(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0131893
(43) 공개일자 2017년12월01일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) *G06F 17/22* (2006.01)
(52) CPC특허분류
G06F 17/30598 (2013.01)
G06F 17/2247 (2013.01)
(21) 출원번호 10-2016-0062645
(22) 출원일자 2016년05월23일
심사청구일자 2016년05월23일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
이상근
서울특별시 강남구 선릉로 221, 205동 1104호 (도곡동, 도곡렉슬아파트)
신해용
서울특별시 강남구 도곡로43길 21, 101동 1202호 (역삼동, 래미안그레이튼)
(74) 대리인
특허법인엠에이피에스

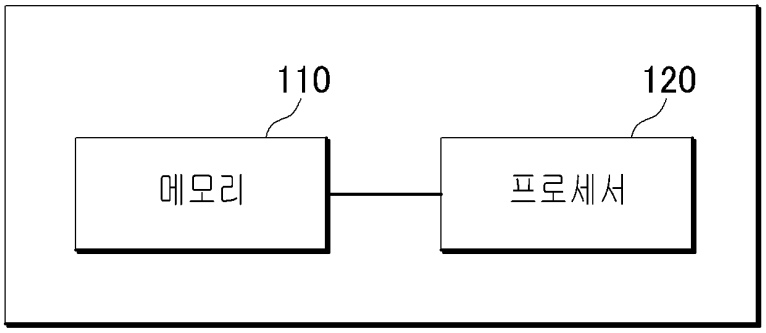
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **오픈 디렉터리 프로젝트 기반의 텍스트 분류기, 및 텍스트 분류기의 생성 및 분류 방법**

(57) 요 약

본 발명은 텍스트 분류 프로그램이 저장된 메모리 및 프로그램을 실행하는 프로세서를 포함한다. 이때, 프로세서는 프로그램의 실행에 따라, 분류 대상 텍스트로부터 단어를 추출하고, 추출된 단어로부터 분류 대상 텍스트의 어구를 추출하며, 추출된 단어, 추출된 어구 및 텍스트 분류 모델에 기초하여, 분류 대상 텍스트에 대응하는 카테고리를 분류한다. 그리고 텍스트 분류 모델은 오픈 디렉터리 프로젝트에 기초하여 생성된다.

대 표 도 - 도2



100

(52) CPC특허분류
G06F 17/30327 (2013.01)
G06F 17/30734 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711030533
부처명 미래창조과학부
연구관리전문기관 한국연구재단
연구사업명 중견연구자지원사업(도약연구지원사업)
연구과제명 초소형 텍스트지능을 이용한 내장형 개인화 서비스 기술
기 여 율 1/1
주관기관 고려대학교
연구기간 2015.11.01 ~ 2016.10.31

명 세 서

청구범위

청구항 1

오픈 디렉터리 프로젝트 기반의 텍스트 분류기에 있어서,

텍스트 분류 프로그램이 저장된 메모리 및

상기 프로그램을 실행하는 프로세서를 포함하고,

상기 프로세서는 상기 프로그램의 실행에 따라, 분류 대상 텍스트로부터 단어를 추출하고, 상기 추출된 단어로부터 상기 분류 대상 텍스트의 어구를 추출하며,

상기 추출된 단어, 상기 추출된 어구 및 텍스트 분류 모델에 기초하여, 상기 분류 대상 텍스트에 대응하는 카테고리를 분류하되,

상기 텍스트 분류 모델은 오픈 디렉터리 프로젝트에 기초하여 생성된 것인, 텍스트 분류기.

청구항 2

제 1 항에 있어서,

상기 프로세서는 상기 오픈 디렉터리 프로젝트의 각 카테고리로부터 단어 및 어구를 추출하고, 상기 카테고리에 대응하는 단어 및 어구 빈도 벡터를 생성하며,

상기 카테고리에 대응하는 단어 및 어구 빈도 벡터에 기초하여, 상기 텍스트 분류 모델을 생성하되,

상기 각 카테고리에 대응하는 단어 및 어구는 상기 각 카테고리에 저장된 문서 집합으로부터 추출되는 것인, 텍스트 분류기.

청구항 3

제 2 항에 있어서,

상기 프로세서는 상기 각 카테고리에 저장된 문서 집합으로부터 구문분석에 기초하여, 후보 어구 집합을 추출하고, 상기 후보 어구 집합에 포함된 후보 어구에 대한 휴리스틱 규칙에 기초하여, 상기 각 카테고리에 대응하는 어구를 추출하는 것인, 텍스트 분류기.

청구항 4

제 3 항에 있어서,

상기 휴리스틱 규칙은,

상기 후보 어구에 포함된 복수의 단어 중 스탑워드(stopword) 및 특정 품사의 단어를 제거하는 규칙, 상기 후보 어구에 포함된 단어에 대한 스태밍(stemming)을 적용하는 규칙 및 상기 후보 어구가 미리 정해진 개수의 단어 이상의 서로 상이한 단어를 포함하는 규칙 중 어느 하나 이상이되,

상기 특정 품사는 한정사, 전치사, 소유격 어미 및 종속 접속사를 포함하는, 텍스트 분류기.

청구항 5

제 3 항에 있어서,

상기 프로세서는 상기 각 카테고리에 대응하여 추출된 어구에 대한 점수를 산출하고, 상기 산출된 점수에 기초하여, 미리 정해진 개수의 어구를 선택하고,

상기 선택된 어구에 대하여, 상기 카테고리에 대응하는 단어 및 어구 빈도 벡터를 생성하는, 텍스트 분류기.

청구항 6

제 5 항에 있어서,

상기 프로세서는 카이-점수(chi-score)에 기초하여, 상기 어구에 대한 점수를 산출하는 것인, 텍스트 분류기.

청구항 7

제 6 항에 있어서,

상기 프로세서는 하기 수학식을 이용하여, 상기 어구에 대한 각 카테고리의 점수를 산출하고 상기 산출된 각 카테고리의 점수 중 최대값을 상기 어구에 대한 점수로 산출하는, 텍스트 분류기.

[수학식]

$$\chi^2(p, c_i) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

p: 상기 점수를 산출하는 어구

c_i: 오픈 디렉터리 프로젝트의 i 번째 카테고리

A: 상기 i 번째 카테고리에 포함되며, 상기 점수를 산출하는 어구를 포함하는 문서의 개수

B: 상기 i 번째 카테고리에 포함되지 않으며, 상기 점수를 산출하는 어구를 포함하는 문서의 개수

C: 상기 i 번째 카테고리에 포함되며, 상기 점수를 산출하는 어구를 포함하지 않는 문서의 개수

D: 상기 i 번째 카테고리에 포함되지 않으며, 상기 점수를 산출하는 어구를 포함하지 않는 문서의 개수

N: 오픈 디렉터리 프로젝트에 포함된 문서 개수의 총합

청구항 8

오픈 디렉터리 프로젝트 기반의 텍스트 분류기의 텍스트 분류 방법에 있어서,

분류 대상 텍스트로부터 단어를 추출하는 단계;

상기 추출된 단어로부터 상기 분류 대상 텍스트에 대응하는 어구를 추출하는 단계; 및

상기 추출된 단어, 상기 추출된 어구 및 텍스트 분류 모델에 기초하여, 상기 분류 대상 텍스트에 대응하는 카테고리를 분류하는 단계를 포함하되,

상기 텍스트 분류 모델은 오픈 디렉터리 프로젝트에 기초하여 생성된 것인, 텍스트 분류기의 텍스트 분류 방법.

청구항 9

오픈 디렉터리 프로젝트 기반의 텍스트 분류기의 텍스트 분류 모델 생성 방법에 있어서,

오픈 디렉터리 프로젝트의 각 카테고리로부터 단어를 추출하는 단계;

상기 추출된 단어에 기초하여, 상기 각 카테고리로부터 어구를 추출하는 단계;

상기 추출된 단어 및 상기 추출된 어구에 기초하여, 상기 각 카테고리에 대응하는 단어 및 어구 빈도 벡터를 생성하는 단계; 및

상기 각 카테고리에 대응하는 단어 및 어구 빈도 벡터에 기초하여, 상기 텍스트 분류 모델을 생성하는 단계를 포함하되,

상기 각 카테고리에 대응하는 단어 및 어구는 상기 각 카테고리에 저장된 문서 집합으로부터 추출되는 것인, 텍스트 분류기의 텍스트 분류 모델 생성 방법.

청구항 10

제 9 항에 있어서,

상기 추출된 단어에 기초하여, 상기 각 카테고리로부터 어구를 추출하는 단계는,

상기 각 카테고리에 저장된 문서 집합으로부터 구문분석에 기초하여, 후보 어구 집합을 추출하고, 상기 후보 어구 집합에 포함된 후보 어구에 대한 휴리스틱 규칙에 기초하여, 상기 각 카테고리에 대응하는 어구를 추출하는 것인, 텍스트 분류 모델 생성 방법.

청구항 11

제 10 항에 있어서,

상기 휴리스틱 규칙은, 상기 후보 어구에 포함된 복수의 단어 중 스탑워드(stopword) 및 특정 품사의 단어를 제거하는 규칙, 상기 후보 어구에 포함된 단어에 대한 스태밍(stemming)을 적용하는 규칙 및 상기 후보 어구가 미리 정해진 개수의 단어 이상의 서로 상이한 단어를 포함하는 규칙 중 어느 하나 이상이고,

상기 특정 품사는 한정사, 전치사, 소유격 어미 및 종속 접속사를 포함하는, 텍스트 분류 모델 생성 방법.

청구항 12

제 8 항 내지 제 11항 중 어느 한 항에 기재된 방법을 컴퓨터 상에서 수행하기 위한 프로그램을 기록한 컴퓨터 판독 가능한 기록 매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 오픈 디렉터리 프로젝트 기반의 텍스트 분류기, 및 텍스트 분류기의 생성 및 분류 방법에 관한 것이다.

배 경 기 술

[0002] 오픈 디렉터리 프로젝트(open directory project)는 일종의 웹 상의 텍스트 데이터베이스이다. 오픈 디렉터리 프로젝트는 웹 상의 390만 개 이상의 웹 문서 등의 텍스트를 계층적 온톨로지(hierarchical ontology)를 이용하여 카테고리(category)로 세분화하고, 트리(tree) 자료 구조(data structure)를 이용하여 분류한 것이다.

[0003] 오픈 디렉터리 프로젝트의 카테고리에 기반하여 신규 텍스트의 카테고리를 분류하기 위한 종래기술은 나이브 베이즈(Naive Bays), k-근접 이웃 방법(k-Nearest Neighbor), 서포트 벡터 머신(Support Vector Machine) 등 기계 학습 알고리즘(machine learning algorithm) 기반 기존 텍스트 분류(text classification) 기법을 활용하는 것이다. 그러나 이러한 종래 기술을 이용하는 경우, 오픈 디렉터리 프로젝트의 카테고리가 증가함에 따라, 텍스트 분류기를 생성할 때 목표(target)가 되는 카테고리에 포함된 학습 데이터가 부족하고, 이 학습 데이터를 이용하여 학습된 텍스트 분류기의 분류 정확도가 낮아진다는 문제점이 있다.

[0004] 이와 관련되어, 한국 등록특허공보 제10-1580152호(발명의 명칭: "조상 후손 카테고리를 활용한 오픈 디렉터리 프로젝트 기반 텍스트 분류 방법 및 장치")는 오픈 디렉토리 프로젝트에 포함된 오픈 디렉터리 프로젝트에 포함된 부모 및 자손 데이터뿐만 아니라 조상 및 후손 데이터를 이용하여 신규 텍스트의 카테고리를 분류하는 방법 및 장치를 개시하고 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 전술한 종래 기술의 문제점을 해결하기 위한 것으로서, 단어 및 어구를 이용하여, 텍스트의 카테고리를 분류하는 오픈 디렉터리 프로젝트 기반의 텍스트 분류기 및 텍스트 분류기의 생성 및 분류 방법을 제공한다.

[0006] 다만, 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되지 않으며, 또 다른 기술적 과제들이 존재할 수 있다.

과제의 해결 수단

[0007] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 제 1 측면에 따른 오픈 디렉터리 프로젝트 기반의 텍스트 분류기는 텍스트 분류 프로그램이 저장된 메모리 및 프로그램을 실행하는 프로세서를 포함한다. 이때, 프로세서는 프로그램의 실행에 따라, 분류 대상 텍스트로부터 단어를 추출하고, 추출된 단어로부터 분류

대상 텍스트의 어구를 추출하며, 추출된 단어, 추출된 어구 및 텍스트 분류 모델에 기초하여, 분류 대상 텍스트에 대응하는 카테고리를 분류한다. 그리고 텍스트 분류 모델은 오픈 디렉터리 프로젝트에 기초하여 생성된다.

[0008] 또한, 본 발명의 제 2 측면에 따른 오픈 디렉터리 프로젝트 기반의 텍스트 분류기의 텍스트 분류 방법은 분류 대상 텍스트로부터 단어를 추출하는 단계; 추출된 단어로부터 분류 대상 텍스트에 대응하는 어구를 추출하는 단계; 및 추출된 단어, 추출된 어구 및 텍스트 분류 모델에 기초하여, 분류 대상 텍스트에 대응하는 카테고리를 분류하는 단계를 포함한다. 이때, 텍스트 분류 모델은 오픈 디렉터리 프로젝트에 기초하여 생성된다.

[0009] 그리고 본 발명의 제 3 측면에 따른 오픈 디렉터리 프로젝트 기반의 텍스트 분류기의 텍스트 분류 모델 생성 방법은 오픈 디렉터리 프로젝트의 각 카테고리로부터 단어를 추출하는 단계; 추출된 단어에 기초하여, 상기 각 카테고리로부터 어구를 추출하는 단계; 추출된 단어 및 상기 추출된 어구에 기초하여, 각 카테고리에 대응하는 단어 및 어구 빈도 벡터를 생성하는 단계; 및 각 카테고리에 대응하는 단어 및 어구 빈도 벡터에 기초하여, 텍스트 분류 모델을 생성하는 단계를 포함한다. 이때, 각 카테고리에 대응하는 단어 및 어구는 각 카테고리에 저장된 문서 집합으로부터 추출된다.

발명의 효과

[0010] 본 발명은 오픈 디렉터리 프로젝트의 각 카테고리 문서로부터 추출된 단어 및 어구에 기초하여, 텍스트 분류기를 생성한다. 그러므로 본 발명은 기존 단어만으로 텍스트 분류기를 학습할 때 보다 분류 정확도가 향상될 수 있다.

도면의 간단한 설명

[0011] 도 1은 오픈 디렉터리 프로젝트의 구조에 대한 예시도이다.

도 2는 본 발명의 일 실시예에 따른 텍스트 분류기의 블록도이다.

도 3은 본 발명의 일 실시예에 따른 텍스트 분류기의 텍스트 분류 모델의 생성 과정에 대한 블록도이다.

도 4는 본 발명의 일 실시예에 따른 어구 추출의 예시도이다.

도 5는 본 발명의 일 실시예에 따른 텍스트 분류기의 텍스트 분류 모듈의 블록도이다.

도 6은 본 발명의 일 실시예에 따른 텍스트 분류기의 텍스트 분류 방법의 순서도이다.

도 7은 본 발명의 일 실시예에 따른 텍스트 분류기의 텍스트 분류 모델 생성 방법의 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0013] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0014] 다음은 도 1을 참조하여, 오픈 디렉터리 프로젝트를 설명한다.

[0015] 도 1은 오픈 디렉터리 프로젝트의 구조에 대한 예시도이다.

[0016] 오픈 디렉터리 프로젝트는 카테고리를 노드(node)로 가지는 트리 구조로 표현될 수 있다. 트리 구조는 그래프(graph) 구조의 일종으로 순환 구조가 없고, 서로 다른 두 노드를 잇는 길이 하나뿐인 자료 구조이다.

[0017] 하나 이상의 노드를 포함하는 트리 구조는 하나의 최상위 노드와 하나 이상의 최하위 노드를 가질 수 있다. 최상위 노드는 루트 노드(root node)라고도 하며, 최하위 노드는 리프 노드(leaf node)라고 한다. 도 1에서 최상위 노드는 카테고리 C_1이 되고, 최하위 노드는 C_4, C_5, C_6 및 C_7이다.

[0018] 트리 구조에서 임의 노드의 부모 노드(parent node)는 임의 노드와 상위에 연결된 노드를 의미한다. 만약, 임

의 노드가 최상위 노드인 경우에는 부모 노드가 없을 수 있다. 최상위 노드가 아닌 임의 노드는 하나의 부모 노드를 가진다. 자식 노드(child node)는 임의 노드와 하위에 연결된 노드를 의미한다. 만약, 임의 노드가 최하위 노드인 경우에는 자식 노드가 없을 수 있다. 최하위 노드가 아닌 임의 노드는 하나 이상의 자식 노드를 가진다. 예를 들어, 도1의 카테고리 C_2의 부모 노드는 카테고리 C_1이며, 자식 노드는 카테고리 C_4와 카테고리 C_5이다.

[0019] 다음은 도 2 내지 도 5를 참조하여, 본 발명의 일 실시예에 따른 오픈 디렉터리 프로젝트(200) 기반 텍스트 분류기(100)를 설명한다.

[0020] 도 2는 본 발명의 일 실시예에 따른 텍스트 분류기(100)의 블록도이다.

[0021] 오픈 디렉터리 프로젝트(200) 기반 텍스트 분류기(100)는 기생성된 분류 모델에 기초하여, 분류 대상 텍스트의 오픈 디렉터리 프로젝트(200) 상의 카테고리를 분류한다. 이때, 텍스트 분류기(100)는 메모리 및 프로세서(120)를 포함한다.

[0022] 메모리(110)는 텍스트 분류 프로그램이 저장된다. 이때, 메모리(110)는 전원이 공급되지 않아도 저장된 정보를 계속 유지하는 비휘발성 저장장치 및 저장된 정보를 유지하기 위하여 전력이 필요한 휘발성 저장장치를 통칭하는 것이다.

[0023] 프로세서(120)는 텍스트 분류 프로그램의 실행에 따라, 분류 대상 텍스트로 단어 및 어구를 추출한다. 그리고 프로세서(120)는 추출된 단어 및 어구와 미리 생성된 텍스트 분류 모델(250)에 기초하여, 분류 대상 텍스트에 대응하는 카테고리를 분류한다. 이때, 텍스트 분류 모델(250)은 프로세서(120)가 오픈 디렉터리 프로젝트(200)에 기초하여 미리 생성된 것이다.

[0024] 이때, 프로세서(120)는 오프라인 작업 및 온라인 작업으로 구분하여 수행할 수 있다. 이때, 오프라인 작업은 프로세서(120)가 텍스트 분류 모델(250)을 생성하는 것을 의미할 수 있다. 또한, 온라인 작업은 프로세서(120)가 오프라인 작업을 통하여, 생성된 텍스트 분류 모델(250)에 기초하여, 신규 텍스트의 카테고리를 분류하는 것을 의미할 수 있다.

[0025] 텍스트 분류 모델(250) 생성 과정은 도 3을 참조하여 자세히 설명한다.

[0026] 도 3은 본 발명의 일 실시예에 따른 텍스트 분류기(100)의 텍스트 분류 모델(250)의 생성 과정에 대한 블록도이다.

[0027] 프로세서(120)는 텍스트 분류 프로그램에 포함된 텍스트 분류 모델(250) 생성 모듈(220)을 실행하여, 오픈 디렉터리 프로젝트(200)의 기 분류된 카테고리에 포함된 웹 페이지(210)를 이용하여 텍스트 분류 모델(250)을 생성할 수 있다. 이때, 텍스트 분류 모델(250) 생성 모듈(220)은 단어 추출 모듈(221), 어구 추출 모듈(222), 어구 선택 모듈(223) 및 분류 모델링 모듈(224)을 포함할 수 있다.

[0028] 이때, 도 3은 오픈 디렉터리 프로젝트(200)에 웹 페이지(210)를 이용하여, 텍스트 분류 모델(250)을 생성하는 것으로 도시되어 있다. 그러나 텍스트 분류 모델은 오픈 디렉터리 프로젝트(200)에 포함된 웹 페이지(210)뿐만 아니라 뉴스 기사, 논문 및 서적과 같은 모든 종류의 텍스트 및 문서를 포함할 수 있다.

[0029] 구체적으로 프로세서(120)는 오픈 디렉터리 프로젝트(200)에 포함된 복수의 카테고리 중 분류 모델링을 위한 하나의 카테고리를 선택할 수 있다. 그리고 프로세서(120)는 단어 추출 모듈(221)을 통하여, 선택된 카테고리에 속한 웹 페이지(210)에 대하여, 단어를 추출할 수 있다.

[0030] 이때, 프로세서(120)는 형태소 분석기(260)(morphological analyzer)를 이용하여 문서로부터 단어를 추출할 수 있다. 즉, 프로세서(120)는 문장을 형태소 분석기(260)로 전송할 수 있다. 그리고 형태소 분석기(260)를 통하여, 문장으로부터 단어가 추출되면, 프로세서(120)는 추출된 단어를 다시 단어 추출 모듈(221)을 통하여, 단어 집합(230)으로 생성할 수 있다.

[0031] 또한, 추출된 단어의 집합은 백-오브-워드(bag-of-words) 자료구조(data structure)로 저장될 수 있다. 백-오브-워드는 특정 원소 및 원소의 개수를 저장할 수 있는 자료구조이다. 그러므로 프로세서(120)는 백-오브 워드 자료구조를 통하여, 특정 텍스트에 포함된 단어를 단어와 해당 단어의 개수로 저장할 수 있다.

[0032] 프로세서(120)는 어구 추출 모듈(222)을 이용하여, 문장에 대하여 어구를 추출할 수 있다. 이때, 프로세서(120)는 단어 추출 모듈(221)을 통하여 추출된 단어 및 형태소 분석기(260)를 사용할 수 있다.

[0033] 도 4는 본 발명의 일 실시예에 따른 어구 추출의 예시도이다.

[0034] 프로세서(120)는 형태소 분석기(260)를 이용하여, 특정 문장(300)에 대한 구문 분석을 수행할 수 있다(S300). 이때, 구문 분석은 문장의 문법을 분석하여, 문장을 파싱(parsing)하고 트리(tree) 구조로 변환하는 것이다.

[0035] 도 4를 참조하면, 프로세서(120)는 문장(S)에 대한 구문 분석을 수행하여, 구문 트리를 생성할 수 있다. 이때, 구문 트리는 문법에 따라, 명사구(NP), 동사구(VP), 전치사구(PP), 명사(NNP), 동사(VB) 및 한정사(DT) 등을 포함할 수 있다.

[0036] 이때, 명사구, 동사구 및 전치사구와 같은 어구는 다른 형태소로 분리될 수 있다. 그러므로 구문 트리에서 명사구, 동사구 및 전치사 구는 비단말 노드(non-terminal node)가 될 수 있다. 또한, 단일 명사, 동사 및 한정사와 같은 단어는 다른 형태소로 분리될 수 없다. 그러므로 구문 트리에서 단일 명사, 동사 및 한정사는 단말 노드(terminal node)가 될 수 있다.

[0037] 프로세서(120)는 문장의 구문 트리를 생성한 이후, 어구 추출 모듈(222)을 통해, 구문으로부터 후보 어구를 추출할 수 있다(S310). 예를 들어, 프로세서(120)는 구문 트리에 포함된 명사구, 동사구 및 전치사 구를 후보 어구를 추출할 수 있다. 즉, 프로세서(120)는 도 4의 구문 트리로부터 "Steve Jobs", "Helped create a Silicon Valley", " create a Silicon Valley" 및 "a Silicon Valley"을 후보 어구로 추출할 수 있다.

[0038] 그리고 프로세서(120)는 어구 선택 모듈(223)을 통하여, 휴리스틱 규칙에 기초하여, 후보 어구 중 어구를 선택할 수 있다. 이때, 휴리스틱 규칙은 복수의 단어 중 스탑워드(stopword) 및 특정 품사의 단어를 제거하는 규칙, 후보 어구에 포함된 단어에 대한 스태밍(stemming)을 적용하는 규칙 및 후보 어구가 미리 정해진 개서의 단어 이상의 서로 상이한 단어를 포함하는 규칙 중 어느 하나 이상이 될 수 있다. 이때, 미리 정해진 개수는 4가 될 수 있다.

[0039] 예를 들어, 특정 품사는 한정사, 전치사, 소유격 어미 및 종속 접속사를 포함할 수 있다. 또한, 스탑워드는 "a", "an" 및 "the"과 같은 관사 또는 "is", "at" 및 "which"와 같이 문장에 자주 발생할 수 있는 단어가 될 수 있으나, 이에 한정된 것은 아니다. 스태밍은 문장에서 추출된 단어에 포함된 접사 등을 제거하여, 어간을 추출하는 것이다. 예를 들어, "cats", "catlike" 및 "catty"에 대하여, 프로세서(120)는 어간인 "cat"을 추출할 수 있다.

[0040] 다시 도 4를 참조하면, 프로세서(120)는 문서로부터 "Steve Jobs", "a Silicon valley" 및 "mercurial visionary"를 추출할 수 있다. 그리고 프로세서(120)는 어구 집합(240)에 어구 선택 모듈(223)을 통하여, 선택된 어구를 저장할 수 있다.

[0041] 또한, 프로세서(120)는 어구 선택 모듈(223)을 통하여, 어구 집합(240)에 저장된 어구 중 텍스트 분류 모델(250)의 성능을 개선할 수 있는 미리 정해진 개수의 어구를 선택할 수 있다.

[0042] 예를 들어, 프로세서(120)는 어구 집합(240)에 포함된 복수에 어구에 대한 카이 점수(chi-score)를 산출할 수 있다. 그리고 프로세서(120)는 산출된 카이 점수가 높은 어구를 선택할 수 있다. 이때, 카이 점수는 수학식 1 및 수학식 2를 통하여, 산출할 수 있다.

수학식 1

[0043]
$$\chi^2(p, c_i) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

수학식 2

[0045]
$$\chi^2_{max}(p) = \max_{i=1}^{K} \{\chi^2(p, c_i)\}$$

[0047] 수학식 1은 해당 어구(p)와 오픈 디렉터리 프로젝트(200)의 특정 카테고리(c_i)의 카이 점수를 산출하는 식이다. 이때, 수학식 1에서 A는 특정 카테고리에 포함되며, 해당 어구를 포함하는 문서의 개수이다. B는 특정 카테고리에 포함되지 않으며, 해당 어구를 포함하는 문서의 개수이다. 그리고 C는 특정 카테고리에 포함되며, 해당 어구를 포함하지 않는 문서의 개수이고, D는 특정 카테고리에 포함되지 않으며, 해당 어구를 포함하지 않은 문서의 개수이다. N은 오픈 디렉터리 프로젝트(200)에 포함된 문서 개수의 총합이다.

[0048] 또한, 수학식 2는 오픈 디렉터리 프로젝트(200)의 모든 카테고리에 대하여 산출된 카이 점수 중 가장 높은 값을 가지는 카테고리의 카이 점수를 해당 어구의 카이 점수로 산출하는 식이다. 그러므로 프로세서(120)는 수학식 2에 기초하여, 어구 집합(240)에 포함된 어구의 카이 점수를 산출할 수 있다. 그리고 프로세서(120)는 어구의 점수를 산출한 다음 전체 어구 집합(240)에서 미리 정해진 개수의 어구를 선택할 수 있다.

[0049] 한편, 프로세서(120)는 텍스트 분류기 생성 모듈(220)의 분류 모델링 모듈(224)을 통해, 각 카테고리에서 추출된 단어 집합(230) 및 어구 집합(240)을 이용하여, 단어 및 어구 발생 빈도 벡터를 생성할 수 있다. 예를 들어, 오픈 디렉터리 프로젝트(200)에서 추출된 단어가 w_1, w_2 및 w_3이고, 추출된 어구가 p_1 및 p_2이라면, 프로세서(120)는 "<w_1 개수, w_2 개수, w_3 개수, p_1 개수, p_2 개수>" 형태로 단어 및 어구 발생 빈도 벡터를 생성할 수 있다. 만약 임의의 카테고리에 w_1가 2개, w_3가 3개, p_1이 1개가 추출되었다면, 단어 및 어구 발생 빈도 벡터는 "<2, 0, 3, 1, 0>"이 될 수 있다.

[0050] 프로세서(120)는 각 카테고리로부터 추출된 단어 및 어구 발생 빈도 벡터를 학습 데이터(train data)로 이용하여, 오픈 디렉터리 프로젝트(200) 기반 텍스트 분류 모델(250)을 생성할 수 있다. 이때, 프로세서(120)는 나이브 베이즈(Naive Bayes), K-근접이웃(k Nearest Neighbor), 서포트 벡터 머신(Support Vector Machine), 센트로이드 분류 장치(centroid classifier) 등의 기계학습(machine learning) 기반의 분류 알고리즘(classification algorithm)을 이용하여 텍스트 분류 모델(250)를 생성할 수 있으나, 이에 한정되지 않는다.

[0051] 프로세서(120)는 도 5와 같이, 텍스트 분류 모델(250)이 생성되면, 생성된 텍스트 분류 모델(250)을 이용하여, 신규 문서 또는 신규 텍스트의 카테고리를 분류할 수 있다.

[0052] 도 5는 본 발명의 일 실시예에 따른 텍스트 분류기(100)의 텍스트 분류 모듈 (410)의 블록도이다.

[0053] 구체적으로 프로세서(120)는 신규 텍스트가 입력되면, 단어 추출 모듈(411)을 통하여, 단어를 추출할 수 있다. 그리고 프로세서(120)는 어구 추출 모듈(412)을 통하여, 신규 텍스트로부터 추출된 단어로부터 어구를 추출할 수 있다. 이때, 단어 추출 모듈(411) 및 단어 및 어구 추출 모듈(412)의 어구 추출 과정은 앞에서 텍스트 분류기 생성 모듈(220)에서의 단어 및 어구 추출 과정과 동일할 수 있다. 단, 프로세서(120)는 신규 텍스트에 대한 형태소 분석 없이 구문 분석만을 통하여, 어구를 추출할 수 있다.

[0054] 즉, 프로세서(120)는 신규 텍스트에 포함된 화이트 스페이스를 이용한 구문 분석을 수행하여, 복수의 토큰(token)을 추출할 수 있다. 그리고 프로세서(120)는 복수의 토큰을 이용한 단어열을 생성할 수 있다. 프로세서(120)는 생성된 단어열 중 텍스트 분류 모델(250)의 어구 집합(240)에 포함된 단어열을 어구로 추출할 수 있다.

[0055] 프로세서(120)는 벡터 생성 모듈(413)을 통하여, 신규 텍스트로부터 추출된 단어 및 어구를 이용하여, 단어 및 어구 발생 빈도 벡터를 생성할 수 있다. 이를 위하여, 프로세서(120)는 신규 텍스트로부터 추출된 단어에 대하여, 앞에서 설명한 휴리스틱 규칙에 기초하여, n개를 선택하여 조합할 수 있다. 그리고 프로세서(120)는 조합된 단어열을 신규 텍스트의 후보 어구로 사용할 수 있다.

[0056] 예를 들어, n이 3이고, 신규 텍스트로부터 3개의 단어 "steve", "jobs" 및 "helped"를 추출하였다면, 프로세서(120)는 추출된 단어를 통하여, "steve jobs", "jobs helped", "steve jobs helped"를 후보 어구를 추출할 수 있다.

[0057] 프로세서(120)는 이렇게 추출된 단어 및 어구를 이용하여, 단어 및 어구 발생 빈도 벡터를 생성할 수 있다. 그리고 프로세서(120)는 분류 모듈(414)을 통하여, 생성된 단어 및 어구 발생 빈도 벡터에 대하여, 텍스트 분류 모델(250)을 적용하여 신규 텍스트의 카테고리를 분류할 수 있다.

[0058] 이때, 프로세서(120)는 분류 모듈(414)을 통하여, 텍스트 분류 모델(250)에 저장된 각 카테고리의 단어 및 어구 발생 빈도 벡터와 신규 텍스트를 통하여, 추출된 단어 및 어구 발생 빈도 벡터의 유사도를 산출할 수 있다. 예를 들어, 유사도는 자카드 계수(jaccard coefficient), 스피어만 상관계수(Spearman's correlation coefficient), 유클리드 거리(Euclidean distance), 코사인(cosine) 유사도 등이 될 수 있으며, 이에 한정된 것은 아니다.

[0059] 프로세서(120)는 각 카테고리의 단어 및 어구 발생 빈도 벡터와의 유사도를 산출한 이후, 유사도가 가장 큰 카테고리 및 해당 카테고리의 유사도를 신규 텍스트의 카테고리 및 점수를 분류 결과(420)으로 산출할 수 있다.

[0060] 다음은 도 6을 참조하여, 본 발명의 일 실시예에 따른 오픈 디렉터리 프로젝트(200) 기반의 텍스트 분류기(100)의 텍스트 분류 방법을 설명한다.

[0061] 도 6은 본 발명의 일 실시예에 따른 텍스트 분류기(100)의 텍스트 분류 방법의 순서도이다.

[0062] 텍스트 분류기(100)는 분류 대상 텍스트로부터 단어를 추출한다(S500).

[0063] 텍스트 분류기(100)는 단어를 추출한 이후, 추출된 단어로부터 분류 대상 텍스트에 대응하는 어구를 추출한다(S510).

[0064] 텍스트 분류기(100)는 추출된 단어, 추출된 어구 및 텍스트 분류 모델(250)에 기초하여, 분류 대상 텍스트에 대응하는 카테고리를 분류한다. 이때, 텍스트 분류기(100)는 텍스트 분류 모델(250)은 오픈 디렉터리 프로젝트(200)에 기초하여 생성된 것이다.

[0065] 다음은 도 7을 참조하여, 본 발명의 일 실시예에 따른 오픈 디렉터리 프로젝트(200) 기반의 텍스트 분류기(100)의 텍스트 분류 모델(250) 생성 방법을 설명한다.

[0066] 도 7은 본 발명의 일 실시예에 따른 텍스트 분류기(100)의 텍스트 분류 모델(250) 생성 방법의 순서도이다.

[0067] 텍스트 분류기(100)는 오픈 디렉터리 프로젝트(200)의 각 카테고리로부터 단어를 추출한다(S600).

[0068] 텍스트 분류기(100)는 추출된 단어에 기초하여, 각 카테고리로부터 어구를 추출한다(S610).

[0069] 텍스트 분류기(100)는 추출된 단어 및 추출된 어구에 기초하여, 각 카테고리에 대응하는 단어 및 어구 빈도 벡터를 생성한다(S620). 이때, 각 카테고리에 대응하는 단어 및 어구는 각 카테고리에 저장된 문서 집합으로부터 추출되는 것이다.

[0070] 구체적으로 텍스트 분류기(100)는 각 카테고리에 저장된 문서 집합으로부터 구문분석에 기초하여, 후보 어구 집합(240)을 추출할 수 있다. 그리고 텍스트 분류기(100)는 후보 어구 집합(240)에 포함된 후보 어구에 대한 휴리스틱 규칙에 기초하여, 각 카테고리에 대응하는 어구를 추출할 수 있다.

[0071] 이때, 휴리스틱 규칙은 후보 어구에 포함된 복수의 단어 중 스탑워드 및 특정 품사의 단어를 제거하는 규칙, 후보 어구에 포함된 단어에 대한 스태밍을 적용하는 규칙 및 후보 어구가 미리 정해진 개수의 단어 이상의 서로 상이한 단어를 포함하는 규칙 중 어느 하나 이상이 될 수 있다. 특정 품사는 한정사, 전치사, 소유격 어미 및 종속 접속사를 포함할 수 있다.

[0072] 텍스트 분류기(100)는 각 카테고리에 대응하는 단어 및 어구 빈도 벡터에 기초하여, 텍스트 분류 모델(250)을 생성한다(S630).

[0073] 본 발명의 일 실시예에 따른 오픈 디렉터리 프로젝트(200) 기반의 텍스트 분류기(100), 및 텍스트 분류기(100)의 생성 및 분류 방법은 오픈 디렉터리 프로젝트(200)의 각 카테고리 문서로부터 추출된 단어 및 어구에 기초하여, 텍스트 분류기(100)를 생성한다. 그러므로 텍스트 분류기(100), 및 텍스트 분류기(100)의 생성 및 분류 방법은 기존 단어만으로 텍스트 분류기(100)를 학습할 때 보다 분류 정확도가 향상될 수 있다.

[0074] 본 발명의 일 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전

송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.

[0075] 본 발명의 방법 및 시스템은 특정 실시예와 관련하여 설명되었지만, 그것들의 구성 요소 또는 동작의 일부 또는 전부는 범용 하드웨어 아키텍처를 갖는 컴퓨터 시스템을 사용하여 구현될 수 있다.

[0076] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0077] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

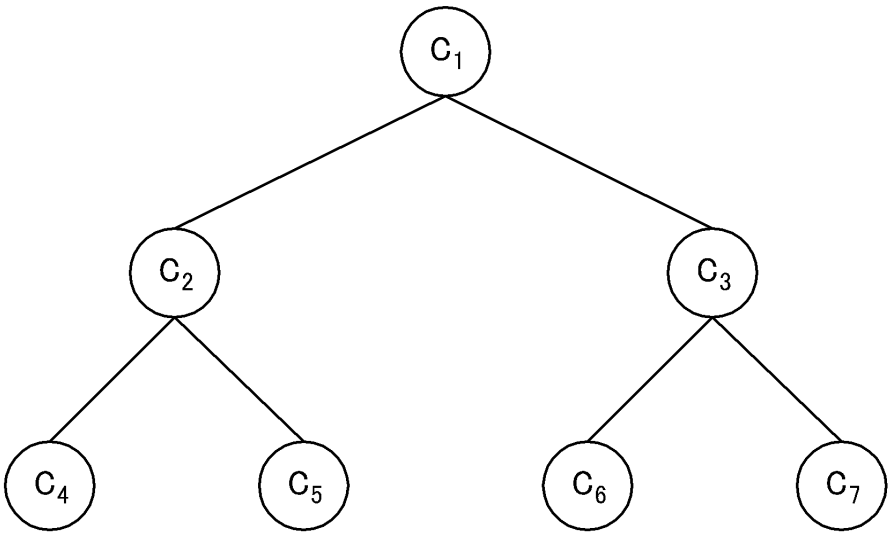
[0078] 100: 오픈 디렉터리 프로젝트 기반의 텍스트 분류기

110: 메모리

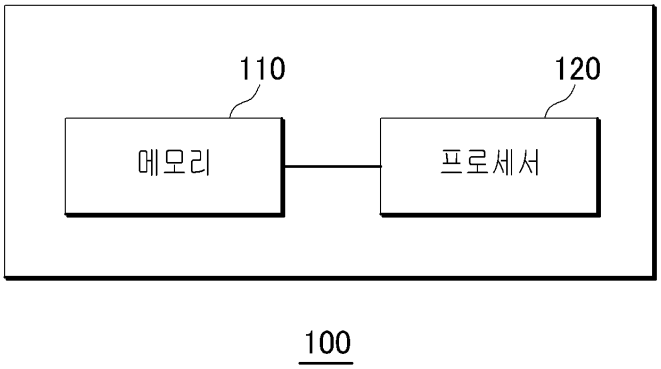
120: 프로세서

도면

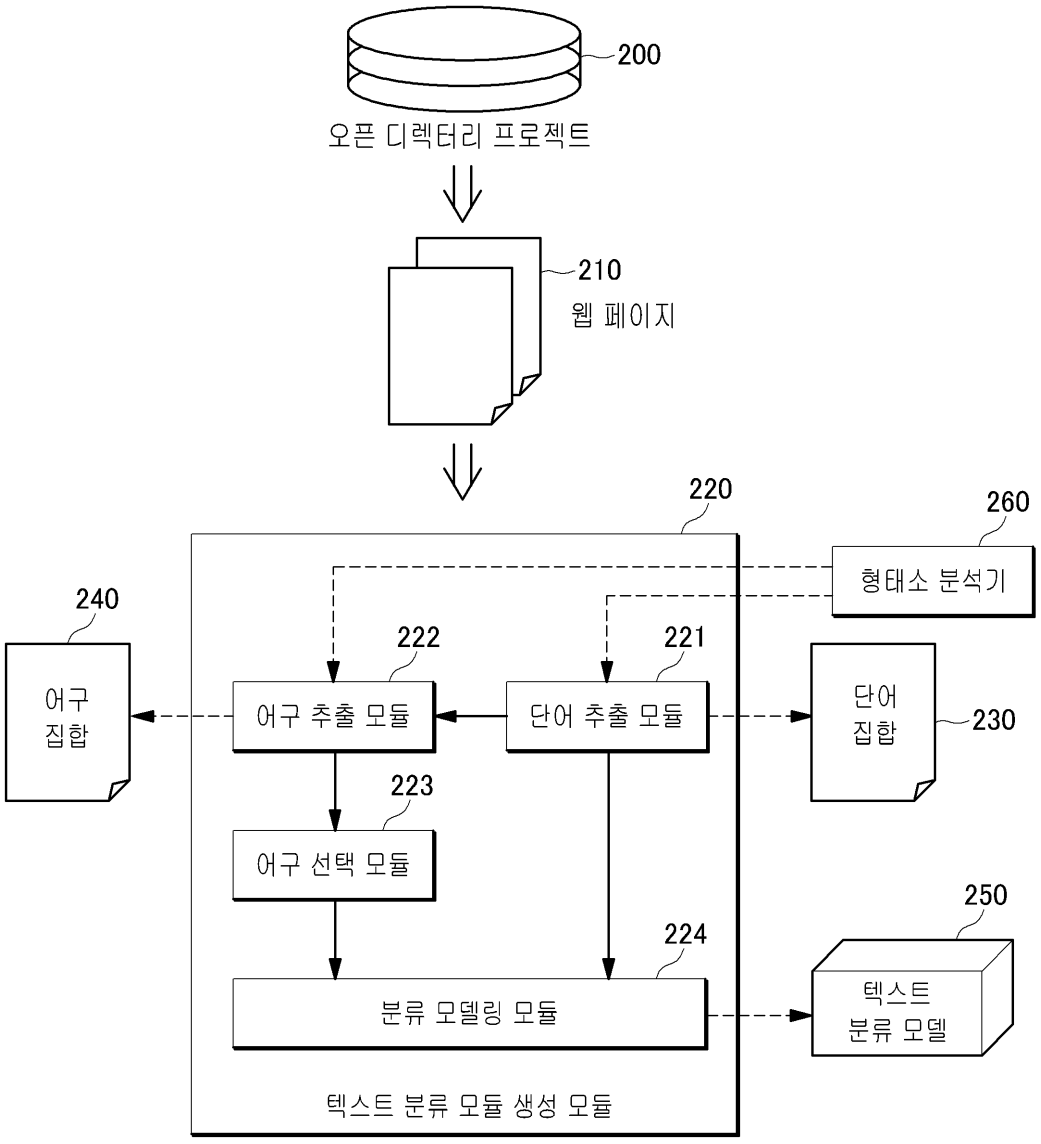
도면1



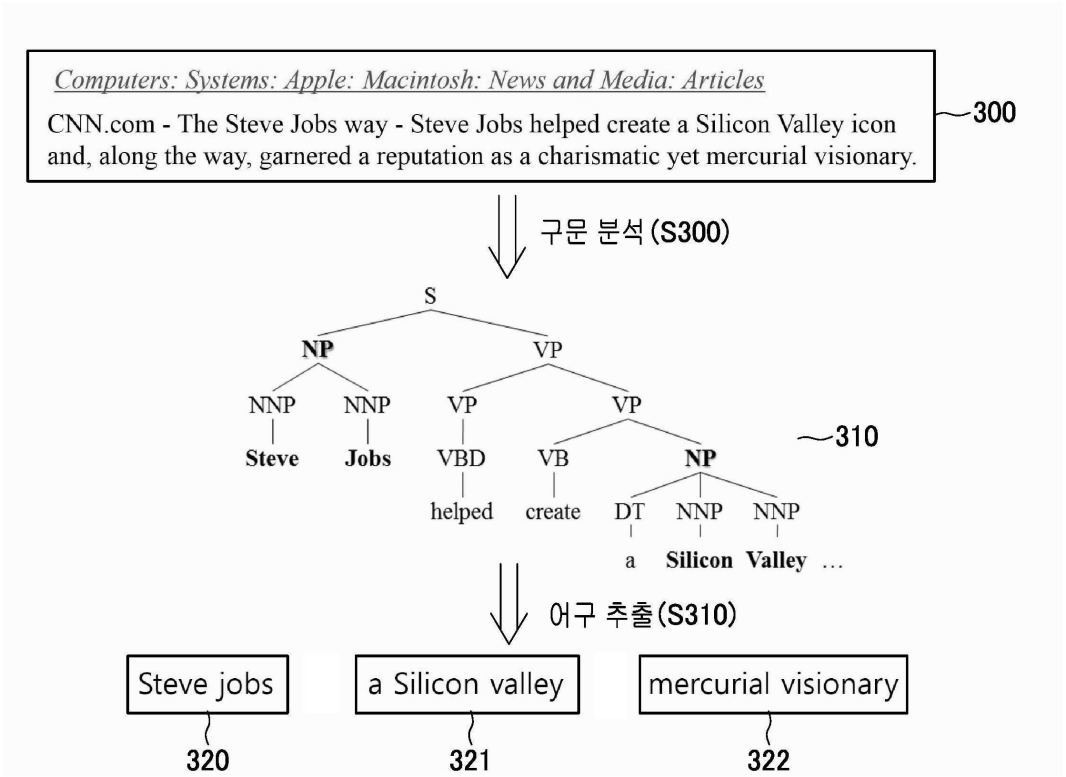
도면2


110
120
메모리
프로세서
100

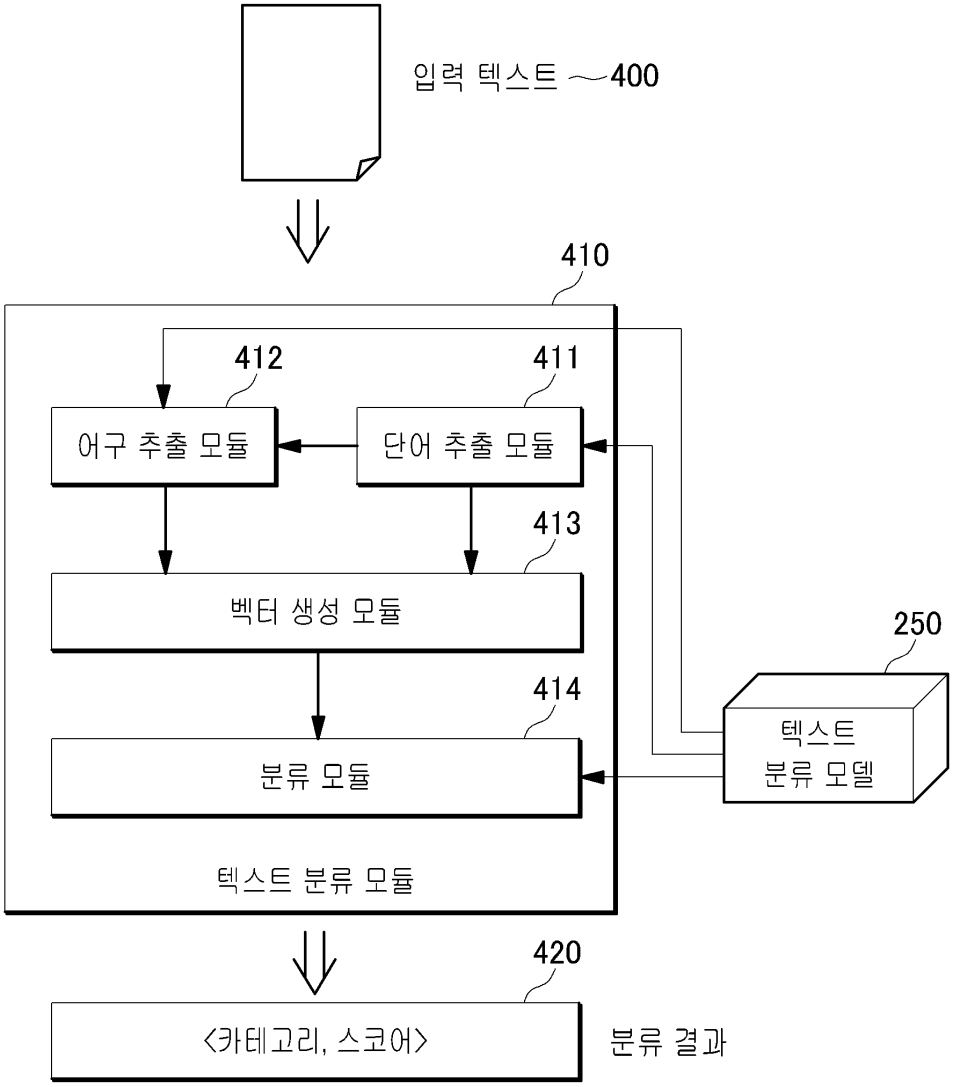
도면3


200
오픈 디렉터리 프로젝트
210
웹 페이지
220
260
형태소 분석기
240
어구
집합
222
221
어구 추출 모듈
단어 추출 모듈
단어
집합
230
223
어구 선택 모듈
250
224
분류 모델링 모듈
텍스트
분류 모델
텍스트 분류 모듈 생성 모듈

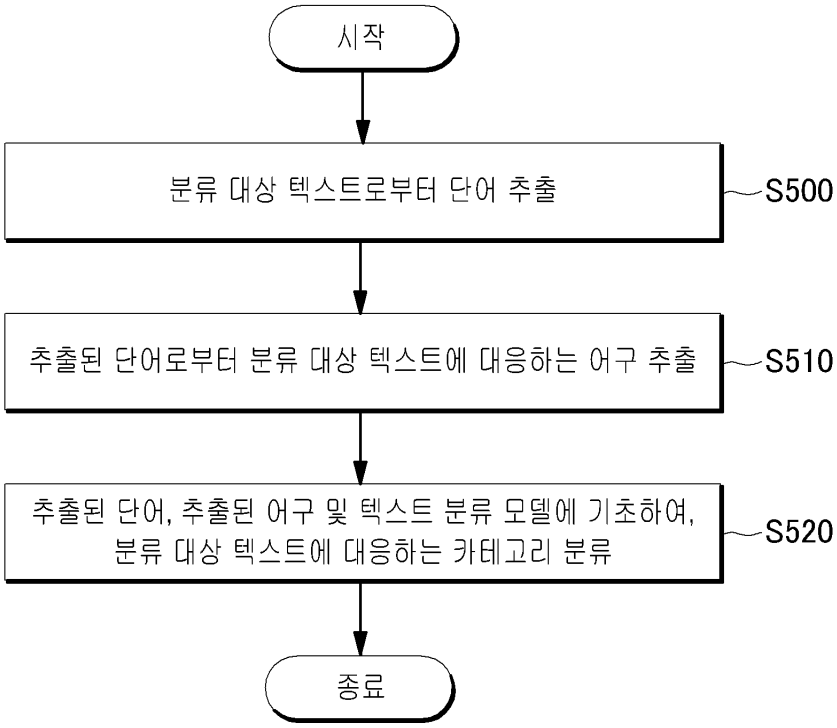
도면4


Computers: Systems: Apple: Macintosh: News and Media: Articles
CNN.com - The Steve Jobs way - Steve Jobs helped create a Silicon Valley icon and, along the way, garnered a reputation as a charismatic yet mercurial visionary.
300
구문 분석(S300)
S
NP
VP
NNP
NNP
VP
VP
Steve
Jobs
VBD
VB
NP
310
helped
create
DT
NNP
NNP
a
Silicon
Valley
…
어구 추출(S310)
Steve jobs
a Silicon valley
mercurial visionary
320
321
322

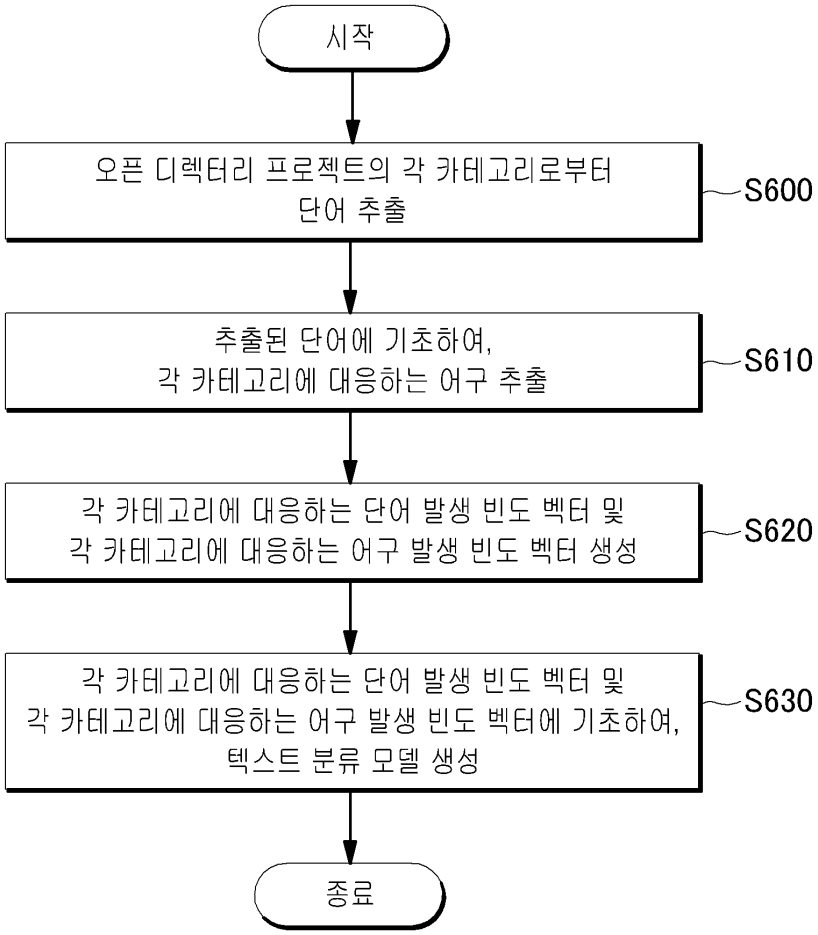
도면5


입력 텍스트
400
410
412
411
어구 추출 모듈
단어 추출 모듈
413
벡터 생성 모듈
250
414
텍스트
분류 모델
분류 모듈
텍스트 분류 모듈
420
<카테고리, 스코어>
분류 결과

도면6


시작
분류 대상 텍스트로부터 단어 추출
S500
추출된 단어로부터 분류 대상 텍스트에 대응하는 어구 추출
S510
추출된 단어, 추출된 어구 및 텍스트 분류 모델에 기초하여, 분류 대상 텍스트에 대응하는 카테고리 분류
S520
종료

도면7


시작
오픈 디렉터리 프로젝트의 각 카테고리로부터 단어 추출
S600
추출된 단어에 기초하여, 각 카테고리에 대응하는 어구 추출
S610
각 카테고리에 대응하는 단어 발생 빈도 벡터 및 각 카테고리에 대응하는 어구 발생 빈도 벡터 생성
S620
각 카테고리에 대응하는 단어 발생 빈도 벡터 및 각 카테고리에 대응하는 어구 발생 빈도 벡터에 기초하여, 텍스트 분류 모델 생성
S630
종료