 단 하나의 서버(120)가 예시되어 있지만, 서버(120)의 기능은 임의의 수의 다수의 서버들(120)에 걸쳐서 분산될 수도 있다는 것을 이해해야 한다. 그러한 다수의 서버들(120)은 한 곳에 배치되거나, 네트워크(140)에 걸쳐서 지리적으로 분산되거나, 또는 그의 임의의 조합으로 될 수 있다.

[0025] 하나 이상의 실시예들에 따르면, 자연 언어 엔진(130)은 검색 엔진 기능을 지원할 수 있다. 검색 엔진 시나리오에서는, 클라이언트 컴퓨터(110A-110D)로부터 네트워크(140)를 통하여 서버(120)로 사용자 쿼리가 발행될 수 있다. 사용자 쿼리는 자연 언어 포맷일 수 있다. 서버에서, 자연 언어 엔진(130)은 자연 언어 쿼리를 처리하여 자연 언어 쿼리로부터 추출된 구문 및 의미에 기초하여 검색을 지원할 수 있다. 그러한 검색의 결과들은 서버(120)로부터 네트워크(140)를 통하여 클라이언트 컴퓨터들(110A-110D)로 제공될 수 있다.

[0026] 하나 이상의 검색 인덱스들이 서버(120)에 저장되거나, 또는 관련될 수 있다. 검색 인덱스 내의 정보는 소스 정보의 세트, 또는 코퍼스(corpus)로부터 파퓰레이트(populate)될 수 있다. 예를 들면, 웹 검색 구현에서, 네트워크(140)에 걸쳐서 다양한 웹 서버들(도시되지 않음) 상의 다양한 웹 사이트들로부터 콘텐트가 수집(collect)되고 인덱싱될 수 있다. 그러한 수집 및 인덱싱은 서버(120) 상에서, 또는 다른 컴퓨터(도시되지 않음) 상에서 실행하는 소프트웨어에 의해 수행될 수 있다. 수집은 웹 크롤러(web crawlers) 또는 스파이더(spider) 애플리케이션에 의해 수행될 수 있다. 자연 언어 엔진(130)은 코퍼스로부터 수집된 자연 언어 콘텐트가 자연 언어 엔진(130)에 의해 추출된 구문 및 의미에 기초하여 인덱싱될 수 있도록 수집된 정보에 적용될 수 있다. 인덱싱 및 검색은 도 2에 관련하여 더 상세히 논의된다.

[0027] 클라이언트 컴퓨터들(110A-110D)은 서버(120)에 대해 터미널 클라이언트, 하이퍼텍스트 브라우저 클라이언트, 그래픽 디스플레이 클라이언트, 또는 다른 네트워킹된 클라이언트들로서 기능할 수 있다. 예를 들면, 클라이언트 컴퓨터들(110A-110D)에 있는 웹 브라우저 애플리케이션은 서버(120)에 있는 웹 서버 애플리케이션과의 인터페이싱을 지원할 수 있다. 그러한 브라우저는 서버(120)에의 인터페이싱을 지원하기 위해 컨트롤(controls), 플러그인(plug-ins), 또는 애플릿(applets)을 이용할 수 있다. 클라이언트 컴퓨터들(110A-110D)은 또한 서버(120)와 인터페이스하기 위해 다른 사용자 지정된 프로그램, 애플리케이션, 또는 모듈을 이용할 수 있다. 클라

이언트 컴퓨터들(110A-110D)은 데스크톱 컴퓨터, 랩톱, 핸드헬드, 이동 단말기, 이동 전화기, 텔레비전 셋톱 박스, 키오스크, 서버, 터미널, 씬 클라이언트(thin-clients), 또는 임의의 다른 컴퓨터화된 장치일 수 있다.

[0028] 네트워크(140)는 클라이언트 컴퓨터들(110A-110D)과 서버(120) 사이의 통신을 지원할 수 있는 임의의 통신 네트워크일 수 있다. 네트워크(140)는 유선, 무선, 광학, 라디오, 패킷 교환, 회선 교환, 또는 그의 임의의 조합일 수 있다. 네트워크(140)는 임의의 토폴로지를 이용할 수 있고, 네트워크(140)의 링크들은 이더넷, DSL, 케이블 모뎀, ATM, SONET, MPLS, PSTN, POTS 모뎀, PONS, HFC, 위성, ISDN, 와이파이(WiFi), 와이맥스(WiMax), 모바일 셀룰러(mobile cellular), 그의 임의의 조합, 또는 임의의 다른 데이터 상호접속 또는 네트워킹 메커니즘과 같은 임의의 네트워킹 기술, 프로토콜, 또는 대역폭을 지원할 수 있다. 네트워크(140)는 인트라넷, 인터넷(internet), 인터넷(the Internet), 월드 와이드 웹, LAN, WAN, MAN, 또는 컴퓨터 시스템들의 상호접속을 위한 임의의 다른 네트워크일 수 있다.

[0029] 예시된 네트워크 환경 외에도, 자연 언어 엔진(130)은 로컬로(locally) 운영될 수 있다는 것을 이해해야 한다. 예를 들면, 서버(120) 및 클라이언트 컴퓨터들(110A-110D)은 단일 컴퓨팅 장치로 결합될 수 있다. 그러한 결합된 시스템은 로컬로 또는 원격으로 저장된 검색 인덱스들을 지원할 수 있다.

[0030] 이제 도 2를 참조하면, 기능 블록도는 하나의 예시적인 실시예에 따른 자연 언어 엔진(130)의 다양한 컴포넌트들을 예시한다. 위에 논의된 바와 같이, 자연 언어 엔진(130)은 정보 검색들을 지원할 수 있다. 그러한 검색들을 지원하기 위하여, 콘텐트 획득 프로세스(200)가 수행된다. 콘텐트 획득(200)에 관련된 동작들은 텍스트 콘텐트(210)로서 제공된 문서들로부터 정보를 추출한다. 이 정보는 검색을 위해 이용될 수 있는 의미 인덱스(250)에 저장될 수 있다. 사용자 검색(205)에 관련된 동작들은 사용자 입력 검색 쿼리의 처리를 지원할 수 있다. 사용자 쿼리는 자연 언어 질문(260)의 형태를 취할 수 있다. 자연 언어 엔진(130)은 사용자 입력을 분석하여 쿼리를 의미 인덱스(250) 내에 표현된 정보와 비교될 표현으로 변환할 수 있다. 의미 인덱스(250) 내의 정보의 콘텐트 및 구조는, 쿼리 또는 자연 언어 질문(260)의 의미에 관련이 있는, 문서들, 또는 문서들의 부분들의 신속한 매칭 및 검색을 지원할 수 있다.

[0031] 텍스트 콘텐트(210)는 매우 일반적인 의미의 문서들을 포함할 수 있다. 그러한 문서들의 예들은 웹 페이지, 텍스트 문서, 스캔된 문서, 데이터베이스, 정보 목록, 기타 인터넷 콘텐트, 또는 임의의 다른 정보 소스를 포함할 수 있다. 이 텍스트 콘텐트(210)는 검색될 정보의 코퍼스를 제공할 수 있다. 텍스트 콘텐트(210)를 처리하는 것은 구문 파싱(syntactic parsing)(215) 및 의미 매핑(semantic mapping)(225)으로서 2개의 스테이지들에서 일어날 수 있다. 파싱(215)의 전에 또는 파싱(215)의 처음에 예비 언어 처리 단계들이 일어날 수 있다. 예를 들면, 텍스트 콘텐트(210)는 문장 경계들에서 분리될 수 있다. 특정한 사람들, 장소들, 물체들 또는 이벤트들의 이름들로서 적당한 명사들이 식별될 수 있다. 또한, 의미 있는 단어 끝부분들의 문법적 속성들이 판정될 수 있다. 예를 들면, 영어에서, "s"로 끝나는 명사는 복수의 명사일 것 같은 반면, "s"로 끝나는 동사는 3인칭 단수의 동사일 수 있다.

[0032] 파싱(215)은, 본 명세서에서 단지 일반적인 예로서 제공되지만, 이러한 설명의 가능한 구현들을 제한하기 위한 것은 아닌, XLE(Xerox Linguistic Environment)와 같은 구문 분석 시스템에 의해 수행될 수 있다. 파서(parser)(215)는 문장들을 단어들 사이의 구문 관계들을 명백하게 하는 표현들로 변환할 수 있다. 파서(215)는 사용되고 있는 특정 언어와 관련된 문법(220)을 적용할 수 있다. 예를 들면, 파서(215)는 영어에 대한 문법(220)을 적용할 수 있다. 문법(220)은, 예를 들면, LFG(lexical functional grammar) 또는 HPSG(Head-Driven Phrase Structure Grammar), CCG(Combinatory Categorial Grammar), PCFG(Probabilistic Context-free Grammar) 또는 임의의 다른 문법 포멀리즘에 기초한 것들과 같은 다른 적합한 파싱 메커니즘으로서 형식화될 수 있다. 문법(220)은 주어진 언어에서 의미 있는 문장들을 구성하기 위한 가능한 방법들을 특정할 수 있다. 파서(215)는 텍스트 콘텐트(210)의 문자열들에 문법(220)의 규칙들을 적용할 수 있다.

[0033] 문법(220)은 다양한 언어들에 대하여 제공될 수 있다. 예를 들면, LFG 문법들은 영어, 불어, 독어, 중국어, 및 일어에 대하여 생성되었다. 다른 문법들이 제공될 수도 있다. 문법(220)은 언어학자 또는 사전 저자에 의해 문법 규칙들이 정의되는 수동 획득(manual acquisition)에 의해 개발될 수 있다. 대안적으로, 기계 학습 획득은 문법 규칙들을 자동으로 판정하기 위해 큰 코퍼스로부터의 텍스트의 많은 예들의 자동화된 관찰 및 분석을 수반할 수 있다. 수동 정의 및 기계 학습의 조합이 문법(220)의 규칙들을 획득하는 데 이용될 수도 있다.

[0034] 파서(215)는 구문 구조를 판정하기 위해 텍스트 콘텐트(210)에 문법(220)을 적용할 수 있다. LFG 기반 파싱의 경우에, 구문 구조들은 구성 요소 구조들(constituent structures)(c-구조들) 및 기능 구조들(functional structures)(f-구조들)로 이루어진다. c-구조는 구성 요소 구들 및 단어들의 계층 구조를 나타낼 수 있다. f-

구조는 c-구조의 다양한 구성 요소들 사이의 역할들 및 관계들을 인코딩할 수 있다. f-구조는 또한 단어들의 형태들로부터 도출되는 정보를 나타낼 수 있다. 예를 들면, 명사의 복수 또는 동사의 시제는 f-구조에서 특정될 수 있다.

[0035] 파싱 프로세스(215)의 다음에 오는 의미 매핑 프로세스(225) 동안에는, 구문 구조들로부터 정보가 추출되어 문장 내의 단어들의 의미들에 관한 정보와 조합될 수 있다. 문장의 의미 맵(semantic map) 또는 의미 표현이 콘텐트 의미(content semantics)(240)로서 제공될 수 있다. 의미 매핑(225)은 파서(215)에 의해 제공된 구문 관계들에 개개의 단어들의 개념적 속성들을 추가(augment)할 수 있다. 그 결과들은 텍스트 콘텐트(210)로부터의 문장들의 의미의 표현들로 변환될 수 있다. 의미 매핑(225)은 문장 내의 단어들에 의해 수행되는 역할들을 판정할 수 있다. 예를 들면, 액션을 수행하는 주체, 액션을 수행하는 데 이용되는 어떤 것, 또는 액션에 의해 영향을 받는 어떤 것. 검색 인덱싱을 위하여, 단어들의 그들의 역할과 함께 의미 인덱스(250)에 저장될 수 있다. 따라서, 의미 인덱스(250)로부터의 검색은 단지 분리된 단어에만 의존하지 않고, 텍스트 콘텐트(210)에서 그 단어가 나타나는 문장들 내의 의미에 의존할 수도 있다. 의미 매핑(225)은 용어들의 명확화(disambiguation), 선행사 관계들의 판정, 및 동의어(synonym), 상위어(hypernym), 또는 하위어(hyponym)에 의한 용어들의 확장을 지원할 수 있다.

[0036] 의미 매핑(225)은 문장들로부터 의미들을 추출하기 위한 기법들 및 규칙들로서 지식 리소스들(230)을 적용할 수 있다. 지식 리소스들은, 문법들(220)의 획득에 관하여 논의한 바와 같이, 수동 정의 및 기계 학습 양쪽 모두를 통하여 획득될 수 있다. 의미 매핑(225) 처리는 의미 확장 가능 마크업 언어(semantic XML 또는 semxml) 표현으로 콘텐트 의미들(240)을 제공할 수 있다. PROLOG, LISP, JSON, YAML 등으로 작성된 표현들과 같은 임의의 적합한 표현 언어가 또한 이용될 수 있다. 콘텐트 의미들(240)은 텍스트 콘텐트(210)의 문장들 내의 단어들에 의해 수행되는 역할들을 특정할 수 있다. 콘텐트 의미들(240)은 인덱싱 프로세스(245)에 제공될 수 있다.

[0037] 인덱스는 단어들 및 구들의 위치들이 인덱스 내에서 신속히 식별될 수 있도록 정보의 큰 코퍼스를 나타내는 것을 지원할 수 있다. 전통적인 검색 엔진은 인덱스가 사용자에 의해 특정된 키워드들로부터 그 키워드들이 나타나는 기사들 또는 문서들에 매핑하도록 검색어들로서 키워드들을 이용할 수 있다. 의미 인덱스(250)는 단어들 자체에 더하여 단어들의 의미론적 뜻을 나타낼 수 있다. 콘텐트 획득(200) 및 사용자 검색(205) 동안에 단어들에 의미 관계들이 할당될 수 있다. 의미 인덱스(250)에 대한 쿼리들은 단어들만이 아니라, 특정 역할들의 단어들에 기초할 수 있다. 그 역할들은 의미 인덱스(250)에 저장된 문장 또는 구에서 그 단어에 의해 수행되는 역할들이다. 의미 인덱스(250)는 그의 항목들이 의미 단어들(즉, 주어진 역할의 단어)과 그 단어들이 나타나는 문서들, 또는 웹 페이지들에의 포인터들인 신속히 검색 가능한 데이터베이스인 반전된 인덱스로 간주될 수 있다. 의미 인덱스(250)는 하이브리드 인덱싱을 지원할 수 있다. 그러한 하이브리드 인덱싱은 키워드 인덱싱 및 의미 인덱싱 양쪽 모두의 특징들 및 기능들을 조합할 수 있다.

[0038] 쿼리들의 사용자 입력은 자연 언어 질문들(260)의 형태로 지원될 수 있다. 쿼리는 콘텐트 획득(200)에서 사용된 것과 유사한, 또는 동일한 자연 언어 파이프라인을 통하여 분석될 수 있다. 즉, 자연 언어 질문(260)은 구문 구조를 추출하기 위해 파서(265)에 의해 처리될 수 있다. 구문 파싱(265)에 이어서, 자연 언어 질문(260)은 의미 매핑(270)을 위해 처리될 수 있다. 의미 매핑(270)은 위에 논의된 바와 같이 의미 인덱스(250)에 대하여 검색 프로세스(280)에서 이용될 질문 의미들(275)을 제공할 수 있다. 검색 프로세스(280)는 키워드 인덱스 검색 및 의미 인덱스 검색 양쪽 모두가 단독으로 또는 조합하여 제공될 수 있는 하이브리드 인덱스 쿼리들을 지원할 수 있다.

[0039] 사용자 쿼리에 응답하여, 질문 의미들(275)과 함께 의미 인덱스(250)로부터의 검색(280)의 결과들은 랭킹 프로세스(285)에 통지할 수 있다. 랭킹은 키워드 및 의미 정보 양쪽 모두를 이용할 수 있다. 랭킹(285) 동안에, 검색(280)에 의해 얻어진 결과들은 가장 바람직한 결과들을 결과 프리젠테이션(290)으로서 사용자에게 제공될 검색된 정보의 최상부에 더 가까이 배치하려는 시도에서 다양한 메트릭들에 의해 정리(order)될 수 있다.

[0040] 이제 도 3을 참조하면, 기능 블록도가 여기에 제시된 실시예의 양태들에 따른 자연 언어 처리 시스템(300) 내의 동일 지시어 분석 및 모호성 분석을 예시한다. 애플리케이션의 예로서, 자연 언어 처리 시스템(300)은 문서 인덱싱 및 검색을 위한 정보 검색 엔진을 지원할 수 있다. 그러한 자연 언어 가능한 검색 엔진은 언어 분석에 기초하여 그의 인덱스 내에 저장된 정보를 확장할 수 있다. 시스템은 또한 쿼리를 언어적으로 분석함으로써 사용자 쿼리 내의 의도의 발견을 지원할 수 있다. 여기에 논의된 동일 지시어 분석 및 모호성 분석 특징들은 도 2에 관하여 논의된 바와 같이 구문 파싱(215), 의미 매핑(225), 및 의미 인덱싱(245)에 관련하여 동작할 수 있다. 동일 지시어 분석은 텍스트 콘텐트(210)에 대해 직접 수행되거나, 또는 파싱(215) 또는 의미 매핑(225)

동작들로부터의 정보를 이용할 수 있다.

[0041] 예시된 바와 같이, 동일 지시어 분석(320, 370)은 세그먼트화된 문서에 대해 직접적으로 또한 의미 매핑(225)의 일부로서 수행될 수 있다. 동일 지시어 분석(320, 370)의 이들 2개의 발생들이 병합될 수 있거나 또는 그들의 정보 출력들이 병합될 수 있다. 동일 지시어 분석은 또한 구문 파싱(215)과 의미 매핑(225) 사이에서 발생할 수도 있다는 것을 이해해야 한다. 동일 지시어 분석은 또한 자연 언어 처리 파이프라인 내의 임의의 다른 스테이지에서 발생할 수 있다. 자연 언어 처리 시스템 내의 다양한 위치에 1개, 2개, 또는 그 이상의 동일 지시어 분석 컴포넌트들, 또는 스테이지들이 있을 수 있다. 텍스트 콘텐트(210)는 의미 인덱스(250)에 저장할 정보를 위하여 분석될 수 있다. 검색은 원하는 정보에 대하여 의미 인덱스(250)에 쿼리하는 것을 포함할 수 있다.

[0042] 콘텐트 세그먼트(310)는 텍스트 콘텐트(210)를 구성하는 문서들에 대해 수행될 수 있다. 문서들은 보다 효율적이고 잠재적으로 보다 정확한 동일 지시어 분석(320)을 위하여 세그먼트화될 수 있다. 동일 지시어 분석(320)은 전체 문서에 걸쳐서 잠재적인 지시 관계들(reference relationships)을 고려할 수 있다. 긴 문서들의 경우, 멀리 떨어진 표현들을 비교하는 데 많은 시간이 소비될 수 있다. 처리의 속도가 고려될 때, 동일 지시어 분석(320) 전에 문서들의 콘텐트 세그먼트화(310)는 처리에 사용되는 시간을 실질적으로 감소시킬 수 있다. 콘텐트 세그먼트화(310)는 동일 지시어 분석(320)의 시도들에서 탐구되는 콘텐트 텍스트(210)의 양을 효과적으로 감소시킬 수 있다.

[0043] 콘텐트 세그먼트화(310)는 새로운 문서 세그먼트화가 시작되는 때를 나타내는 정보를 의미적 동일 지시어 분석(370)에 제공할 수 있다. 그러한 정보는 세그먼트화 신호(312)로서 또는 콘텐트 문서 세그먼트에 마크업(mark-up)을 삽입함으로써 제공될 수 있다. 메타 정보를 포함하는 외부 파일 또는 다른 메커니즘들이 또한 이용될 수 있다.

[0044] 문서의 구조는 지시 관계들이 교차할 것 같지 않은 세그먼트 경계들을 식별하는 데 이용될 수 있다. 문서 구조는 단락 경계들, 챕터들, 또는 섹션 표제들과 같은 명백한 마크업으로부터 추론될 수 있다. 문서 구조는 또한 언어 처리를 통하여 발견될 수 있다. 지정된 길이를 초과하는 세그먼트들은 더욱 서브분할될 수 있다. 원하는 서브분할 길이는, 예를 들면, 문장들의 수 또는 단어들의 수에 의하여 표현될 수 있다.

[0045] 확실한 문서 구조화가 이용 가능하지 않은 경우, 휴리스틱(heuristic) 또는 통계적 기준들이 적용될 수 있다. 그러한 기준들은 세그먼트의 사이즈를 소정의 최대값으로 제한하면서 동일 지시어들을 함께 유지하는 경향을 갖도록 지정될 수 있다. 텍스트 콘텐트(210) 문서들을 세그먼트화하기 위한 다양한 다른 접근법들이 또한 적용될 수 있다. 콘텐트 세그먼트화(310)는 또한 전체 문서를 하나의 세그먼트로서 지정할 수도 있다.

[0046] 동일 지시어 분석(320, 370)은 콘텐트 텍스트(210) 내의 동일 지시어 및 별칭들을 식별하는 데 이용될 수 있다. 예를 들면, 문장 "He painted Guernica"를 인덱싱할 때, 그것은 "he"가 Picasso를 지시한다는 것을 판정하는 데에 판정적일 수 있다. 이것은 특히 사실 기반 검색이 이용되고 있는 경우에 그러하다. Picasso에 대한 대명사 별칭을 분석하는 것은 어떤 남성 개인인 "he"가 Guernica를 그렸다는 덜 유익할 사실보다는, Picasso가 Guernica를 그렸다는 사실을 인덱싱하는 것을 지원할 수 있다. 대명사의 지시 대상을 식별하고 인덱싱하는 이러한 능력이 없다면, 사실 기반 검색 방법을 이용하여, 쿼리 "Picasso painted"에 응답하여 문서를 검색하는 것은 어려울 수 있다. 시스템의 리콜(recall)은 다른 경우라면 반환되지 않았을 수 있는 쿼리에 관련된 문서가 반환될 때 개선될 수 있다.

[0047] 주석(annotation)(330)은 엔티티들 및 가능한 동일 지시 관계들을 추적하는 것을 지원하기 위해 텍스트 콘텐트(210)에 적용될 수 있다. 분석 판정의 신뢰 값들이 또한 텍스트 콘텐트(210) 내에 주석되거나 마크업될 수 있다. 분석 판정들은 텍스트에 명백한 주석 마크들을 추가하는 것에 의해 기록될 수 있다. 예를 들면 텍스트, "John visited Mary. He met her in 2003."가 주어진다. 주석(330)은 "[E1:0.9 John] visited [E2:0.8 Mary]. [E1:0.9 He] met [E2:0.8 her] in 2003."으로서 적용될 수 있다. 여기서 단어들 "John" 및 "He"는 0.9의 신뢰 값을 갖는 엔티티 1(E1)로서 관련될 수 있다. 마찬가지로, 단어들 "Mary" 및 "her"는 0.8의 신뢰 값을 갖는 엔티티 2(E2)로서 관련될 수 있다. 신뢰 값은 동일 지시어 분석(320) 판정에서의 신뢰의 측정값을 나타낼 수 있다. 주석은 동일 지시어 판정들을 직접 인코딩할 수 있거나, 또는 주석은 주석이 달린 텍스트 내의 관련 용어들을 스탠드 어사이드(stand aside) 주석(325) 내의 추가 정보에 연결하는 식별자들로서 기능할 수 있다.

[0048] 동일 지시어 분석(320) 판정들은 의미 매핑(225)을 구성하는 프로세스의 일부로서 이용될 수 있다. 동일 지시어 분석(320)에 의해 이용되는 참조 표현들은 텍스트 콘텐트(210) 내의 인라인 주석들에 의해 의미 매핑(225)을

위한 입력 표현에 통합될 수 있다. 그 지시들은 또한 외부 스탠드-어사이드 엔티티 맵(325)에서 별도로 제공될 수도 있다.

[0049] 월드 와이드 웹과 같은, 텍스트 콘텐트(210)의 많은 양의 문서 컬렉션 내에서, 동일한 문장이 상이한 문맥들에서 다수 회 나타날 수 있다. 이들 상이한 문맥들은 동일 지시어 분석(320)을 위한 상이한 후보들을 제공할 수 있다. 구문 파싱(215)은 계산상 비용이 많이 들 수 있으므로, 문장들에 대한 파싱 결과들을 캐시에 저장하는 것이 유익할 수 있다. 그러한 캐싱 메커니즘(350)은 문장이 향후에 직면하는 경우에 파싱 정보를 검색하는 것을 신속히 지원할 수 있다.

[0050] 동일 지시어 분석(320)이 상이한 문맥들에서 나타나는 단일 문장에 적용된다면, 그것은 동일 지시어가 문맥에 의존할 수 있으므로 동일한 참조 표현들에 대하여 상이한 동일 지시 관계들을 식별할 수 있다. 따라서, 상이한 엔티티 식별자들이 텍스트에 인라인으로 삽입될 수 있다. 예를 들면, 2개의 상이한 문서들에서 나타나는 텍스트 "He is smart"는 2개의 상이한 식별자들, "[E21 He] is smart." 및 "[E78 He] is smart."로 주석을 달 수 있다. 여기서 제1 문서 내의 단어 "He"는 제2 문서 내의 단어 "He"와 다른 사람을 지칭한다.

[0051] 얕은(shallow) 동일 지시어 분석(320)을 위한 상이한 정보 소스들이 있을 수 있다. 예를 들면, 동일 지시어 분석(320) 동안에 수행되는 표현 검출 외에도, 텍스트 콘텐트(210)에서 적당한 이름들을 찾아내는 데에 전용되는 시스템이 있을 수 있다. 이들 상이한 소스들은 상충되는 분석 정보를 식별할 수 있다. 예를 들면, 경계들이 교차하는 곳에서 상충되는 분석이 일어날 수 있다. 예를 들면, 2개의 시스템들이 다음의 상충되는 참조 표현들을 식별하였을 수 있다:

[0052] "[John] told [George Washington][Irving] was a great writer."

[0053] "[John] told [George] [Washington Irving] was a great writer."

[0054] 교차되는 경계들의 다음의 상충들을 생각해보자: 제1 문자열 내의 [George Washington]은 제2 문자열 내의 [George]와 상충된다. 또한 제1 문자열 내의 [George Washington]은 제2 문자열 내의 [Washington Irving]과 상충된다. 신뢰 정보 또는 문맥상의 요소들에 기초하여, 이러한 상충을 분석하고 그것을 보존하기 위해 상이한 전략들이 반복하여 적용될 수 있다. "드롭(drop)" 전략에서는, 2개 이상의 상충되는 경계들은 가장 낮은 신뢰를 갖는 것을 드롭함으로써 해결될 수 있다. "병합(merge)" 전략에서는, 양립할 수 있는 문맥들에서 2개 이상의 경계들이 동등하게 그럴듯한 경우에 그 경계들은 그에 따라서 이동될 수 있다. 예를 들면, "[Mr. John] Smith" 및 "Mr. [John Smith]"는 "[Mr. John Smith]"를 제공하도록 병합될 수 있다. "보존(preserve)" 전략에서는, 다수의 경계들은 그 경계들의 구성 및 그들의 신뢰 값들이 병합도 드롭도 지원하지 않는 경우에 그것들을 모호한 출력으로서 유지함으로써 보존될 수 있다. 예를 들면, "[Alexander the Great]" 및 "[Alexander][the Great]"는 대안적인 모호한 분석들로서 제공될 수 있다.

[0055] 파싱 컴포넌트(215)는 구문 파스(syntactic parse)(355)가 모호성을 보존할 수 있는 모호한 입력의 직접 파싱을 지원하는 모호성 인식 파서일 수 있다. 대안적으로, 모호한 입력 분석들이 분리되어 파싱될 필요가 있을 수 있고, 다수의 출력 구조들이 분리되어 의미 컴포넌트(225)에 전달될 수 있다. 의미 처리(225)는, 아래에서 더 상세히 논의되는 바와 같이, 구문 파서(215)의 각 출력에 다수 회 적용될 수 있다. 이에 따라 상이한 구문 입력들에 대하여 상이한 의미 출력들이 생성될 수 있다. 대안적으로, 의미 매핑(225)은 다양한 입력들을 조합하고 그것들을 일제히 처리할 수 있다.

[0056] 의미 매핑(225)은 의미 표준화(semantic normalization)(360)와 함께 존재할 수 있다. 문장의 다수의 모호한 구문 파스(355) 출력들은 상이한 형태들을 가지면서 의미를 공유할 수 있다. 예를 들면, 이것은 수동적 언어의 표준화에서 일어날 수 있다. "John gave Mary a present"를 고려하면, 단어 "John"은 주어이고 "Mary"는 간접 목적어이다. "a present was given to Mary by John"을 고려하면, 주어는 "Mary"이고 "John"은 목적어이다. 표준화(360)는 이들 2개의 예들이 "John"은 의미 주어(semantic-subject)이고 "Mary"는 의미 간접 목적어(semantic-indirect-object)인 것과 동일한 것을 나타내는 출력들을 제공할 수 있다. 대안적으로, "John"은 동작 에이전트로서 식별될 수 있고, "Mary"는 수령인으로서 식별될 수 있다. 마찬가지로, "Rome's destruction of Carthage" 및 "Rome destroyed Carthage"에 대하여 동일한 표현들이 제공될 수 있다.

[0057] 의미 표준화는 또한 파싱된 문장의 상이한 단어들에 관한 정보를 추가할 수 있다. 예를 들면, 단어들은 어휘 목록(lexicon)에서 식별되고 그들의 동의어들, 상위어들, 가능한 별칭들, 및 다른 어휘 정보와 관련될 수 있다.

[0058] 의미 기반 동일 지시어 분석(370)은 구문 및 의미 정보에 기초하여 표현들을 분석할 수 있다. 예를 들면, "John saw Bill. He greeted him."은 "he"를 "John"으로 "him"을 "Bill"로 분석할 수 있다. 이러한 분석은

"he"와 "John"은 둘 다 주어들이고, "him"과 "Bill"은 둘 다 목적어들이기 때문에 지정될 수 있다.

[0059] 얕은 동일 지시어 분석(320)은 용어들이 나타나는 문서 세그먼트를 면밀히 조사함으로써 기능할 수 있다. 이와 대조적으로, 의미적 동일 지시어 분석(370), 또는 깊은 동일 지시어 분석은 한 번에 하나의 문장을 처리할 수 있다. 나중의 문장들의 의미적 동일 지시어 분석(370)이 더 일찍이 도입된 엘리먼트들에 액세스할 수 있도록 문장들의 가능한 선행사들이 선행사 저장소(antecedent store)(375) 내에 배치될 수 있다. 선행사들은 문장에서의 그들의 문법적 기능 및 역할들, 텍스트에서의 그들의 거리에 관한 정보, 그들의 다른 선행사들과의 관계들에 관한 정보, 및 다양한 다른 정보들과 함께 저장될 수 있다.

[0060] 표현 병합(expression merging)(380)은 얕은 동일 지시어 분석(320)으로부터의 표현들, 스탠드 어사이드 주석(325), 및 의미적 동일 지시어 분석(370)으로부터의 정보를 조합할 수 있다. 조합될 용어들에 대한 정보는 문자열 정렬 또는 주석들(330)을 이용하여 식별될 수 있다. 동일한 텍스트에 대한 2개의 주석들을 조합하기 위한 다른 메커니즘들이 이용될 수도 있다.

[0061] 구문 파싱(215)은 옵션으로 검출된 참조 표현들에 대한 자연스러운 통합점(point of integration)일 수 있다. 파서는 구성 요소들과 같은 문장들 내의 구조, 또는 주어 및 목적어와 같은 문법적 관계들을 추론하는 것을 지원할 수 있다. 모호성 인에이블 구문 파서(215)는 문장의 다수의 대안적인 구조적 표현들을 식별할 수 있다. 일례로, 동일 지시어 분석(320)으로부터의 정보는 각 참조 표현의 좌측 경계가 파스로부터 양립할 수 있는 부분의 처음과 일치하는 표현들만을 계속 유지함으로써 구문 파서(215)의 출력을 필터링하는 데 이용될 수 있다. 예를 들면, 동일 지시어 분석은 "[E0 John] told [E1 George][E2 Washington Irving] was a great writer."에서와 같이 동일 지시 대상들을 확립할 수 있다. 구문 파서(215)는 4개의 파싱 가능성들을 별도로 제공할 수 있다:

[0062] 1. [John] and [George] and [Washington Irving]

[0063] 2. [John] and [George] and [Washington] and [Irving]

[0064] 3. [John] and [George Washington] and [Irving]

[0065] 4. [John] and [George Washington Irving]

[0066] 파서 가능성 번호 3 및 4는 지시 분석(320)에 의해 제공된 엔티티 E2 "Washington Irving"의 좌측 경계와 양립할 수 없기 때문에 필터링될 수 있다.

[0067] 확장(385)의 프로세스는 표현에 추가적인 정보를 추가할 수 있다. 예를 들면, "John sold a car from Bill"에 대하여, 확장(385)은 "Bill bought a car from John"에 대한 표현을 추가로 출력할 수 있다. 마찬가지로, "John killed Bill"에 대하여, 확장(385)은 "Bill died"에 대한 표현을 추가로 출력할 수 있다.

[0068] 전통적인 검색 엔진들은 매칭하는 키워드들 또는 용어들에 기초하여 사용자 쿼리들에 응답하여 문서들을 검색할 수 있다. 문서들은, 이들 전통적인 시스템들에서, 쿼리들로부터의 용어들 중 얼마나 많은 것이 문서들 내에서 나타나는지, 그 용어들이 얼마나 자주 나타나는지, 또는 그 용어들이 얼마나 가까이 함께 나타나는지와 같은 요소들에 따라서, 랭킹될 수 있다.

[0069] "Picasso was born in Malaga. He painted Guernica."를 포함하는 제2 예시 문서와 함께 "Picasso's friend Matisse painted prolifically."를 포함하는 제1 예시 문서에서 예시 쿼리 "Picasso painted"를 고려해 보자. 그 밖에 모든 것이 동일한 경우, 단어들 "Picasso" 및 "painted"는 제2 문서에서 더 가까이 함께 있기 때문에, 전통적인 시스템은 제2 문서를 제1 문서보다 더 상위에 랭킹시킬 수 있다. . 이와 대조적으로, 제1 문서 내의 단어 "He"가 Picasso를 지시한다는 것을 분석할 수 있는 시스템은 이 지식에 기초하여 정확하게 제1 문서를 더 상위에 랭킹할 수 있다. 쿼리 "Picasso painted"가, Picasso가 무엇을 그렸는지를 알아내려는 사용자의 의도를 반영한다고 가정할 때, 제1 문서는 명백히 보다 관련된 결과이다.

[0070] 자연 언어 처리 시스템(300)은 상이한 아키텍처들을 가질 수 있다. 일 실시예에서, 언어 처리의 하나의 스테이지로부터의 정보가 나중의 스테이지들에의 입력으로서 전달되는 파이프라인이 제공될 수 있다. 이들 접근법들은 자연 언어 텍스트 콘텐트(210)로부터 인덱싱될 사실들을 추출하도록 동작할 수 있는 임의의 다른 아키텍처로 구현될 수도 있다는 이해해야 한다.

[0071] 이제 도 4를 참조하여, 모호성 민감 자연 언어 처리 시스템에서의 동일 지시어 분석을 위해 여기에 제시된 실시예들에 관한 추가적인 상세들이 제공될 것이다. 특히, 도 4는 여기에 제시된 실시예의 양태들에 따른 동일 지

시어 분석에 의한 모호성 민감 인덱싱을 위한 프로세스들(400)의 양태들을 예시하는 흐름도이다.

[0072] 여기에 설명된 논리 동작들은 (1) 컴퓨팅 시스템에서 실행하는 프로그램 모듈들 또는 컴퓨터 구현 행위의 시퀀스로서 및/또는 (2) 컴퓨팅 시스템 내의 상호 접속된 기계 논리 회로들 또는 회로 모듈들로서 구현된다는 것을 이해해야 한다. 본 구현은 컴퓨팅 시스템의 성능 및 기타 요건들에 의존하는 선택의 문제이다. 따라서, 여기에 설명된 논리 동작들은 상태 동작들, 구조 장치들, 단계들, 또는 모듈들로서 다양하게 언급된다. 이들 동작들, 구조 장치들, 행위들 및 모듈들은 소프트웨어로, 하드웨어로, 특수 용도 디지털 로직, 및 이들의 임의의 조합으로 구현될 수 있다. 또한 도면들에서 도시되고 본 명세서에 설명된 것보다 더 많은 또는 더 적은 수의 동작들이 수행될 수도 있다는 것을 이해해야 한다. 이들 동작들은 또한 순차적으로, 병행하여, 또는 여기에 설명된 것들과는 다른 순서로 수행될 수도 있다.

[0073] 루틴(400)은 동작 410에서 시작하여, 분석 및 인덱싱을 위해 텍스트 콘텐트(210)의 부분이 검색될 수 있다. 동작 420에서 텍스트 콘텐트(210)는 분석 처리가 많이 검색하고 분석하는 텍스트의 영역들의 경계를 위해 세그먼트화된다. 이 세그먼트화는 문장들, 단락들, 페이지들, 챕터들, 또는 섹션들과 같은, 텍스트 내의 구조에 기초할 수 있다. 세그먼트화는 또한 단어들의 수, 문장들의 수, 또는 공간 또는 복잡성의 다른 메트릭들에 기초할 수 있다.

[0074] 동작 430에서는 텍스트 콘텐트(210) 내에서 동일 지시어들이 분석될 수 있다. 동작 430 내에서 확립된 경계들과 협력하여, 동일 지시어들이 식별되고 매칭될 수 있다. 별칭 집단들이 확립될 수 있다. "얕은" 분석을 제공하기 위해 표면 구조가 이용될 수 있다. 동일 지시어 분석 동안에 발생하는 모호성들은 주석이 달릴 수 있다. 그러한 주석(340)은 텍스트 콘텐트(210) 내의 마크업으로서 또는 외부 엔티티 맵의 이용을 통하여 제공될 수 있다. 또한 지시들 및 지시 대상들을 엔티티 번호들로 라벨링하기 위해 유사한 주석이 이용될 수도 있다. 또한 확립된 동일 지시어 분석들의 신뢰 레벨을 표시하기 위해 주석이 제공될 수도 있다.

[0075] 동작 440에서는, 구문 파싱이 문장들을 단어들 사이의 구문 관계들을 명백하게 하는 표현들로 변환할 수 있다. 파서(215)는 구문 파스(355) 정보를 제공하기 위해 특정 언어와 관련된 문법(220)을 적용할 수 있다.

[0076] 동작 450에서는, 텍스트 콘텐트(210)로부터 의미 표현들이 추출될 수 있다. 텍스트 콘텐트(210) 내의 문서에서 표현된 정보는 텍스트 내의 엔티티들 사이의 관계들의 표현들에 의하여 형식적으로 조직될 수 있다. 이들 관계들은 일반적인 의미에서 사실로서 지칭될 수 있다.

[0077] 동작 455에서는, 구문 파스(215)로부터 출력된 구문 파스(355) 정보가 깊은 동일 지시어 분석(370)을 지원하는 데 이용될 수 있다. 또한 동작 450 동안에 생성된 의미 표현들이 이용될 수도 있다.

[0078] 동작 460에서는, 얕은 동일 지시어 분석 동작(430)으로부터의 표현들이 깊은 동일 지시어 분석 동작(455)으로부터의 정보와 통합될 수 있다. 모호성 인에이블 구문 파서(215)는 문장의 다수의 대안적인 구조 표현들을 식별할 수 있다. 동일 지시어 분석으로부터의 정보는 구문 파서(215)의 출력을 필터링하는 데 이용될 수 있다.

[0079] 동작 470에서는, 선택된 함축된 표현들을 포함하도록 텍스트 콘텐트(210)의 의미들이 확장될 수 있다. 동작 475에서는, 콘텐트 텍스트 내의 엔티티들, 이벤트들 및 사건들의 상태들 사이의 관계들을 표현하는 의미 표현들로부터 사실들이 추출될 수 있다. 동작 480에서는, 사실들 및 엔티티들이 의미 인덱스(250)에 저장될 수 있다.

[0080] 루틴(400)은 동작 480 후에 종료할 수 있다. 그러나, 루틴(400)은 의미 인덱스(250)에 적용될 텍스트 콘텐트(210) 부분들을 검색하기 위해 되풀이하여 또는 연속적으로 적용될 수 있다는 것을 이해해야 한다.

[0081] 이제 도 5를 참조하면, 예시적인 컴퓨터 아키텍처(500)는 모호성 민감 자연 언어 처리 시스템에서의 동일 지시어 분석을 위해 여기에 설명된 소프트웨어 컴포넌트들을 실행할 수 있다. 도 5에 도시된 컴퓨터 아키텍처는 종래의 데스크톱, 랩톱, 또는 서버 컴퓨터를 예시하고 여기에 제시된 소프트웨어 컴포넌트들의 임의의 양태들을 실행하는 데 이용될 수 있다. 그러나, 설명된 소프트웨어 컴포넌트들은 또한 이동 장치, 텔레비전, 셋톱 박스, 키오스크, 차량 정보 시스템, 이동 전화기, 내장 시스템, 또는 그 밖의 것들과 같은, 다른 예시적인 컴퓨팅 환경들에서 실행될 수도 있다는 것을 이해해야 한다. 클라이언트 컴퓨터들(110A-110D) 또는 서버 컴퓨터들(120) 중 임의의 하나 이상의 것들은 실시예들에 따른 컴퓨터 시스템(500)으로서 구현될 수 있다.

[0082] 도 5에 예시된 컴퓨터 아키텍처는 중앙 처리 장치(10)(CPU), 랜덤 액세스 메모리(14)(RAM) 및 읽기 전용 메모리(16)(ROM)을 포함하는 시스템 메모리(13), 및 시스템 메모리(13)를 CPU(10)에 연결할 수 있는 시스템 버스(11)를 포함할 수 있다. 기동 중인 경우 등에서, 컴퓨터(500) 내의 엘리먼트들 사이에 정보를 전송하는 데 도움이 되는 기본 루틴들을 포함하는 기본 입력/출력 시스템이 ROM(16)에 저장될 수 있다. 컴퓨터(500)는 운영 체제

(18), 소프트웨어, 데이터, 및 자연 언어 엔진(130)과 관련된 것들과 같은, 다양한 프로그램 모듈들을 저장하기 위한 대용량 저장 장치(15)를 더 포함할 수 있다. 자연 언어 엔진(130)은 여기에 설명된 소프트웨어 컴포넌트들의 부분들을 실행할 수 있다. 자연 언어 엔진(130)과 관련된 의미 인덱스(250)는 대용량 저장 장치(15) 내에 저장될 수 있다.

[0083] 대용량 저장 장치(15)는 버스(11)에 접속된 (도시되지 않은) 대용량 저장 컨트롤러를 통하여 CPU(10)에 접속될 수 있다. 대용량 저장 장치(15) 및 그와 관련된 컴퓨터 판독 가능한 매체는 컴퓨터(500)를 위한 비휘발성 저장을 제공할 수 있다. 비록 여기에 포함된 컴퓨터 판독 가능한 매체의 설명은 하드 디스크 또는 CD-ROM 드라이브와 같은 대용량 저장 장치를 참조하지만, 숙련된 당업자들은 컴퓨터 판독 가능한 매체가 컴퓨터(500)에 의해 액세스될 수 있는 임의의 이용 가능한 컴퓨터 저장 매체일 수 있다는 것을 알 것이다.

[0084] 제한이 아니라, 예로서, 컴퓨터 판독 가능한 매체는 컴퓨터 판독 가능한 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위해 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 이동불가식 매체를 포함할 수 있다. 예를 들면, 컴퓨터 판독 가능한 매체는, RAM, ROM, EPROM, EEPROM, 플래시 메모리 또는 기타 솔리드 스테이트 메모리 기술, CD-ROM, DVD(digital versatile disk), HD-DVD, BLU-RAY, 또는 기타 광 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터(500)에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 이용될 수 있는 임의의 기타 매체를 포함하지만 이에 제한되는 것은 아니다.

[0085] 다양한 실시예들에 따르면, 컴퓨터(500)는 네트워크(140)와 같은 네트워크를 통하여 원격 컴퓨터들로의 논리적 접속들을 이용하여 네트워크화된 환경에서 동작할 수 있다. 컴퓨터(500)는 버스(11)에 접속된 네트워크 인터페이스 유닛(19)을 통하여 네트워크(140)에 접속될 수 있다. 네트워크 인터페이스 유닛(19)은 또한 다른 유형의 네트워크들 및 원격 컴퓨터 시스템들에 접속하는 데 이용될 수도 있다는 것을 이해해야 한다. 컴퓨터(500)는 또한 (도시되지 않은) 키보드, 마우스, 또는 전자 스타일러스를 포함하는 다수의 다른 장치들로부터 입력을 수신하고 처리하기 위한 입력/출력 컨트롤러(12)를 포함할 수 있다. 마찬가지로, 입력/출력 컨트롤러(12)는 (또한 도시되지 않은) 비디오 디스플레이, 프린터, 또는 다른 유형의 출력 장치에 출력을 제공할 수 있다.

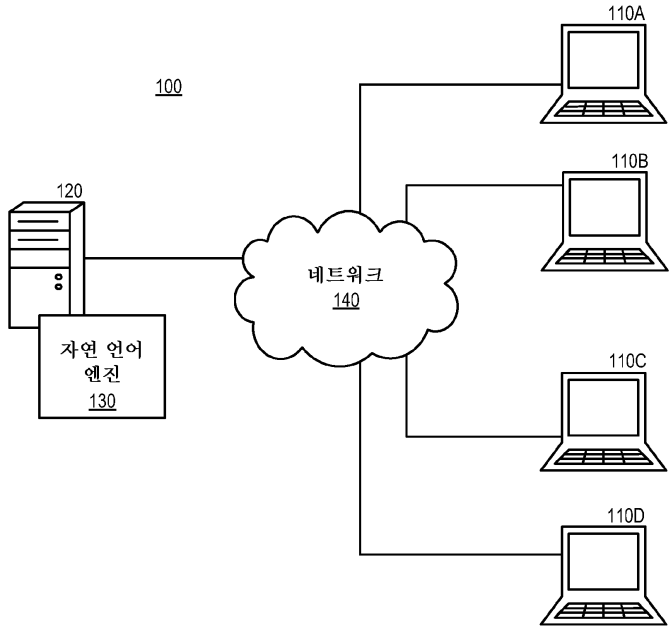
[0086] 위에서 간단히 언급한 바와 같이, 네트워크화된 데스크톱, 랩톱, 서버 컴퓨터, 또는 기타 컴퓨팅 환경의 동작을 제어하기에 적합한 운영 체제(18)를 포함하여, 다수의 프로그램 모듈들 및 데이터 파일들이 컴퓨터(500)의 대용량 저장 장치(15) 및 RAM(14)에 저장될 수 있다. 대용량 저장 장치(15), ROM(16), 및 RAM(14)은 또한 하나 이상의 프로그램 모듈들을 저장할 수 있다. 특히, 대용량 저장 장치(15), ROM(16), 및 RAM(14)은 CPU(10)에 의한 실행을 위한 자연 언어 엔진(130)을 저장할 수 있다. 자연 언어 엔진(130)은 도 2-4에 관련하여 상세히 논의된 프로세스들의 부분들을 구현하기 위한 소프트웨어 컴포넌트들을 포함할 수 있다. 대용량 저장 장치(15), ROM(16), 및 RAM(14)은 또한 다른 유형의 프로그램 모듈들을 저장할 수 있다. 대용량 저장 장치(15), ROM(16), 및 RAM(14)은 또한 자연 언어 엔진(130)과 관련된 의미 인덱스(250)를 저장할 수 있다.

[0087] 전술한 것에 기초하여, 여기서는 모호성 민감 자연 언어 처리 시스템에서의 동일 지시어 분석을 위한 기술들이 제공되었다는 것을 이해해야 한다. 비록 여기에 제시된 내용은 컴퓨터 구조 특징들, 방법적 행위들, 및 컴퓨터 판독 가능한 매체들에 특정한 언어로 설명되었지만, 첨부된 청구항들에서 정의된 발명은 반드시 여기에 설명된 특정한 특징들, 행위들, 또는 매체들에 제한되지는 않는다는 것을 이해해야 한다. 오히려, 그 특정한 특징들, 단계들 및 매체들은 청구항들을 구현하는 예시적인 형태들로서 개시되어 있다.

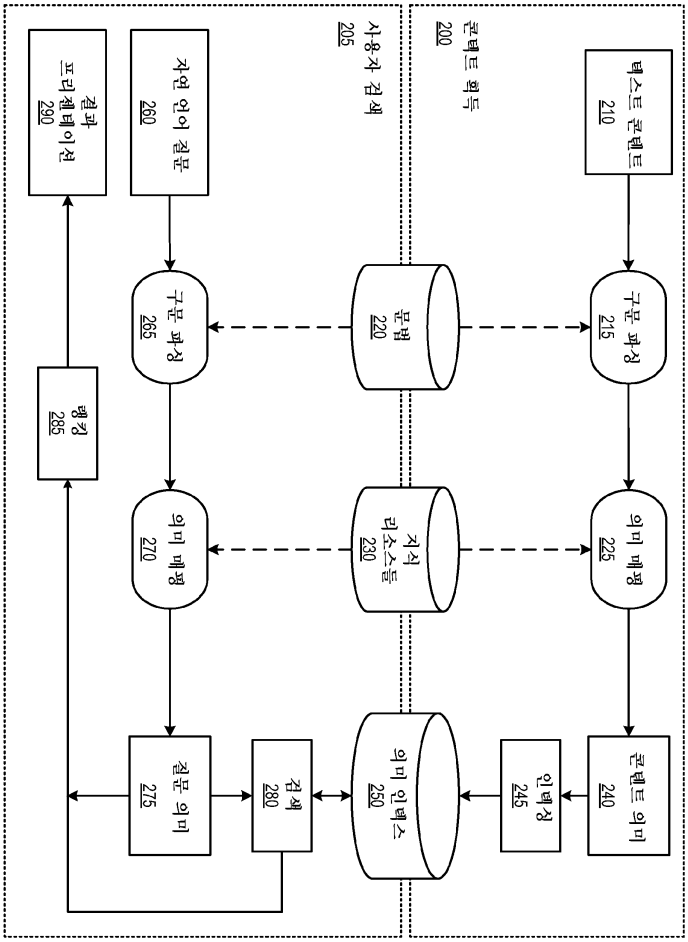
[0088] 전술한 내용은 단지 예시로서 제공되는 것일 뿐이고 제한적인 것으로 해석되지 않아야 한다. 예시되고 설명된 예시적인 실시예들 및 응용들을 따르지 않고, 또한 다음의 청구항들에서 제시되는, 본 발명의 참된 정신 및 범위에서 벗어나지 않고 여기에 설명된 내용에 다양한 수정들 및 변경들이 행해질 수 있다.

도면

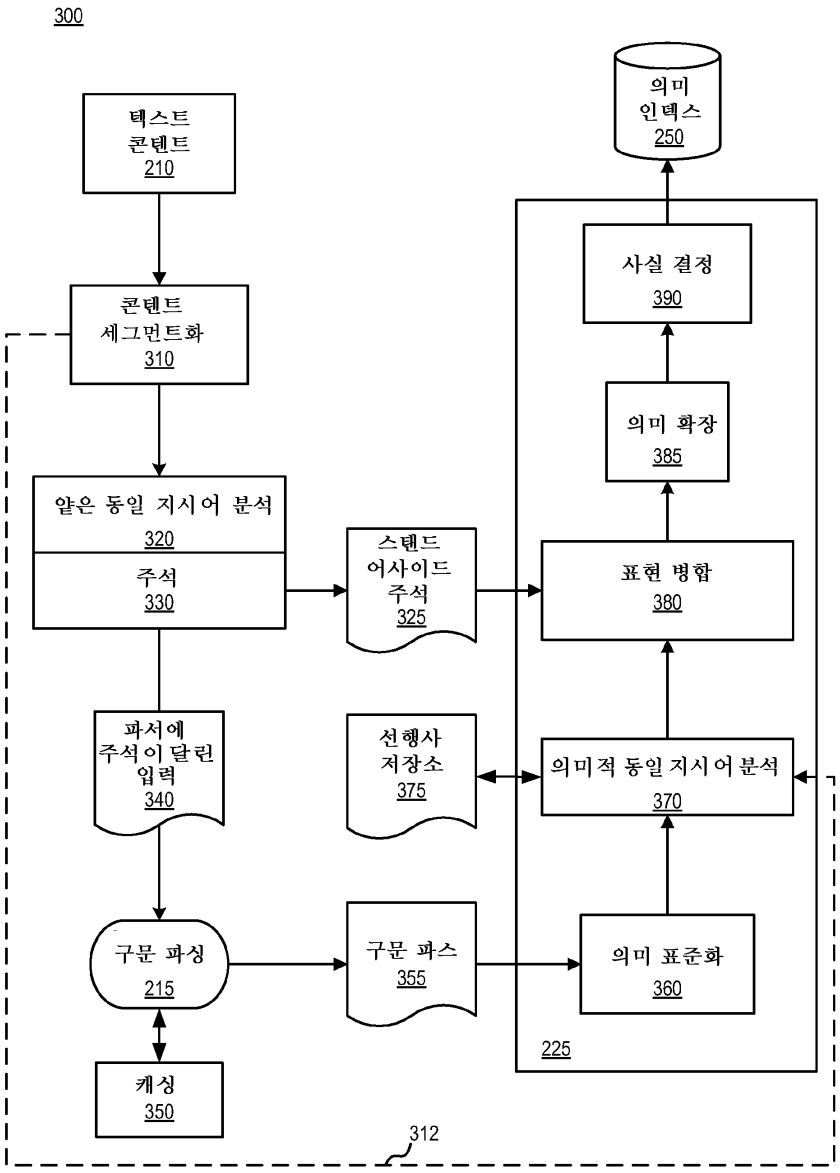
도면1



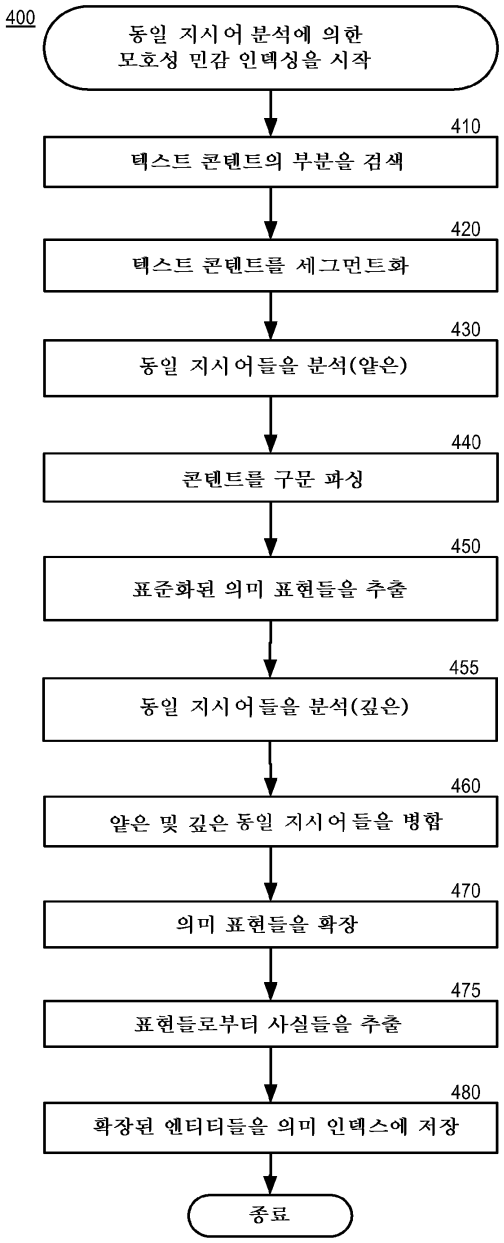
도면2


콘텍트 획득
200
사용자 검색
205
텍스트 콘텍트
210
구문 파싱
215
문법
220
지식
리소스들
230
의미 매핑
225
콘텍트 의미
240
인덱싱
245
의미 인덱스
250
검색
280
질문 의미
275
구문 파싱
265
의미 매핑
270
랭킹
285
결과
프리젠테이션
290
자연 언어 질문
260

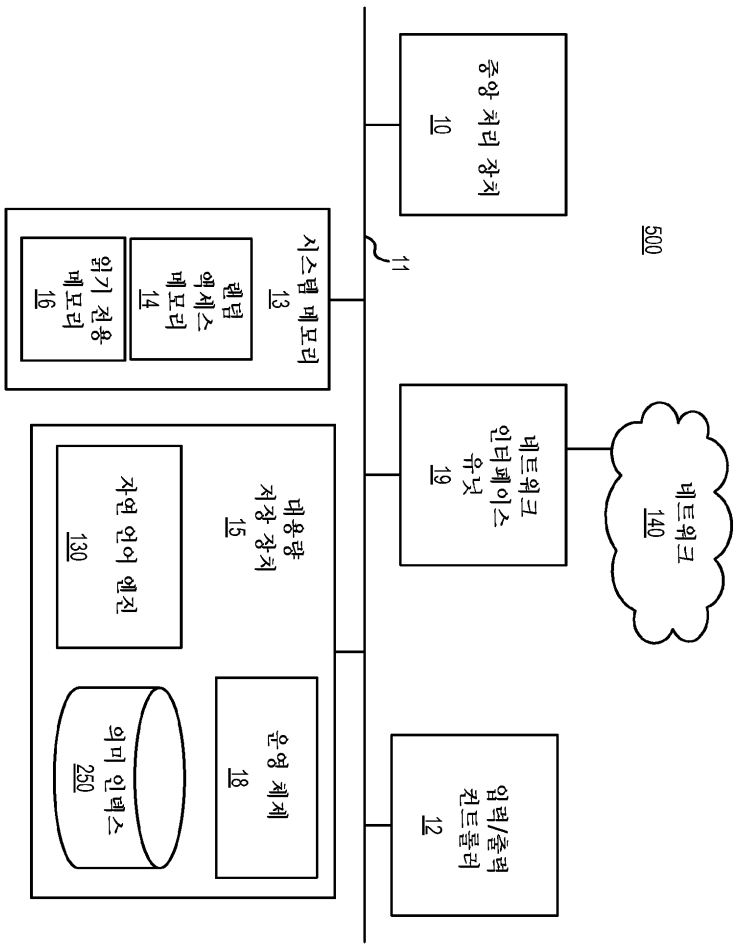
도면3


300
텍스트 콘텐트 210
콘텐트 세그먼트화 310
얕은 동일 지시어 분석 320
주석 330
파서에 주석이 달린 입력 340
구문 파싱 215
캐싱 350
스텐드 어사이드 주석 325
선행사 저장소 375
구문 파스 355
의미 인덱스 250
사실 결정 390
의미 확장 385
표현 병합 380
의미적 동일 지시어 분석 370
의미 표준화 360
225
312

도면4


400
동일 지시어 분석에 의한 모호성 민감 인덱싱을 시작
410
텍스트 콘텐트의 부분을 검색
420
텍스트 콘텐트를 세그먼트화
430
동일 지시어들을 분석(얕은)
440
콘텐트를 구문 파싱
450
표준화된 의미 표현들을 추출
455
동일 지시어들을 분석(깊은)
460
얕은 및 깊은 동일 지시어들을 병합
470
의미 표현들을 확장
475
표현들로부터 사실들을 추출
480
확장된 엔티티들을 의미 인덱스에 저장
종료

도면5


중앙 처리 장치
10
500
11
시스템 메모리
13
랜덤
액세스
메모리
14
읽기 전용
메모리
16
네트워크
인터페이스
유닛
19
네트워크
140
대용량
저장 장치
15
자연 언어 엔진
130
의미 인덱스
250
운영 체제
18
입력/출력
컨트롤러
12