(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년10월12일
(11) 등록번호 10-1190230
(24) 등록일자 2012년10월05일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01)
(21) 출원번호 10-2005-0068058
(22) 출원일자 2005년07월26일
심사청구일자 2010년07월23일
(65) 공개번호 10-2006-0048779
(43) 공개일자 2006년05월18일
(30) 우선권주장
10/900,021 2004년07월26일 미국(US)
(56) 선행기술조사문헌
Finding Co-occurring Text Phrases by Combining Sequence and Frequent Set Discovery(1999)
전체 청구항 수 : 총 25 항

(73) 특허권자
구글 인코포레이티드
미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이 1600 (우:94043)
(72) 발명자
안나 엘. 패터슨
미국 95120 캘리포니아주 산 조세 손트리 코트 1127
(74) 대리인
문기상, 문두현

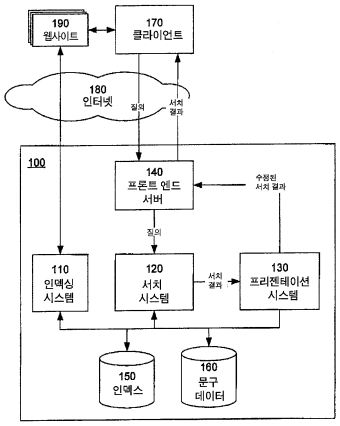
심사관 : 이명진

(54) 발명의 명칭 **정보 검색 시스템에서의 문구 식별**

(57) 요 약

정보 검색 시스템은 문서를 인덱싱하고, 검색하고, 조직화하고, 설명하기 위해서 문구를 사용한다. 문서 내의 다른 문구의 존재를 예측하는 문구가 식별된다. 문서는 그것에 포함되는 문구에 따라서 인덱싱된다. 관련 문구 및 문구 확장이 또한 식별된다. 질의 내의 문구는 식별되고 문서를 검색하고 랭킹하기 위해서 사용된다. 문구는 서치 결과 내의 문서를 클러스터링하고, 문서 설명을 생성하고, 서치 결과 및 인덱스로부터 중복 문서를 제거하기 위해서 또한 사용된다.

대 표 도 - 도1



특허청구의 범위

청구항 1

문서 컬렉션(document collection) 내의 문구를 식별하는 컴퓨터 구현 방법으로서, 컴퓨터가,

상기 문서 컬렉션 내의 문서로부터 가능 문구(possible phrases)를 수집하는 단계;

개개의 가능 문구를 상기 개개의 가능 문구의 출현 빈도에 따라 양호 문구 또는 불량 문구 중 어느 하나로서 분류하는 단계;

상기 문서 컬렉션 내의 양호 문구 g_j 및 g_k의 쌍에 대하여, 상기 문서 컬렉션 내의 g_j 및 g_k의 실질 공통 출현율 및 g_j 및 g_k의 예측 공통 출현율의 함수로서, g_j에 대한 g_k의 정보 이득을 판정하는 단계;

상기 문서 컬렉션 내에서 적어도 하나의 다른 양호 문구의 출현을 예측하는 양호 문구를 유효 문구로서 선택적으로 보유하는 단계 - g_j의 존재하에 g_k의 판정된 정보 이득이 미리 결정된 제 1 임계치를 넘는 경우에, 양호 문구 g_j는 상기 문서 컬렉션 내에서 다른 양호 문구 g_k의 출현을 예측함 - ;

선택적으로 유지된 복수의 유효 문구 g_x에 대하여, 상기 g_x의 존재하에 g_y의 정보 이득이 미리 결정된 상기 제 1 임계치보다 더 제한적인 미리 결정된 제 2 임계치를 넘을 경우에, 상기 문구 g_y를 상기 g_x의 관련 문구로서 식별하는 단계;

상기 유효 문구 및 상기 식별된 관련 문구를 컴퓨터 판독 가능한 저장 매체에 저장하는 단계를 실행하는 컴퓨터 구현 방법.

청구항 2

제 1 항에 있어서,

상기 가능 문구를 수집하는 단계는,

문서의 단어(word)를 복수단어(multiword) 문구 윈도(window)로 트래버스(traverse)하는 단계, 및 상기 윈도 내의 제 1 단어로 시작하는 상기 윈도 내의 모든 단어들의 시퀀스를 후보 문구(candidate phase)로서 선택하는 단계를 포함하는 컴퓨터 구현 방법.

청구항 3

제 2 항에 있어서,

상기 문구 윈도는 적어도 4개의 단어를 포함하는 컴퓨터 구현 방법.

청구항 4

제 1 항에 있어서,

상기 가능 문구를 수집하는 단계는,

각각의 상기 가능 문구와 각각의 상기 양호 문구마다 상기 문구를 포함하는 문서 수의 빈도 카운트(frequency count)를 유지하는 단계;

각각의 상기 가능 문구와 각각의 상기 양호 문구마다 상기 문구의 인스턴스(instance) 수의 빈도 카운트를 유지하는 단계; 및

각각의 상기 가능 문구와 각각의 상기 양호 문구마다 상기 문구의 구별된 인스턴스(distinguished instance) 수의 빈도 카운트를 유지하는 단계를 포함하는 컴퓨터 구현 방법.

청구항 5

제 4 항에 있어서,

상기 문구의 구별되는 인스턴스는 문법 또는 포맷 마커(grammatical or format markers)에 의해서 상기 문서 내

의 인접하는 컨텐트(neighboring content)로부터 구별되는 문구를 포함하는 컴퓨터 구현 방법.

청구항 6

제 1 항에 있어서,

각각의 상기 가능 문구를 양호 문구 또는 불량 문구 중 어느 하나로서 분류하는 단계는,

상기 가능 문구가 최소 수의 문서에 나타나고, 상기 문서 컬렉션에서 최소 수의 인스턴스로 나타나는 경우에, 상기 가능 문구를 양호 문구로 분류하는 단계를 포함하는 컴퓨터 구현 방법.

청구항 7

제 1 항에 있어서,

각각의 상기 가능 문구를 양호 문구 또는 불량 문구 중 어느 하나로서 분류하는 단계는,

상기 가능 문구가 상기 문서 컬렉션에서 최소 수의 구별되는 인스턴스로 나타나는 경우에, 상기 가능 문구를 양호 문구로 분류하는 단계를 포함하는 컴퓨터 구현 방법.

청구항 8

제 1 항에 있어서,

상기 미리 정해진 제 1 임계치는 1인 컴퓨터 구현 방법.

청구항 9

제 1 항에 있어서,

상기 g_k에 대한 상기 g_j의 정보 이득은,

$I(j,k) = A(j,k)/E(j,k)$이고,

여기서, 상기 $A(j,k)$는 g_j와 g_k의 실질 공통 출현율이고;

상기 $E(j,k)$는 g_j와 g_k의 예측 공통 출현율인 컴퓨터 구현 방법.

청구항 10

제 9 항에 있어서,

상기 g_j와 상기 g_k가 서로 미리 정해진 단어 수 이내인 경우에, 양호 문구 g_j와 g_k는 문서 내에 공통으로 출현하는 컴퓨터 구현 방법.

청구항 11

제 1 항에 있어서,

상기 문서 컬렉션 내에서 적어도 하나의 다른 양호 문구의 출현을 예측하는 양호 문구를 보유하는 단계는,

복수의 다른 양호 문구에 대한 정보 이득이 미리 정해진 임계치보다 작은 양호 문구를 제거하는 단계를 포함하는 컴퓨터 구현 방법.

청구항 12

제 1 항에 있어서,

상기 양호 문구로부터 불완전한 문구를 제거하는 단계를 더 포함하는 컴퓨터 구현 방법.

청구항 13

제 12 항에 있어서,

상기 불완전한 문구는 그것의 문구 확장만을 예측하는 문구이고, 상기 문구의 문구 확장은 상기 문구로 시작하

는 상기 문구의 수퍼-시퀀스(super sequence)인 컴퓨터 구현 방법.

청구항 14

제 12 항에 있어서,

각각의 상기 불완전한 문구마다 상기 불완전한 문구의 적어도 하나의 문구 확장을 유지하는 단계, 및

불완전한 문구인 서치 질의(search query) 내의 문구에 응하여, 상기 불완전한 서치 문구의 적어도 하나의 문구 확장을 상기 서치 질의에 포함시키는 단계를 더 포함하는 컴퓨터 구현 방법.

청구항 15

제 1 항에 있어서,

미리 정해진 상기 제 2 임계치는 100인 컴퓨터 구현 방법.

청구항 16

제 1 항에 있어서,

각각의 상기 문구 g_x마다, 상기 문구 및 적어도 하나의 관련 문구 g_y를 포함하는 클러스터를 식별하는 단계를 더 포함하는 컴퓨터 구현 방법.

청구항 17

제 1 항에 있어서,

각각의 상기 문구 g_x마다, 복수의 관련 문구를 포함하는 세트 R을 식별하는 단계;

상기 세트 R 내의 각 쌍의 관련 문구마다, 상기 관련 문구 쌍의 정보 이득을 판정하는 단계; 및

문구 g_x와, 상기 세트 R 내의 서로 다른 문구에 대하여 제로가 아닌 정보 이득을 가지는 상기 세트 R 내의 각각의 관련 문구를 상기 g_x의 관련 문구의 클러스터로서 식별하는 단계를 더 포함하는 컴퓨터 구현 방법.

청구항 18

제 17 항에 있어서,

각각의 클러스터에, 상기 클러스터에 포함되는 상기 관련 문구의 함수로서 고유의 클러스터 번호를 할당하는 단계를 더 포함하는 컴퓨터 구현 방법.

청구항 19

제 17 항에 있어서,

상기 클러스터에, 상기 클러스터 내의 상기 관련 문구의 최고의 정보 이득을 가지는 관련 문구를 포함하는 이름을 할당하는 단계를 더 포함하는 컴퓨터 구현 방법.

청구항 20

제 1 항에 있어서,

문구 g_j에 대하여, 각각의 비트 위치가 유효 문구에 대응하고 있는 비트 벡터를 저장하는 단계를 더 포함하고, 상기 비트 위치에 대응하는 비트는 상기 유효 문구가 상기 g_j의 관련 문구인지를 나타내는 컴퓨터 구현 방법.

청구항 21

문서 컬렉션 내의 문구를 식별하기 위한 컴퓨터 실시 가능한 명령을 저장한 컴퓨터 판독 가능한 저장 매체로서, 프로세서가,

상기 문서 컬렉션 내의 문서로부터 가능 문구를 수집하는 단계;

개개의 가능 문구를 상기 가능 문구의 출현 빈도에 따라 양호 문구 또는 불량 문구 중 어느 하나로서 분류하는 단계;

상기 문서 컬렉션 내의 양호 문구 g_j 및 g_k의 쌍에 대하여, 상기 문서 컬렉션 내의 g_j 및 g_k의 실질 공통 출현율 및 g_j 및 g_k의 예측 공통 출현율의 함수로서, g_j에 대한 g_k의 정보 이득을 판정하는 단계;

상기 문서 컬렉션 내에서 적어도 하나의 다른 양호 문구의 출현을 예측하는 양호 문구를 유효 문구로서 선택적으로 보유하는 단계 - g_j의 존재하에 g_k의 판정된 정보 이득이 미리 결정된 제 1 임계치를 넘는 경우에, 양호 문구 g_j는 상기 문서 컬렉션 내에서 다른 양호 문구 g_k의 출현을 예측함 - ;

선택적으로 보유된 복수의 유효 문구 g_x에 대하여, 상기 g_x의 존재하에 g_y의 정보 이득이 미리 결정된 상기 제 1 임계치보다 더 제한적인 미리 결정된 제 2 임계치를 넘을 경우에, 상기 문구 g_y를 상기 g_x의 관련 문구로서 식별하는 단계; 및

상기 선택적으로 보유된 양호 문구를 컴퓨터 판독 가능한 저장 매체에 저장하는 단계

를 포함하는 연산을 수행하도록 제어하게 구성된 컴퓨터 실시 가능한 명령을 저장한 컴퓨터 판독 가능한 저장 매체.

청구항 22

제 21 항에 있어서,

상기 g_k에 대한 상기 g_j의 정보 이득은 $I(j,k) = A(j,k)/E(j,k)$로서 정의되고,

여기서, 상기 $A(j,k)$는 g_j와 g_k의 실질 공통 출현율이고, 상기 $E(j,k)$는 g_j와 g_k의 예측 공통 출현율인 컴퓨터 실시 가능한 명령을 저장한 컴퓨터 판독 가능한 저장 매체.

청구항 23

문서 컬렉션 내의 문서를 인덱싱하는 컴퓨터 구현 방법으로서, 컴퓨터가,

상기 문서 컬렉션 내의 문서로부터 가능 문구를 수집하는 단계;

각각의 문구를, 상기 가능 문구의 출현 빈도에 따라, 양호 문구 또는 불량 문구 중 어느 하나로 분류하는 단계;

상기 문서 컬렉션 내의 양호 문구 g_j 및 g_k의 쌍에 대하여, 상기 문서 컬렉션 내의 g_j 및 g_k의 실질 공통 출현율 및 g_j 및 g_k의 예측 공통 출현율의 함수로서, g_j에 대한 g_k의 정보 이득을 판정하는 단계;

상기 문서 컬렉션 내의 적어도 하나의 다른 양호 문구의 출현을 예측하는 양호 문구만을 유효 문구로서 컴퓨터 판독 가능한 저장 매체에 저장하는 단계 - g_j의 존재하에 g_k의 판정된 정보 이득이 미리 결정된 제 1 임계치를 넘는 경우에, 양호 문구 g_j는 상기 문서 컬렉션 내에서 다른 양호 문구 g_k의 출현을 예측함 - ;

선택적으로 보유된 복수의 유효 문구 g_x에 대하여, 상기 g_x의 존재하에 g_y의 정보 이득이 미리 결정된 상기 제 1 임계치보다 더 제한적인 미리 결정된 제 2 임계치를 넘을 경우에, 상기 문구 g_y를 상기 g_x의 관련 문구로서 식별하는 단계; 및

상기 유효 문구 및 상기 관련 문구에 대해 상기 문서 컬렉션 내의 문서를 인덱싱하는 단계를 실행하는 컴퓨터 구현 방법.

청구항 24

제 23 항에 있어서,

상기 g_j에 대한 상기 g_k의 정보 이득은 $I(j,k) = A(j,k)/E(j,k)$로서 정의되고,

여기서, 상기 $A(j,k)$는 g_j와 g_k의 실질 공통 출현율이고, 상기 $E(j,k)$는 g_j와 g_k의 예측 공통 출현율인 컴퓨터

구현 방법.

청구항 25

제 23 항에 있어서,

상기 가능 문구를 수집하는 단계는,

각각의 상기 가능 문구와 각각의 상기 양호 문구마다 상기 문구를 포함하는 문서 수의 빈도 카운트를 유지하는 단계;

각각의 상기 가능 문구와 각각의 상기 양호 문구마다 상기 문구의 인스턴스 수의 빈도 카운트를 유지하는 단계; 및

각각의 상기 가능 문구와 각각의 상기 양호 문구마다, 문법 또는 포맷 마커에 의해 문서 내의 인접하는 컨텐트로부터 구별되는 문구를 포함하는 상기 문구의 구별된 인스턴스 수의 빈도 카운트를 유지하는 단계를 포함하는 컴퓨터 구현 방법.

청구항 26

삭제

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0015] 본 발명은 인터넷과 같은 대규모 자료에서 문서를 인덱싱, 서치(searching), 및 분류하는 정보 검색 시스템에 관한 것이다.

[0016] 오늘날 정보 검색 시스템(일반적으로 서치 엔진으로 불림)은 인터넷과 같은 대규모적이고, 다양하고, 증대해 가는 자료들에서 정보를 찾기 위한 필수 도구이다. 일반적으로, 서치 엔진은 문서(또는 "페이지")들을 각 문서 내에 존재하는 개개의 단어와 관련시키는 인덱스를 생성한다. 많은 질의 용어를 담고 있는 질의에 응하여, 전형적으로는 문서 내에 존재하는 일부의 질의 용어를 갖는 것에 기초하여 문서가 검색된다. 그리고 나서 검색된 문서들은 질의 용어, 호스트 도메인, 링크 분석 등의 발생 빈도와 같은 기타 다른 통계 측도에 따라 랭킹된다. 그리고 나서 검색된 문서들은 전형적으로는 랭킹된 순서대로 어떠한 부가적인 그룹화 또는 부과되는 계층 없이 유저에게 제공된다. 일부의 경우에, 유저가 문서의 내용을 일견할 수 있도록 선택된 문서의 텍스트 부분이 제공된다.

[0017] 질의 용어의 직접적인 "부울" 매칭은 잘 알려진 제한을 갖고, 특히 질의 용어를 갖지 않고 관련 단어를 갖는 문서들은 식별하지 않는다. 예를 들면, 전형적인 부울 시스템에서는, "Australian Shepherds"에 대한 서치는 정확한 질의어를 갖지 않는 보더 콜리(Border Collie)와 같은 목양견(牧羊犬)에 대한 문서를 불러오지 않는다. 오히려, 이러한 시스템은 Australia에 관한 문서(개와는 관계 없음)와 일반적으로 "shepherds"에 관한 문서를 검색하여 높게 랭킹할 가능성이 있다.

[0018] 여기에서 문제는 종래의 시스템이 개념보다는 개별적인 용어에 기초하여 문서를 인덱싱한다는 것이다. 개념은 종종 "Australian Shepherd", "President of the United States", 또는 "Sundance Film Festival"과 같은 문구로 표현된다. 기껏해야, 일부의 선행 시스템들은 인간 운영자에 의해서 전형적으로 선택되는 소정의 매우 제한된 세트의 '알려진' 문구에 대해서 문서를 인덱싱할 것이다. 문구의 인덱싱은 인지된 계산적 메모리 요건으로 인해, 모든 가능한 문구 즉 3, 4, 또는 5 이상의 단어를 식별하는 것이 통상적으로 회피된다. 예를 들면, 임의의 5개 단어가 문구를 구성하고, 큰 자료가 적어도 200,000개의 고유 용어를 갖는다고 가정하면, 대략 3.2×10^{26}개의 가능한 문구가 존재하고, 현존하는 것보다 훨씬 더 많은 시스템이 메모리에 저장될 수 있거나 그렇지 않으면 프로그램적으로 조작될 수 있다. 또 다른 문제는 새로운 개개의 단어들이 생성되는 것보다 훨씬 더 종종, 사용에 의해 문구들이 계속적으로 용어집에서 등록 및 사라진다. 과학기술, 예술, 월드 이벤트, 및 법률

등의 자료로부터 새로운 문구들이 항상 생성되고 있다. 다른 문구들은 시간이 지날수록 사용이 감소될 것이다.

[0019] 일부의 기존 정보 검색 시스템은 개별 단어의 공통 출현 패턴을 이용하여 개념의 검색을 제공하는 시도를 한다. 이러한 시스템에서 "President"와 같은 한 단어의 서치는 또한 "White"와 "House"와 같이 "President"와 함께 종종 나타나는 다른 단어들을 갖는 문서를 검색할 것이다. 이러한 방법이 개별 단어의 레벨에서 개념적으로 관련된 문서를 갖는 서치 결과를 생성할 수 있지만, 이것은 통상적으로 공통 출현 문구 사이에서 내제되어 있는 주제적인 관계를 잡아내지 않는다.

발명이 이루고자 하는 기술적 과제

[0020] 따라서, 대규모 자료에서 문구를 포괄적으로 식별하고, 문구에 따라 문서를 인덱싱하고, 이들 문구에 따라 문서를 서치 및 랭킹하고, 문서에 관한 부가적인 클러스터링 및 설명 정보를 제공할 수 있는 정보 검색 시스템 및 방법을 제공할 필요가 있다.

발명의 구성 및 작용

[0021] 정보 검색 시스템 및 방법은 문구를 사용하여 문서 컬랙션에서 문서를 인덱싱, 서치, 랭킹 및 설명한다. 이 시스템은 문서 컬랙션에서 상당히 자주 사용되어 "유효" 또는 "양호" 문구인 것을 나타내는 문구를 식별하는데 적합하다. 이러한 방식으로 복수의 어구, 예를 들면 4, 5, 또는 그 이상의 용어의 문구가 식별될 수 있다. 이는 주어진 개수의 단어의 모든 가능한 시퀀스로부터의 결과의 가능한 문구마다 식별 및 인덱싱해야 할 문제를 해소한다.

[0022] 또한 이 시스템은 문서 내의 다른 문구의 존재를 예측하는 문구의 능력에 기초하여 서로 관련된 문구를 식별하는데 적합하다. 보다 구체적으로, 실제의 2 문구의 공통 출현율을 예측되는 2 문구의 공통 출현율과 관련시키는 예상 측도가 사용된다. 실제의 공통 출현율과 예상되는 공통 출현율과의 비율로서의 정보 이득은 이러한 하나의 예상 측도이다. 예상 측도가 소정의 임계치를 초과하는 경우에 2 문구가 관련된다. 그 경우, 제 2 문구는 제 1 문구에 대해서 큰 정보 이득을 갖는다. 의미적으로, 관련된 문구는 "President of the United States"와 "White House"와 같이 주어진 주제 또는 개념을 논의 또는 설명하는데 공통적으로 사용되는 것들이다. 주어진 문구에 대해, 관련되는 문구는 각 예상 측도에 기초하여 그들의 관련성 또는 중요성에 따라 배열될 수 있다.

[0023] 정보 검색 시스템은 유효 또는 양호 문구에 의해 문서 컬렉션에서 문서를 인덱싱한다. 각 문구에 대해, 포스팅 리스트(posting list)는 문구를 담고 있는 문서를 식별한다. 또한, 주어진 문구에 대해, 제 2 리스트, 벡터, 또는 다른 구조가 사용되어, 주어진 문구와 관련되는 문구 중 어느 것이 또한 주어진 문구를 담고 있는 각 문서에 존재하는지를 나타내는 데이터를 저장한다. 이러한 방식으로, 시스템은 어느 문서가 서치 질의에 응하여 어느 문구를 담고 있는지뿐만 아니라 어느 문서가 또한 질의 문구와 관련되는 문구를 담고 있는지를 쉽게 식별할 수 있고, 따라서 질의 문구에서 표현되는 주제 또는 개념에 대해서 구체적이 될 가능성이 크다.

[0024] 문구 및 관련 문구의 사용은 또한 의미적으로 유의한 문구의 그룹화를 나타내는 관련 문구의 클러스터를 생성 및 사용하게 한다. 클러스터는 클러스터 내의 모든 문구들 사이에서 매우 높은 예상 측도를 갖는 관련 문구로부터 식별된다. 클러스터는 어떤 문서가 서치 결과 및 이들의 배열에 포함될지를 선택할 뿐만 아니라 서치 결과로부터 문서를 제거하는 것을 포함하여 서치 결과를 편성하는데 사용될 수 있다.

[0025] 정보 검색 시스템은 또한 질의에 응하여 문서를 서치하는 경우에 문구를 사용하는데 적합하다. 질의는 이 질의에 존재하는 임의의 문구를 식별하도록 처리되어, 질의 문구 및 관련 문구 정보에 대한 연관된 포스팅 리스트를 검색한다. 또한, 어떤 경우에 유저는 "President of the"와 같이 서치 질의에서 불완전한 문구를 입력할 수 있다. 이와 같은 불완전한 문구는 "President of the United States"와 같이 문구 확장에 의해 식별되어 대체될 수 있다. 이는 유저의 가장 가능성 있는 서치가 사실상 실행된다는 것을 보장하는데 도움을 준다.

[0026] 관련 문구 정보는 또한 시스템에 의해 어떤 문서가 서치 결과에 포함되는지를 식별 또는 선택하는데 사용될 수 있다. 관련 문구 정보는 주어진 문구 및 주어진 문서에 대해, 이 주어진 문구의 관련 문구가 주어진 문서에 존재하는지를 나타낸다. 따라서, 2개의 질의 문구를 담고 있는 질의에 대해, 제 1 질의 문구에 대한 포스팅 리스트는 제 1 질의 문구를 담고 있는 문서를 식별하도록 처리되고 나서, 관련 문구 정보는 이 문서들 중 어느 것이 또한 제 2 질의 문구를 담고 있는지를 식별하도록 처리된다. 그리고 나서 이 후자의 문서가 서치 결과에 포함된다. 이는 시스템이 제 2 질의 문구의 포스팅 리스트를 개별적으로 처리할 필요성을 해소함으로써 보다 빠른 서치 시간을 제공한다. 물론, 이 방법은 질의 내의 많은 문구로 확장되어 상당한 계산적 시간을 절약할 수 있

다.

[0027] 시스템은 또한 문구 및 관련 문구 정보를 사용하여 한 세트의 서치 결과에서 문서를 랭킹하는데 적합할 수 있다. 주어진 문구의 관련 문구 정보는 바람직하게는 비트 벡터와 같은 포맷으로 저장되고, 이는 주어진 문구에 대하여 각 관련 문구의 상대적 중요도를 나타낸다. 예를 들면, 관련 문구 비트 벡터는 주어진 문구의 관련 문구마다 비트를 갖고, 비트는 관련 문구에 대한 예측 측도(예컨대, 정보 이득)에 따라 배열된다. 관련 문구 비트 벡터의 최상위 비트는 가장 높은 예측 측도를 갖는 관련 문구와 연관되고, 최하위 비트는 가장 낮은 예측 측도를 갖는 관련 문구와 연관된다. 이러한 방식으로, 주어진 문서와 주어진 문구에 대해, 관련 문구 정보가 문서를 스코어링하는데 사용될 수 있다. (값으로서의) 비트 벡터 자체의 값은 문서 스코어로서 사용될 수 있다. 이러한 방식으로 질의 문구의 높은 순위의 관련 문구를 담고 있는 문서는 낮은 순위의 관련 문구를 갖는 것보다 더 원칙적으로 질의에 관련되어 있을 가능성이 있다. 비트 벡터 값은 또한 보다 복잡한 스코어링 함수 내의 성분으로서 사용될 수 있고, 부가적으로 가중될 수 있다. 그러면 문서는 문서 스코어에 따라 랭킹될 수 있다.

[0028] 문구 정보는 또한 유저를 위해 서치를 개인화하도록 정보 검색 시스템에서 사용될 수 있다. 유저는 예를 들면 유저가 액세스한(예컨대, 스크린 상에 보여지고, 인쇄되고, 저장된 등등) 문서로부터 도출된 문구의 컬렉션으로서 모델링된다. 보다 구체적으로, 문서가 유저에 의해 액세스된 경우, 이 문서에 존재하는 관련 문구는 유저 모델 또는 프로파일에 포함된다. 뒤이어 서치하는 동안, 유저 모델 내의 문구는 서치 질의의 문구를 필터링하고 검색된 문서의 문서 스코어를 가중시키는데 사용된다.

[0029] 문서, 예컨대 한 세트의 서치 결과에 포함된 문서의 설명을 생성하도록 정보 검색 시스템에서 문구 정보가 또한 사용될 수 있다. 서치 질의가 주어지면, 시스템은 관련 문구와 함께 질의에 존재하는 문구 및 그들의 문구 확장을 식별한다. 주어진 문서에 대해, 각 문서의 문장은 질의 문구, 관련 문구 및 문구 확장이 문장 내에 얼마나 많이 존재하는가의 카운트를 갖는다. 문서의 문장은 이들 카운트(개별적으로 또는 결합하여)에 의해 랭킹될 수 있고, 일부의 톱 랭킹 문장(예컨대, 5 문장)이 선택되어 문서 설명을 형성한다. 그러면 문서 설명은 문서가 서치 결과에 포함되는 경우 유저에게 제공될 수 있으므로, 유저는 질의에 관하여 문서를 보다 양호하게 이해할 수 있다.

[0030] 문서 설명을 생성하는 이러한 프로세스를 추가 개선함으로써, 시스템은 유저의 관심을 반영하는 개인화된 설명을 제공할 수 있다. 전과 같이, 유저 모델은 유저에게 관심을 반영하는 관련 문구를 식별하는 정보를 저장한다. 이 유저 모델은 질의 문구에 관련된 문구의 리스트와 교차하여 양 그룹에 공통하는 문구를 식별한다. 그러면 공통 세트가 관련 문구 정보에 따라 배열된다. 결과되는 세트의 관련 문구가 사용되어 각 문서에 존재하는 이들 관련 문구의 인스턴스의 수에 따라 문서의 문장을 랭킹한다. 가장 높은 수의 공통 관련 문구를 갖는 많은 문장들이 개인화된 문서 설명으로서 선택된다.

[0031] 정보 검색 시스템은 또한 문구 정보를 사용하여 문서 컬렉션을 인덱싱(크롤링)하는 동안 또는 서치 질의를 처리하는 경우 중복 문서를 식별 및 제거할 수 있다. 주어진 문서에 대해, 문서의 각 문장은 얼마나 많은 관련 문구가 문장에 존재하는지의 카운트를 갖는다. 문서의 문장은 이 카운트에 의해 랭킹될 수 있고, 많은 상위 랭킹 문장(예컨대, 5 문장)이 선택되어 문서 설명을 형성한다. 그러면 이 설명은 예컨대 문장의 스트링 또는 해시로서 문서와 연관되어 저장된다. 인덱싱하는 동안, 새로 크롤링된 문서는 동일한 방식으로 처리되어 문서 설명을 생성한다. 새로운 문서 설명은 이전의 문서 설명에 대하여 매칭(예컨대, 해싱)될 수 있고, 매치가 발견되면, 새로운 문서는 사본이다. 마찬가지로, 서치 질의의 결과를 준비하는 동안, 서치 결과 세트 내의 문서가 처리되어 중복을 제거할 수 있다.

[0032] 본 발명은 시스템 및 소프트웨어 아키텍처, 컴퓨터 프로그램 제품 및 컴퓨터 구현 방법, 및 컴퓨터 생성 유저 인터페이스와 프리젠테이션에서 부가적인 실시예를 갖는다.

[0033] 전기한 사항은 문구에 기초한 정보 검색 시스템 및 방법의 일부의 특징에 지나지 않는다. 정보 검색의 당업자는 문구 정보의 보편적 융통성이 인덱싱, 문서 주석 달기, 서치, 랭킹 및 기타 다른 분야의 문서 분석과 처리에 다양하게 사용 및 응용될 수 있다는 것을 이해할 것이다.

[0034] 본원의 도면은 단지 예시할 목적으로 본 발명의 바람직한 실시예를 나타낸다. 당업자는 본원에서 예시된 구조 및 방법의 변형 실시예가 본원에서 설명된 본 발명의 원칙을 벗어나지 않고 이용될 수 있음을 다음의 논의로부터 쉽게 이해할 것이다.

[0035] I. 시스템 개요

[0036] 도 1을 참조하면, 본 발명의 일 실시예에 따른 서치 시스템(100)의 실시예의 소프트웨어 아키텍처가 도시되어 있다. 본 실시예에서, 시스템은 인덱싱 시스템(110), 서치 시스템(120), 프리젠테이션 시스템(130) 및 프론트 엔드 서버(140)를 포함한다.

[0037] 인덱싱 시스템(110)은 다양한 웹사이트(190)와 기타 다른 문서 컬렉션에 액세스하여 문서 내의 문구를 식별하고 이들 문구에 따라 문서를 인덱싱하는 일을 담당한다. 프론트 엔드 서버(140)는 클라이언트(170)의 유저로부터 질의를 수신하고, 이 질의를 서치 시스템(120)에 제공한다. 서치 시스템(120)은 서치 질의 내의 임의의 문구를 식별하는 일을 포함하여 서치 질의(서치 결과)에 관련된 문서를 서치하고 나서, 문구의 존재를 이용하여 서치 결과 내의 문서를 랭킹하여 랭킹 순위에 영향을 주는 일을 담당한다. 서치 시스템(120)은 서치 결과를 프리젠테이션 시스템(130)에 제공한다. 프리젠테이션 시스템(130)은 거의 중복된 문서를 제거하는 일을 포함하여 서치 결과를 수정하고, 문서의 주제 설명문을 생성하고, 수정된 서치 결과를 다시 프론트 엔드 서버(140)(결과를 클라이언트(170)에게 제공함)에 제공하는 일을 담당한다. 시스템(100)은 문서에 관련된 인덱싱 정보를 저장하는 인덱스(150), 문구를 저장하는 문구 데이터 저장부(160), 및 관련 통계 정보를 더 포함한다.

[0038] 본 출원의 문맥에서, "문서"는 웹 문서, 이미지, 멀티미디어 파일, 텍스트 문서, PDF 또는 기타 다른 이미지 포맷 파일 등을 포함하여 서치 엔진에 의해 인덱싱 및 검색될 수 있는 임의 형식의 미디어인 것으로 이해된다. 문서는 하나 이상의 페이지, 파티션, 세그먼트 또는 기타 다른 구성요소를 그 내용 및 유형에 알맞게 갖는다. 마찬가지로 문서는 인터넷 상의 문서로 불리는데 일반적으로 사용되는 바와 같이 "페이지"로 불릴 수 있다. 일반적인 용어 "문서"의 사용에 의해 본 발명의 범위에 대해서 어떠한 제한도 수반되지 않는다. 서치 시스템(100)은 인터넷 및 월드 와이드 웹과 같은 큰 자료의 문서를 다루지만, 마찬가지로 도서관 또는 개인 기업의 문서 컬렉션과 같이 보다 제한된 컬렉션에 사용될 수 있다. 어느 일 문맥에서는, 문서가 전형적으로 많은 다른 컴퓨터 시스템 및 사이트를 경유하여 배포됨을 이해할 것이다. 보편성을 잃지 않고, 포맷 또는 위치(예컨대, 어느 웹사이트 또는 데이터베이스)에 관계없이 문서는 일반적으로 자료 또는 문서 컬렉션으로 일괄하여 불린다. 각 문서는 문서를 고유하게 식별하는 연관된 식별자를 갖고, 식별자는 바람직하게는 URL이지만, 다른 유형의 식별자(예컨대, 문서 번호)가 또한 사용될 수 있다. 본 명세서에서, 문서를 식별하는데 URL을 사용하는 것으로 가정한다.

[0039] II. 인덱싱 시스템

[0040] 일 실시예에서, 인덱싱 시스템(110)은 3가지 주요 기능적 동작을 제공한다: 1) 문구 및 관련 문구의 식별, 2) 문구에 대한 문서의 인덱싱, 및 3) 문구 기반의 분류의 생성 및 유지. 당업자는 인덱싱 시스템(110)이 종래의 인덱싱 기능을 지원하면서 다른 기능도 이행한다는 것을 이해할 것이며, 따라서 다른 동작들은 본원에서 더 이상 설명되지 않는다. 인덱싱 시스템(110)은 문구 데이터의 데이터 저장소(160) 및 인덱스(150)에 작용한다. 이 데이터 저장소는 아래에서 더 설명된다.

[0041] 1. 문구 식별

[0042] 인덱싱 시스템(110)의 문구 식별 동작은 문서 컬렉션에서 문서를 인덱싱하고 서치하는데 유용한 "양호(good)" 및 "불량(bad)" 문구를 식별한다. 일 관점에서, 양호 문구는 문서 컬렉션 중 특정 비율 이상의 문서에서 발생하는 경향이 있는 문구, 및/또는 그러한 문서에서 활자 태그 또는 다른 어형, 포맷, 또는 문법적인 마커에 의해 범위가 정해지는 것과 같이 구별된 모양을 갖는 것으로 표시된 것이다. 양호 문구의 다른 특징은 다른 양호 문구를 예측할 수 있고, 용어집에 나타나는 단순한 단어 시퀀스가 아닌 것이다. 예를 들어, 문구 "President of the United States"는 "George Bush" 및 "Bill Clinton"과 같은 다른 문구를 예측하는 문구이다. 그러나, "fell down the stairs" 또는 "top of the morning", "out of the blue"와 같은 관용어구와 구어체 표현은 상이하고 관련성이 없는 많은 다른 문구와 함께 나타나는 경향이 있기 때문에, 다른 문구가 예측되지 않는다. 따라서, 문구 식별 단계는 어느 문구가 양호 문구이고, 어느 문구가 불량(즉, 예측 능력이 결여된) 문구인지를 결정한다.

[0043] 도 2를 참조하면, 문구 식별 프로세스는 다음의 기능적인 단계들을 포함한다.

[0044] 200: 문구들의 빈도 및 공통 출현 통계와 함께, 가능 및 양호 문구들을 수집한다.

[0045] 202: 빈도 통계에 기초하여 가능 문구들을 양호 또는 불량 문구 중 어느 하나로 분류한다.

[0046] 204: 공통 출현 통계로부터 유도된 예측 측도에 기초하여 양호 문구 리스트를 간결화한다.

[0047] 이제 이들 단계의 각각에 대하여 보다 구체적으로 설명한다.

[0048] 제 1 단계(200)는 인덱싱 시스템(110)이 문서 컬렉션의 문서 세트들을 크롤링하는 프로세스이고, 시간이 경과하면 문서 컬렉션의 반복되는 파티션들을 생성한다. 패스마다 하나의 파티션이 처리된다. 패스마다 크롤링되는 문서 수는 바뀔 수 있고, 파티션 마다 대략 1,000,000 정도가 바람직하다. 모든 문서가 처리되거나, 다른 종료 기준(termination criteria)을 충족하기 전에는, 이미 크롤링되지 않은 문서만이 각 파티션에서 처리되는 것이 바람직하다. 사실상, 문서 컬렉션에 새로운 문서가 계속적으로 추가되므로, 크롤링은 계속된다. 다음 스텝은 크롤링된 각 문서가 인덱싱 시스템(110)에 의해 처리된다.

[0049] 문서의 단어들을 문구 윈도 길이 n으로 트래버스(traverse)하고, 여기서 n은 원하는 최대 문구 길이이다. 윈도의 길이는 일반적으로 최소 2이고, 바람직하게는 4 또는 5 어구(단어)이다. 바람직하게는, 문구는 문구 윈도 내의 모든 단어를 포함하고, "a", "the" 등과 같이, 스톱 단어(stop word)로서 다르게 특징지어질 수 있는 것을 포함한다. 문구 윈도는 라인의 끝, 단락 복귀, 활자 태그, 또는 문맥 또는 포맷 내에서 다른 변경 표시에 의하여 종료될 수 있다.

[0050] 도 3은 트래버스 동안 문서(300)의 일부를 나타낸 것으로, 단어 "stock"에서 시작하고, 우측으로 5 단어 확장된 문구 윈도(302)를 보여주고 있다. 윈도(302)의 첫번째 단어는 후보 문구 i이고, 각각의 시퀀스(i+1, i+2, i+3, i+4, 및 i+5)도 마찬가지로 후보 문구이다. 따라서, 본 예에서, 후보 문구들은 "stock", "stock dogs", "stock dogs for", "stock dogs for the", "stock dogs for the Basque", 및 "stock dogs for the Basque shepherds"이다.

[0051] 각각의 문구 윈도(302)에서, 각 후보 문구는 양호 문구 리스트(208) 또는 가능 문구 리스트(206)에 이미 존재하는지를 판단하기 위하여 차례로 체크된다. 만약, 후보 문구가 양호 문구 리스트(208) 또는 가능 문구 리스트(206)의 어디에도 존재하지 않는 경우에는, 후보는 이미 "불량"으로 결정되어 스킵된다.

[0052] 만약 후보 문구가 양호 문구 리스트(208)에 엔트리(g_j)로서 존재하면, 문구 g_j에 대한 인덱스(150) 엔트리는 상기 문서(예컨대, 그 URL 또는 다른 문서 식별자)를 포함하도록 갱신되어, 이 후보 문구 g_j가 현재 문서에 출현하고 있음을 나타낸다. 문구 g_j(또는 어구)에 대한 인덱스(150) 엔트리는 문구 g_j의 포스팅 리스트(posting list)라 한다. 포스팅 리스트는 상기 문구가 발생한 문서 리스트 d를 (그 문서 식별자, 예컨대 문서 번호 또는 대안적으로 URL에 의하여) 포함한다.

[0053] 나아가, 후술하는 바와 같이, 공통 출현 매트릭스(212)가 갱신된다. 최초 패스에서, 양호 그리고 불량 리스트는 비어 있을 것이고, 따라서, 대부분의 문구가 가능 문구 리스트(206)에 추가될 것이다.

[0054] 후보 문구가 양호 문구 리스트(208)에 존재하지 않는 경우, 이미 가능 문구 리스트(206)에 존재하지 않는 한, 가능 문구 리스트(206)에 추가된다. 가능 문구 리스트(206)의 각 엔트리(p)는 다음의 세 개의 관련 카운트를 갖는다.

[0055] P(p): 가능 문구가 출현한 문서 수

[0056] S(p): 가능 문구의 모든 인스턴스 수

[0057] M(p): 가능 문구의 관심 인스턴스 수

[0058] 가능 문구가 문서 내에서 문법적으로 또는 포맷 마커 예컨대, 볼드체, 또는 언더라인에 의하여, 또는 하이퍼링크의 앵커 택스트로서, 또는 물음표 내에서, 인접하는 콘텐트(content)로부터 구별되는 경우, 가능 문구의 인스턴스는 "관심"이다. 이러한(그리고 다른) 구별되는 외형은 다양한 HTML 마크업 언어 태그 및 문법적 마커에 의하여 표시된다. 문구가 양호 문구 리스트(208)에 존재하는 경우, 이러한 통계치는 문구를 위하여 유지된다.

[0059] 덧붙여, 다양한 리스트에서, 양호 문구에 대한 공통 출현 매트릭스(212(G))가 유지된다. 매트릭스(G)는 m x m 차원이고, 여기서 m은 양호 문구의 개수이다. 매트릭스의 각 엔트리 G(j,k)는 양호 문구의 쌍(g_j,g_k)을 나타낸다. 공통 출현 매트릭스(212)는 논리적으로 (반드시 형태적으로는 아니더라도) 현재 단어(i)를 중심으로, +/- h 단어로 확장된 제 2 윈도(304)에 대하여 양호 문구의 각 쌍(g_j,g_k)에 대한 세 개의 별개의 카운트를 유지한다. 일 실시예에서, 도 3에 도시된 바와 같이, 제 2 윈도(304)는 30 단어이다. 따라서 공통 출현 매트릭스(212)는 다음의 카운트를 유지한다.

[0060] R(j,k): 미처리 공통 출현 카운트(Raw Co-occurrence count). 문구 g_j가 문구 g_k와 함께 제 2 윈도(304)에 출현하는 횟수.

[0061] D(j,k): 이접적 관심 카운트(Disjunctive Interesting count). 문구 g_j 또는 문구 g_k 중 어느 하나가 제 2 윈도 내에서 구별된 텍스트로서 출현하는 횟수.

[0062] C(j,k): 접속적 관심 카운트(Conjunctive Interesting count): 문구 g_j 및 문구 g_k가 모두 제 2 윈도 내에서 구별된 텍스트로서 출현하는 횟수. 접속적 관심 카운트의 사용은 문구(예컨대, 저작권 통지)가 사이드 바(sidebars), 푸터(footers), 또는 헤더에 빈번하게 출현하여, 실질적으로 다른 텍스트를 예측하지 않을 환경을 피하는데 특히 유익하다.

[0063] 도 3의 예를 참조하면, 문구 "Australian Shepherd" 및 "Australian Shepard Club of America"뿐만 아니라 "stock dogs"가 양호 문구 리스트(208) 상에 있는 것으로 가정한다. 이러한 두 개의 전자의 문구는 현재 문구 "stock dogs" 주위에 제 2 윈도(304) 내에 출현한다. 그러나, 문구 "Australian Shepherd Club of America"는 웹 사이트로의 하이퍼링크(언더라인으로 나타냄)를 위한 앵커 텍스트로서 출현한다. 따라서, 쌍 {"stock dogs", "Australian Shepherd"}에 대한 미처리 공통 출현 카운트는 증가하고, {"stock dogs", "Australian Shepherd Club of America"}에 대한 미처리 출현 카운트 및 이접적 관심 카운트는 후자가 구별된 텍스트로서 출현하기 때문에 둘다 증가한다.

[0064] 시퀀스 윈도(302) 및 제 2 윈도(304) 양쪽으로 각 문서를 트래버스하는 프로세스는 파티션 내의 각 문서에 대하여 반복된다.

[0065] 일단 파티션 내의 문서가 트래버스되면, 인덱싱 동작의 다음 단계는 가능 문구 리스트(206)로부터 양호 문구 리스트(208)를 갱신(202)하는 것이다. 가능 문구 리스트(206) 상의 가능 문구들(p)은, 문구의 출현 빈도와 문구가 출현하는 문서 수에 의하여, 그 문구가 의미 있는 문구로서 충분히 사용되는 것으로 나타난 경우, 양호 문구 리스트(208)로 이동된다.

[0066] 일 실시예에서, 이것은 다음과 같이 테스트된다. 가능 문구(p)는 다음의 경우 가능 문구 리스트(206)으로부터 제거되고, 양호 문구 리스트(208) 상에 위치하게 된다.

[0067] a) P(p)>10 및 S(p)>20 (문구(p)를 포함하는 문서 수가 10 이상이고, 문구(p)의 출현 수가 20 이상인 경우); 또는

[0068] b) M(p)>5 (문구(p)의 관심 인스턴스 수가 5 이상인 경우)

[0069] 이 임계치들은 파티션 내의 문서 수에 의하여 스케일링된다. 예를 들어, 만약 2,000,000 문서가 파티션 내에서 크롤링되면, 임계치들은 대략 두배가 된다. 물론, 임계치의 구체적인 값이나 임계치를 테스트하기 위한 로직은 실시예에 따라서 변경될 수 있음은 당업자에게 자명하다.

[0070] 문구(p)가 양호 문구 리스트(208)에 적합하지 않으면, 불량 문구에 해당되는지 체크한다. 문구(p)는 이하의 경우 불량 문구이다.

[0071] a) 문구를 포함하는 문서 수, P(p)<2인 경우 ; 그리고

[0072] b) 문구의 관심 인스턴스 수, M(p)=0인 경우

[0073] 이러한 조건들은 문구가 빈번하지 않으며, 중요한 내용을 나타내는데 사용되지 않음을 나타내며, 또한 다시 이러한 임계치들은 파티션 내의 문서 수에 대하여 스케일링될 수 있다.

[0074] 양호 문구 리스트(208)는 상술한 바와 같이, 복수-단어 문구에 더하여, 문구로서 개별적인 단어들을 당연히 포함함을 유의해야 한다. 이것은 문구 윈도(302) 내에서 각 첫번째 단어가 항상 후보 문구이고, 적절한 인스턴스 카운트가 축적될 것이기 때문이다. 따라서, 인덱싱 시스템(110)은 개개의 단어들(즉, 하나의 단어를 가지는 문구) 및 복합 단어 문구들을 모두 자동적으로 인덱싱할 수 있다. 또한, 양호 문구 리스트(208)는 m개의 문구의 모든 가능한 조합에 기초한 이론상의 최대값 보다 상당히 작을 것이다. 대표적인 실시예에서, 양호 문구 리스트(208)는 대략 6.5×10^5 문구를 포함할 수 있다. 시스템은 가능 및 양호 문구들만을 계속 파악할 필요가 있기 때문에, 불량 문구들의 리스트는 저장할 필요가 없다.

[0075] 문서 컬렉션을 통한 최종 패스에 의하여, 큰 자료 내의 문구들의 사용의 예측 분포로 인하여 가능 문구들의 리스트는 상대적으로 짧을 것이다. 따라서, 10번째 패스(예를 들면, 10,000,000 문서)에 의하여, 문구가 바로 그 최초에 출현한다고 하면, 문구는 그 때 양호 문구일 가능성은 매우 적다. 그것은 막 사용된 새로운 문구일 것이고, 따라서 다음의 크롤링 동안 점차 일반적으로 된다. 이 경우, 그것의 각 카운트는 증가할 것이고, 결국에

는 양호 문구가 되기 위한 임계치를 충족할 것이다.

[0076] 인덱싱 동작의 세번째 단계는 공통 출현 매트릭스(212)로부터 도출된 예측 측도를 이용하여 양호 문구 리스트(208)를 간결화하기 위한 것이다(204). 간결화 동작 없이는, 양호 문구 리스트(208)는 용어집에 정규적으로 출현하는 동안 다른 문구들의 존재를 충분하게 예측할 수 없거나, 더 긴 문구 열이 되는 많은 문구들을 포함하게 될 것이다. 이러한 불충분한 양호 문구들을 제거함으로써, 양호 문구들의 매우 강화된다. 양호 문구들을 식별하기 위하여 한 문구가 다른 문구가 존재하는 주어진 하나의 문서에 출현하는 가능성의 증가를 표현하는 예측 측도가 사용된다. 이것은 일 실시예에서 다음과 같이 수행된다.

[0077] 상술한 바와 같이, 공통 출현 매트릭스(212)는 양호 문구들에 관련된 데이터를 저장하는 m × m 매트릭스이다. 매트릭스의 각 열(j)은 양호 문구 g_j를 나타내고, 각 행(k)은 양호 문구 g_k를 나타낸다. 각각의 양호 문구 g_j에 대하여, 예측 값 $E(g_j)$가 계산된다. 예측 값 E는 문구 g_j를 포함할 것으로 예측되는 컬렉션 중의 문서의 비율이다. 예컨대, 이는 문구 g_j를 포함하는 문서 수와 크롤링된 문서 컬렉션의 총 수(T)에 대한 비율로 계산된다. 즉, P(j)/T이다.

[0078] 상술한 바와 같이, 문구 g_j를 포함하는 문서 수는 문구 g_j가 문서에 출현할 때마다 갱신된다. E(gj)에 대한 값은 문구 g_j에 대한 카운트가 증가할 때마다, 또는, 이 세번째 단계 동안 갱신될 수 있다.

[0079] 다음으로, 각 다른 양호 문구(예를 들어, 매트릭스의 행들)에 대하여, 문구 g_j는 문구 g_k를 예측하는지의 여부를 판단한다. 문구 g_j에 대한 예측 측도는 다음과 같이 정해진다.

[0080] i) 예측 값 $E(g_k)$를 계산한다. 문구 g_j 및 문구 g_k가 서로 관련되지 않은 문구인 경우에는, 예측 공통 출현율 E(j,k)는 $E(g_j)*E(g_k)$이다.

[0081] ii) 문구 g_j 및 문구 g_k의 실질 공통 출현율 A(j,k)를 계산한다. 이는 문서 전체 수인 T에 의하여 나누어진 미처리 공통 출현 카운트 R(j,k)이다.

[0082] iii) 실질 공통 출현율 A(j,k)가 예측 공통 출현율 E(j,k)를 임계량 만큼 초과하는 경우, 문구 g_j는 문구 g_k를 예측한다고 한다.

[0083] 일 실시예에서, 예측 측도는 정보 이득이다. 따라서, 문구 g_j의 존재에서 문구 g_k의 정보 이득(I)이 임계치를 초과하는 경우, 문구 g_j는 다른 문구 g_k를 예측한다. 일 실시예에서, 이것은 다음과 같이 계산된다.

[0084] I(j,k)=A(j,k)/E(j,k)

[0085] 그리고, 다음의 경우에 양호 문구 g_j는 양호 문구 g_k를 예측한다.

[0086] I(j,k) > 정보 이득 임계치

[0087] 일 실시예에서, 이득 정보 임계치는 1.5이나, 1.1 과 1.7 사이가 바람직하다. 임계치를 1.0 이상으로 올리는 것은 두 개의 다른 관련되지 않은 문구가 임의적으로 예측된 것보다 더 공통 출현할 가능성을 감소시킨다.

[0088] 언급한 바와 같이, 정보 이득의 계산은 주어진 열(j)에 대하여 매트릭스(G)의 각 행(k)에서 반복된다. 일단 하나의 열이 완료되었을 때, 양호 문구들 g_k 어떤 것의 정보 이득도 정보 이득 임계치를 넘지 않는 경우에는, 이는 문구 g_j가 어떠한 다른 양호 문구도 예측하지 않는 것을 의미한다. 이 경우, 문구 g_j는 양호 문구 리스트(208)에서 제거되고, 본질적으로 불량 문구로 된다. 문구 g_j에 대한 행(j)은, 이 문구 자체가 다른 양호 문구들에 의하여 예측될 수도 있기 때문에 제거되지 않음을 유의해야 한다.

[0089] 이 스텝은 공통 출현 매트릭스(212)의 모든 열이 평가된 이후에 끝난다.

[0090] 이 과정의 마지막 스텝은 양호 문구 리스트(208)를 간결화하여 불완전한 문구를 제거하는 것이다. 불완전한 문구는 그 문구의 확장만을 예측하는 문구와, 문구의 가장 좌측(즉, 문구의 시작)에서 시작하는 문구이다. 문구(p)의 "문구 확장"은 문구(p)로 시작하는 수퍼-시퀀스이다. 예를 들어, 문구 "President of"는 "President of the United States", "President of Mexico", "President of AT&T" 등을 예측한다. 이러한 후자 문구들은 모두 "President of"로 시작되므로 문구 "President of"의 문구 확장이고, 그것의 수퍼-시퀀스이다.

[0091] 따라서, 양호 문구 리스트(208)에 남아있는 각 문구 g_j는 상술한 정보 이득 임계치에 기초하여 몇 개의 다른 문구들을 예측할 것이다. 이제, 각 문구 g_j에 대하여, 인덱싱 시스템(110)은 예측된 각 문구 g_k로 스트링 매치를 수행한다. 스트링 매치는 예측된 각 문구 g_k가 문구 g_j의 문구 확장인지를 검사한다. 만약, 예측된 모든 문구 g_k가 문구 g_j의 문구 확장인 경우, 문구 g_j는 불완전하고, 양호 문구 리스트(208)에서 삭제되고, 불완전 문구 리스트(216)에 추가된다. 따라서, 만약 문구 g_j의 확장이 아닌 적어도 하나의 문구 g_k가 존재한다면, 문구 g_j는 완전하고, 양호 문구 리스트(208)에 유지된다. 예를 들어, "President of the United"는 예측하는 다른 문구가 오직 그 문구의 확장인 "President of the United States"이므로, 불완전 문구이다.

[0092] 불완전 문구 리스트(216) 자체는 실제 서치에 있어서 매우 유용하다. 서치 질의(query)가 수신된 경우, 불완전 문구 리스트(216)와 대조하여 비교될 수 있다. 만약, 질의어(또는 그 일부)가 이 리스트의 엔트리에 매치되면, 그 서치 시스템(120)은 불완전 문구의 가장 가능성 있는 문구 확장(불완전 문구에 주어진 최고 정보 이득을 갖는 문구 확장)을 찾을 수 있고, 이 문구 확장을 유저에게 제안할 수 있거나, 문구 확장에 대해 자동적으로 서치할 수 있다. 예를 들어, 만약 서치 질의가 "President of the United"인 경우, 서치 시스템(120)은 자동적으로 유저에게 서치 질의로서 "President of the United States"를 제안할 수 있다.

[0093] 인덱싱 프로세스의 마지막 단계가 완료된 이후에, 양호 문구 리스트(208)는 자료에서 발견된 많은 수의 양호 문구를 포함할 것이다. 이러한 양호 문구 각각은 문구 확장이 아닌 적어도 하나의 다른 문구를 예측할 것이다. 즉, 각각의 양호 문구는 자료에서 표현된 의미있는 개념이나 아이디어를 표현하는데 충분한 빈도와 독립성을 가지고 사용된다. 미리 정해지거나 손으로 선택된 문구들을 사용하는 현재의 시스템과 달리, 양호 문구 리스트는 자료에 사용되는 실제의 문구들을 반영한다. 나아가, 상기의 크롤링 및 인덱싱 프로세스는 새로운 문서가 문서 컬렉션에 추가될 때마다 주기적으로 반복되기 때문에, 인덱싱 시스템(110)은 새로운 문구들이 용어집에 들어갈 때, 새로운 문구들을 자동적으로 탐색한다.

[0094] 2. 관련 문구 및 관련 문구 클러스터 식별

[0095] 도 4를 참조하면, 관련 문구 식별 프로세스는 다음과 같은 기능적인 동작들을 포함한다.

[0096] 400: 높은 정보 이득 값을 갖는 관련 문구를 식별한다.

[0097] 402: 관련 문구의 클러스터를 식별한다.

[0098] 404: 클러스터 비트 벡터 및 클러스터 번호를 저장한다.

[0099] 이제, 이들 각각의 동작에 대하여 상세히 설명한다.

[0100] 먼저, 양호 문구들 g_j를 포함하는 공통 출현 매트릭스(212)를 상기한다. 여기서, 양호 문구 g_j 각각은 정보 이득 임계치보다 높은 정보 이득으로 적어도 하나의 다른 양호 문구 g_k를 예측한다. 이 후, 관련 문구를 식별(400)하기 위하여, 각각의 양호 문구 쌍(g_j,g_k)에 대한 정보 이득이 관련 문구 임계치 예컨대, 100과 비교된다. 즉, 다음의 경우 문구 g_j와 문구 g_k는 관련 문구이다.

[0101] $I(g_j,g_k)>100$

[0102] 이러한 높은 임계치는 통계적으로 충분히 예측율 이상인 양호 문구들의 공통 출현을 식별하기 위하여 사용된다. 통계적으로, 이것은 문구 g_j와 문구 g_k가 예측 공통 출현율보다 100배 이상 공통 출현하는 것을 의미한다. 예를 들어, 하나의 문서 내에서 문구 "Monica Lewinsky"가 주어지고, 문구 "Bill Clinton"이 동일 문서에 100번 이상 출현할 가능성이 있다면, 문구 "Bill Clinton"은 임의적으로 선택된 어떠한 문서에도 출현할 것이다. 이를 다르게 말하면, 출현율이 100:1이므로, 예측의 정확도가 99.999%이다는 것이다.

[0103] 따라서, 관련 문구 임계치보다 적은 어떤 엔트리(g_j,g_k)가 문구 g_j와 문구 g_k가 관련되지 않음을 나타내며, 전부 삭제된다. 이제, 공통 출현 매트릭스(212)에 남아있는 어떤 엔트리도 전부 관련 문구들임을 의미한다.

[0104] 이후, 공통 출현 매트릭스(212)의 각 열(g_j)의 행들(g_k)은 정보 이득 값 $I(g_j,g_k)$에 의해 소팅(sorting)되어, 가장 높은 정보 이득을 갖는 관련 문구 g_k가 첫번째로 리스트에 올려진다. 따라서, 이러한 소팅은 주어진 문구 g_j에 대하여, 정보 이득의 관점에서 가장 관련될 가능성이 있는 다른 문구들을 식별한다.

[0105] 다음 스텝은 어느 관련 문구들이 서로 관련 문구의 클러스터를 형성하는지를 결정(402)하는 것이다. 클러스터는 관련 문구들의 세트로서, 여기서 각각의 문구는 적어도 하나의 다른 문구에 대하여 높은 정보 이득을 갖는다. 일 실시예에서, 클러스터는 다음과 같이 식별된다.

[0106] 매트릭스의 각 열(g_j) 내에서, 문구 g_j에 관련된 하나 이상의 다른 문구들이 있을 것이다. 이 세트는 관련 문구 세트(R_j)이고, 여기서, $R=\{g_k, g_l, \ldots, g_m\}$ 이다.

[0107] 관련 문구 세트(R_j) 내에서 각 관련 문구(m)에 대하여, 인덱싱 시스템(110)은 R 내의 다른 관련 문구 각각이 문구 g_j에 또한 관련되어 있는지를 판단한다. 따라서, 만약 $I(g_k,g_l)$가 0이 아닌 경우에는, g_j, g_k, 및 g_l은 클러스터의 일부이다. 이 클러스터 검사는 R 내의 각각의 쌍(g_l,g_m)에 대하여 반복된다.

[0108] 예를 들어, "Bill Clinton"에 대한 이들 문구 각각의 정보 이득(information gain)이 관련 문구 임계치(Related Phrase threshold)를 초과하기 때문에, 양호 문구 "Bill Clinton"이 문구 "President", "Monica Lewinsky"와 관련이 있다고 가정한다. 또한, 문구 "Monica Lewinsky"가 문구 "purse designer"과 관련이 있다고 가정한다. 그러면, 이들 문구들은 세트 R을 형성한다. 클러스터를 판정하기 위하여, 인덱싱 시스템(110)은, 이들 문구의 대응하는 정보 이득을 판정함으로써, 이들 문구 각각의 다른 문구에 대한 정보 이득을 평가한다. 따라서, 인덱싱 시스템(110)은 정보 이득 I("President", "Monica Lewinsky"), I("President", "purse designer") 등과 같은, R 내의 모든 쌍에 대한 정보 이득 I를 판정한다. 이 예에서, "Bill Clinton", "President", 및 "Monica Lewinsky"가 하나의 클러스터를 형성하고, "Bill Clinton", 및 "President"가 제 2 클러스터를 형성하고, 그리고 "Monica Lewinsky", 및 "purse designer"가 제 3 클러스터를 형성하고, 그리고 "Monica Lewinsky", "Bill Clinton", 및 "purse designer"가 제 4 클러스터를 형성한다. 이것은 "Bill Clinton"은 충분한 정보 이득으로 "purse designer"를 예측하지 않는 반면에, "Monica Lewinsky"는 이들 2개의 문구 모두를 예측하기 때문이다.

[0109] 클러스터 정보를 기록(404)하기 위하여, 각 클러스터에는 고유의 클러스터 번호(클러스터 ID)가 할당된다. 그리고, 이 정보는 각 양호 문구 g_j와 관련하여 기록된다.

[0110] 일 실시예에서, 클러스터 번호는, 문구들 사이의 직교 관련성을 가리키는, 클러스터 비트 벡터에 의하여 판정된다. 상기 클러스터 비트 벡터는 양호 문구 리스트에서 양호 문구의 수인 길이 n의 일련의 비트이다. 주어진 양호 문구 g_j에 대하여, 비트 위치는 g_j의 소팅된 관련 문구들 R에 대응한다. R 내의 관련 문구 g_k가 문구 g_j와 동일한 클러스터에 속하는 경우에 비트가 세트된다. 보다 일반적으로, 이것은, g_j 와 g_k 사이에 어떠한 하나의 방향으로 정보 이득이 있는 경우에, 클러스터 비트 벡터에서 대응하는 비트가 세트되는 것을 의미한다.

[0111] 그리고 클러스터 번호는 결과적인 비트 스트링의 값이다. 이 실시예는 다수의 또는 일 방향의 정보 이득을 가지는 관련 문구들이 동일한 클러스터에 나타나는 특성을 가진다.

[0112] 상기한 문구들을 사용한 클러스터 비트 벡터의 예는 다음과 같다.

[0113]

	Bill Clinton	President	Monica Lewinsky	purse designer	클러스트 ID
Bill Clinton	1	1	1	0	14
President	1	1	0	0	12
Monica Lewinsky	1	0	1	1	11
purse designer	0	0	1	1	3

[0114] 그리고 요약하면, 이 프로세스 이후에, 각 양호 문구 g_j에 대하여, 정보 이득 $I(g_j,g_k)$가 최상위값에서 최하위값의 순서로 소팅된 관련 문구의 세트 R이, 식별될 것이다. 게다가, 각 양호 문구 g_j에 대하여, 클러스터 비트 벡터가 존재할 것인데, 그것의 값은, 문구 g_j가 멤버로 포함되어 있는 제 1 클러스터를 식별하는 클러스터 번호, 및 R 내의 관련 문구들 중에서 g_j와 동일한 클러스터에 해당되는 관련 문구를 지시하는 직교값(orthogonality value)(각 비트 위치에 대하여 1 또는 0)이다. 따라서 상기 예에서, "Bill Clinton", "President", 및 "Monica Lewinsky"는, 문구 "Bill Clinton"의 열에서의 비트 값에 기초하여, 클러스터 14에 속한다.

[0115] 이 정보를 저장하기 위하여, 2개의 기본적인 표현 방법을 이용할 수 있다. 우선, 상기한 바와 같이, 상기 정보는 공통 출현 매트릭스(212)에 저장될 수 있는데, 여기서:

[0116] 엔트리 G[열 j, 행 k] = (I(j,k), 클러스터 번호, 클러스터 비트 벡터)

[0117] 이다.

[0118] 대신에, 상기 매트릭스 표현 방법을 사용하지 않고, 양호 문구 리스트(208)에 모든 정보가 저장될 수 있는데, 여기서 그것의 각 열은 양호 문구 g_j를 나타낸다:

[0119] 문구 열$_j$ = 리스트 [문구 g_k, (I(j,k), 클러스터 번호, 클러스터 비트 벡터)]

[0120] 이와 같은 접근법에 의하면 클러스터를 유용하게 조직화할 수가 있다. 우선, 이러한 접근법은, 엄격하게 -그리고 종종 임의적으로- 정의된 주제과 개념의 분류 계층이라기보다는, 관련 문구들에 의해 지시될 때, 주제는 관련성의 복잡한 그래프를 형성하는데, 여기서 어떤 문구들은 많은 다른 문구들과 관련되고, 그리고 어떤 문구들은 보다 제한된 범위를 가지며, 그리고 상기 관련성은 상호성(각 문구가 다른 문구를 예측함)을 가지거나 또는 일 방향성(어떤 문구가 다른 문구를 예측하지만, 역은 성립하지 않음)을 가진다는 것을 인식한다. 그 결과, 클러스터는 각 양호 문구에 대하여 "로컬(local)"로 특징지워 질 수 있고, 그리고 어떤 클러스터들은 하나 또는 그 이상의 공통된 관련 문구들을 가짐으로써 중첩될 수도 있다.

[0121] 주어진 양호 문구 g_j에 대하여 정보 이득에 의하여 관련 문구들을 랭킹하는 것은 상기 문구의 클러스터를 명명하기 위한 분류를 제공한다: 클러스터 이름은 클러스터에서 최고의 정보 이득을 가지는 관련 문구의 이름이다.

[0122] 상기한 프로세스는 문서 컬렉션에 나타나는 중요한 문구를 식별하기 위한 아주 강력한(robust) 방법을 제공하며, 그리고 이 방법은, 실제적으로 사용되는 자연스러운 "클러스터"에 이들 관련 문구들이 같이 사용된다는 점에서 유익한 방법이다. 그 결과, 이러한 관련 문구들의 데이터 구동 클러스터링 방법(data-driven clustering of related phrase)에 의하면, 많은 시스템에서 공통으로 사용되는 것과 같은, 관련 용어들과 개념들에 대한 어떠한 작위적인 "편집"의 선택에 내재되어 있는 편향을 회피할 수가 있다.

[0123] 3. 문구 및 관련 문구들로 문서의 인덱싱

[0124] 관련 문구 및 클러스터와 관련된 정보를 포함하는 상기 양호 문구 리스트(208)가 주어지면, 인덱싱 시스템(110)의 다음 기능적인 동작은, 양호 문구 및 클러스터에 대하여 문서 컬렉션 내의 문서들을 인덱싱하고, 그리고 인덱스(150)에 갱신된 정보를 저장하는 것이다. 도 5에는 이러한 프로세스가 도시되어 있는데, 여기에는 문서를 인덱싱하기 위한 다음과 같은 기능적인 단계가 있다:

[0125] 500 : 문서에서 발견되는 양호 문구들의 포스팅 리스트(posting list)에 문서를 포스팅.

[0126] 502 : 관련 문구 및 제 2 관련 문구에 대한 인스턴스 카운트(instance count)및 관련 문구 비트 벡터를 갱신.

[0127] 504 : 관련 문구 정보로 문서에 주석 달기.

[0128] 506 : 포스팅 리스트 크기에 따라 인덱스 엔트리를 재배열.

[0129] 이들 단계에 대하여 지금부터 보다 상세하게 설명한다.

[0130] 앞에서와 마찬가지로, 문서 세트를 트레버스하거나 크롤링한다: 이것은 동일한 문서 세트이거나 다른 문서 세트일 수 있다. 주어진 문서 d에 대하여, 상술한 것과 동일한 방법으로, 위치i로부터, 길이 n의 시퀀스 윈도(302)를 사용하여 단어별로 문서 단어를 트레버스한다(500).

[0131] 주어진 문구 윈도(302)에서, 위치i에서 시작하여, 윈도 내의 모든 양호 문구들을 식별한다. 각각의 양호 문구는 g_i 로 표시된다. 따라서, g1은 제 1 양호 문구이고, g2는 제 2 양호 문구이다.

[0132] 각 양호 문구 g_i(예를 들어, g1은 "President"이고, g4는 "President of ATT"이다)에 대하여, 인덱스(150)의 양호 문구 g_i 에 대한 포스팅 리스트에 문서 식별자(예, URL)를 포스팅한다. 이러한 갱신에 의하여, 양호 문구 g_i 가 이런 특정 문서에 나타난다는 것이 식별된다.

[0133] 일 실시예에서, 문구 g_j 에 대한 포스팅 리스트는 다음과 같은 논리적인 형태를 가진다.

[0134] 문구 g_j : 리스트: (문서 d, [리스트: 관련 문구 카운트][관련 문구 정보])

[0135] 각 문구 g_j에 대하여, 그 문구가 나타나는 문서 d의 리스트가 있다. 각 문서에 대하여, 역시 문서 d에 나타나는 문구 g_j의 관련 문구 R들의 출현 횟수를 카운트한 리스트가 있다.

[0136] 일 실시예에서, 관련 문구 정보는 관련 문구 비트 벡터이다. 이 비트 벡터는, 각 관련 문구 g_k에 대하여 2개의 비트 위치, g_k-1, g_k-2가 있다는 점에서, "2-비트" 벡터로서 특징지어질 수 있다. 제 1 비트 위치는 관련 문구 g_k가 문서 d(즉, 문서 d에서 g_k 에 대한 카운트가 0보다 크다)에 존재하는지의 여부를 나타내는 플래그(flag)를 저장한다. 제 2 비트 위치는 문구 g_k 의 관련 문구 g_l이 문서 d에 존재하는지의 여부를 나타내는 플래그를 저장한다. 문구 g_j의 관련 문구 g_k의 관련 문구 g_l은 여기서는 "g_j의 제 2 관련 문구"로 칭해진다. 카운트와 비트 위치는 (정보 이득이 감소하는 순서로 소팅된) R에서 문구들의 표준적인(canonical) 순서에 대응한다. 이러한 소팅 순서는 g_j에 의하여 가장 높이 예측되는 관련 문구 g_k가 관련 문구 비트 벡터의 최상위 비트에 연관되도록 하고, 그리고 g_j에 의하여 가장 낮게 예측되는 관련 문구 g_l가 최하위 비트와 연관되도록 하는 효과가 있다.

[0137] 주어진 문구 g에 대하여, 관련 문구 비트 벡터의 길이, 및 그 벡터의 각각의 비트에 대한 관련 문구들의 연관성은, 문구 g를 포함하고 있는 모든 문서에 대하여 동일할 것이라는 것에 주의하는 것은 유용하다. 이 실시예는 시스템이, 문구g를 포함하고 있는 어떤 (또는 모든) 문서에 대한 관련 문구 비트 벡터를, 쉽게 비교하여, 어떤 문서가 주어진 관련 문구를 가지고 있는지를 알아볼 수 있게 해주는 특징이 있다. 이것은 어떤 서치 질의에 대한 응답으로 문서를 식별하는 서치 프로세스를 용이하게 해주는 이점이 있다. 따라서, 주어진 문서는 많은 다른 문구의 포스팅 리스트에 나타날 것이며, 그리고 각각의 그러한 포스팅 리스트에는, 그 문서에 대한 그 관련 문구 벡터가 상기 포스팅 리스트를 가지는 문구에 특유할 것이다. 이러한 측면은 개별 문구 및 문서에 대한 관련 문구 비트 벡터의 국소성(locality)을 유지시킨다.

[0138] 따라서, 다음 단계(502)는 문서에서 현재의 인덱스 위치의 제 2 윈도(304)(이전과 같이 +/- K 용어, 예를 들어, 30 용어의 제 2 윈도), 예컨대, i-K부터 i+K까지를 트래버스하는 것을 포함한다. 제 2 윈도(304)에 나타나는 g_i의 관련 문구 g_k 각각에 대하여, 인덱싱 시스템(110)은 관련 문구 카운트에서 문서 d에 대한 g_k의 카운트를 증가시킨다. g_i 가 문서에서 뒤에 나타나고, 그리고 그 관련 문구가 그 뒤쪽의 제 2 윈도에 다시 발견되는 경우에는, 카운트는 다시 증가한다.

[0139] 언급된 바와 같이, 관련 문구 비트 맵에서 대응하는 제 1 비트 g_k-1는 상기 카운트에 기초하여 세트되는데, g_k에 대한 카운트가 0보다 큰 경우에는 그 비트는 1로 세트되고, 또는 상기 카운트가 0인 경우에는 0으로 세트된다.

[0140] 다음으로, 인덱스(150)에서 관련 문구 g_k를 찾아보고, g_k의 포스팅 리스트에서 문서 d에 대한 엔트리를 식별하고, 다음으로 제 2 관련 문구 카운트(또는 비트)를 임의의 그것의 관련 문구에 대해 체크함으로써, 제 2 비트 g_k-2가 세트된다. 만일 이들 제 2 관련 문구 카운트/비트 중 어느 것이든지 세트되면, 이것은 g_j의 제 2 관련 문구가 문서 d에 역시 존재한다는 것을 지시한다.

[0141] 문서 d가 이러한 방식으로 완전히 처리되는 경우에, 인덱싱 시스템(110)은 다음의 사항들을 식별할 것이다:

[0142] i) 문서 d에서 각 양호 문구g_j;

[0143] ii) 각 양호 문구 g_j에 대하여, 그것의 관련 문구 g_k 중에서 어떠한 것이 문서 d에 존재하는지;

[0144] iii) 문서 d에 존재하는 각 관련 문구 g_k에 대하여, 그것의 관련 문구 g_l(g_j의 제 2 관련 문구)중에서 어떠한 것이 또한 문서 d에 존재하는지.

[0145] a) 문서에 대한 주제를 판정하기

[0146] 문구에 의하여 문서를 인덱싱하고 클러스터링 정보를 사용하는 것은 인덱싱 시스템(110)의 다른 하나의 이점을 제공하는데, 이것은 관련 문구 정보에 기초하여 문서의 주제를 판정하는 능력이다.

[0147] 주어진 양호 문구 g_j 및 주어진 문서 d에 대하여, 포스팅 리스트 엔트리가 다음과 같다고 가정하자:

[0148] g_j : 문서 d : 관련 문구 카운트: = {3,4,3,0,0,2,1,1,0}

[0149] 관련 문구 비트 벡터: = {11 11 10 00 00 10 10 10 01}

[0150] 여기서, 관련 문구 비트 벡터는 2비트 쌍(bi-bit pairs}으로 나타나 있다.

[0151] 관련 문구 비트 벡터로부터, 문서 d의 제 1 및 제 2 주제를 판정할 수 있다. 제 1 주제는 비트 쌍 (1,1)으로 지시되며, 그리고 제 2 주제는 비트 쌍 (1,0)으로 지시된다. 관련 문구 비트 쌍 (1,1)은, 제 2 관련 문구g_l 와 함께 비트 쌍에 대한 관련 문구 g_k 둘 다 문서 d에 존재한다는 것을 지시한다. 이것은 문서 d의 저자가 그 문서를 작성할 때에 몇 개의 관련 문구g_j, g_k, 및 g_l을 함께 사용한 것으로 해석할 수 있다. 비트 쌍 (1,0)은 g_j 및 g_k 는 둘 다 존재하지만, g_k로부터의 제 2 관련 문구는 더 이상 존재하지 않는다는 것을 지시하며, 따라서 이것은 덜 중요한 주제이다.

[0152] b) 개선된 랭킹을 위한 문서 주석

[0153] 인덱싱 시스템(110)의 다른 하나의 측면은, 후속 서치 프로세스에서 개선된 랭킹을 제공하는 정보를 가지고 인덱싱 프로세스 중에 각 문서 d에 주석을 다는(504) 능력이다. 주석을 다는 프로세스(506)는 다음과 같다.

[0154] 문서 컬렉션 내의 주어진 문서 d는 다른 문서에 대하여 얼마의 아웃링크(outlink)를 가질 수 있다. 각 아웃링크(하이퍼링크)는 앵커 텍스트(anchor text) 및 타깃 문서의 문서 식별자를 포함한다. 설명의 편의를 위하여, 현재 처리가 이루어지고 있는 문서 d는 URL0으로 하고, 문서 d의 아웃링크인 타깃 문서는 URL1이라고 하자. 어떤 다른 URLi를 가리키는, URL0의 각 링크에 대해, 서치 결과에 나타나는 랭킹 문서의 추후 사용을 위하여, 인덱싱 시스템(110)은, URL0에 대한 그 링크의 앵커 문구에 대한 아웃링크 스코어, 및 URLi에 대한 그 앵커 문구에 대한 인링크(inlink) 스코어를 생성한다. 다시 말해서, 문서 컬렉션에서 각 링크는 아웃링크 스코어와 인링크 스코어로 이루어진 한 쌍의 스코어를 가진다. 이들 스코어는 다음과 같이 계산된다.

[0155] 주어진 문서 URL0에 대하여, 인덱싱 시스템(110)은 다른 문서 URL1에 대한 각 아웃링크를 식별하는데, 여기서 앵커 텍스트 A는 양호 문구 리스트(208)에 있는 문구이다. 도 8의 (a)는 이러한 관련성을 도식적으로 보여주고 있는데, 여기서 문서 URL0의 앵커 텍스트 "A"는 하이퍼링크(800)로 사용된다.

[0156] 문구 A에 대한 포스팅 리스트에서, URL0은 문구 A의 아웃링크로서 포스팅되고, URL1은 문구 A의 인링크로서 포스팅된다. URL0의 경우에, 관련 문구 비트 벡터는 상술한 바와 같이 완성되어, URL0에 존재하는 A의 관련 문구 및 제 2 관련 문구를 식별한다. 이러한 관련 문구 비트 벡터는 앵커 문구 A를 포함하고 있는 URL0으로부터 URL1로의 링크에 대한 아웃링크 스코어로서 사용된다.

[0157] 다음으로, 인링크 스코어는 다음과 같이 판정된다. 앵커 문구 A를 포함하는 URL1에 대한 각 인링크에 대하여, 인덱싱 시스템(110)은 URL1를 스캐닝하고, 그리고 URL1의 본문(body)에 문구 A가 나타나는지의 여부를 판정한다. 만일 문구 A가 (URL0에 대한 아웃링크를 경유하여) URL1를 지시할 뿐만 아니라 URL1의 내용에 나타난다면, 이것은 URL1이 문구 A에 의하여 나타난 개념에 의식적으로 관련되어 있다고 말할 수 있음을 암시하고 있다. 도 8의 (b)는 이러한 경우를 나타내는데, 여기서 문구 A는 (앵커 텍스트로서) URL0과 URL1의 본문에 둘 다 나타난다. 이 경우에, URL1에 대한 문구 A의 관련 문구 비트 벡터는, URL0으로부터 문구 A를 포함하는 URL1로의 링크에 대한 인링크 스코어로서 사용된다.

[0158] 만일 (도 8의 (a)에 도시된 바와 같이) 앵커 문구 A가 URL1의 본문에 나타나지 않는 경우에는, 인링크 스코어를 판정하기 위하여 다른 단계가 사용된다. 이 경우에, 인덱싱 시스템(110)은 (마치 문구 A가 URL1에 존재한 것과 같이) 문구 A에 대한 URL1의 관련 문구 비터 벡터를 생성하여 문구 A의 어떠한 관련 문구가 URL1에 나타나는지를 지시한다. 이러한 관련 문구 비트 벡터는 URL0으로부터 URL1으로의 링크에 대한 인링크 스코어로서 사용된다.

[0159] 예를 들어, 최초에 다음과 같은 문구들이 URL0과 URL1에 존재한다고 가정해보자.

[0160]

문서	앵커 문구	관련 문구 비트 벡터				
	Australian Shepherd	Aussie	blue merle	red merle	tricolor	agility training
URL0	1	1	0	0	0	0
URL1	1	0	1	1	1	0

[0161] (위의 표와 다음의 표에는 제 2 관련 문구 비트는 표시되어 있지 않다). URL0열은 앵커 텍스터A로부터의 아웃링크 스코어이고, 그리고 URL1열은 상기 링크의 인링크 스코어이다. 여기서, URL0은 URL1을 타깃으로 하는 "Australian Shepahrd"를 포함한다. "Australian Shephard"의 다섯 개의 관련 문구들 중에서, 단 하나의 관련 문구, "Aussie"만이 URL0에 나타난다. 그러면 직관적으로, URL0은 Anstralian Shepherd에 대하여 아주 약하게 관련되어 있다는 것을 알 수 있다. URL1은, 대조적으로, 문서의 본문에 "Australian Shephard"가 존재할 뿐만 아니라, "blue merle", "red merle", 및 "tricolor"와 같은 많은 관련 문구를 포함하고 있다. 따라서, 앵커 문구인 "Australian Shepard"는 URL0과 URL1에 모두 나타나기 때문에, URL0의 아웃링크 스코어와 URL1의 인링크 스코어는 상기 표의 각각의 열에 나타내져 있는 것과 같다.

[0162] 상술한 두 번째 경우는 앵커 문구 A가 URL1에 나타나지 않는 경우이다. 이 경우에, 인덱싱 시스템(110)은 URL1을 스캐닝하고, 그리고 관련 문구들 즉, "Aussie", "blue merle", "red merle", "tricolor", 및 "agility training" 중에서 어떤 것이 URL1에 존재하는지를 판정하여, 예를 들어 다음과 같은 관련 문구 비트 벡터를 생성한다:

[0163]

문서	앵커 문구	관련 문구 비트 벡터				
	Australian Shepherd	Aussie	blue merle	red merle	tricolor	agility training
URL0	1	1	0	0	0	0
URL1	0	0	1	1	1	0

[0164] 여기서, 상기 표는 URL1은 앵커 문구인 "Australian Shepard"는 포함하고 있지 않지만, 관련 문구들인 "blue merle", "red merle", 및 "tricolor"는 포함하고 있다는 것을 보여준다.

[0165] 이러한 접근법에 의하면, 서치의 결과를 왜곡하기 위하여 (문서의 일부인) 웹 페이지를 어떠한 방식으로 조작하는 것을 완전히 방지할 수 있는 이점이 있다. 그 문서를 랭킹하기 위하여 주어진 문서를 지시하는 링크의 수에 의존하는 랭킹 알고리즘을 사용하는 서치 엔진은, 소망하는 페이지를 지시하는 주어진 앵커 텍스트를 가지고 많은 수의 페이지들을 인위적으로 생성시키는 것에 의하여 "폭격"될 수 있다. 그 결과, 앵커 텍스트를 사용하는 서치 질의가 입력되었을 경우에, 실제 이 페이지가 앵커 텍스트와 거의 관련이 없거나 아무런 관련이 없는 경우에도, 상기 소망하는 페이지가 대표적으로 찾아진다. 타깃 문서 URL1로부터 문서 URL0에 대한 문구 A의 관련 문구 비트 벡터로 관련 문구 벡터를 들여오게 되면, 서치 시스템이, 앵커 텍스트 문구에 대한 중요도 또는 URL1를 지시하는 것으로서, URL0에서 문구 A의 관련성에만 의존하는 것을 없앨 수가 있다.

[0166] 또한, 자료에서 나타나는 빈도에 기초하여, 인덱스(150)의 각 문구에는 문구 번호가 부여된다. 그 문구가 더 공통적이면 공통적일수록, 인덱스에서는 더 작은 문구 번호가 주어진다. 그러면 인덱싱 시스템(110)은, 각 포스팅 리스트의 문구 번호를 리스팅하는 문서의 수에 따라서, 인덱스(150)에서 모든 포스팅 리스트를 내림차순으로 소팅(506)하여, 그 결과 가장 빈번하게 나타나는 문구가 가장 앞에 포스팅된다. 그러면 그 문구 번호는 특정한 문구를 찾아보는데 사용될 수 있다.

[0167] III. 서치 시스템

[0168] 서치 시스템(120)은 질의를 받아서 그 질의에 관련된 문서를 찾고, 그리고 서치 결과의 세트에 (그 문서에 대한 링크를 가지는) 이들 문서의 리스트를 제공하도록 동작한다. 도 6에는 상기 서치 시스템(120)의 주요한 기능적인 동작이 도시되어 있다.

[0169] 600 : 질의에 대한 문구를 식별

[0170] 602 : 질의 문구에 관련된 문서를 검색

[0171] 604 : 문구에 따라 서치 결과 내의 문서들을 랭킹

[0172] 이들 단계들 각각에 대한 자세한 설명은 다음과 같다.

[0173] 1. 질의 및 질의 확장에서 문구들을 식별

[0174] 서치 시스템(120)의 제 1 단계(600)는 인덱스를 효과적으로 서치하기 위하여 질의에 존재하는 문구들을 식별하는 것이다. 이 섹션에서는 다음과 같은 용어들이 사용된다:

[0175] q : 입력으로서 서치 시스템(120)에 의해 수신되는 질의.

[0176] Qp : 질의에 존재하는 문구들.

[0177] Qr: Qp의 관련 문구

[0178] Qe: Qp의 문구 확장

[0179] Q: Qp와 Qr의 병합

[0180] 질의 q는 클라이언트(190)로부터 수신되며, 문자 또는 단어의 어떤 최대수까지 갖는다.

[0181] 서치 시스템(120)은 사이즈 N(예를 들면 5)의 문구 윈도를 사용하여 질의 q의 용어들을 트래버스한다. 문구 윈도는 질의의 제 1 용어부터 시작하여 우측으로 N개의 용어들을 확장한다. 그 후, 이 윈도는 M-N회 우측으로 시프트되고, 여기서 M은 질의 내의 용어 수이다.

[0182] 각 윈도 위치에서는, 윈도 내의 N개(또는 소수)의 용어가 있을 수 있다. 이들 용어들은 가능 질의 문구를 이룬다. 가능 문구를 양호 문구 리스트(208) 내에서 찾아내어, 양호 문구인지의 여부를 결정한다. 가능 문구가 양호 문구 리스트(208) 내에 존재하면, 문구 번호가 문구를 대신하여 불러오게 된다. 이 가능 문구가 현재 후보 문구이다.

[0183] 각 윈도 내의 모든 가능 문구를 테스트하여 양호 후보 문구인지를 결정한 후에, 서치 시스템(120)은 질의 내의 대응하는 문구에 대하여 문구 번호들의 세트를 가질 것이다. 그 후, 이들 문구 번호들이 소팅된다(내림차순).

[0184] 서치 시스템(120)은 1차 후보 문구로서 가장 높은 문구 번호부터 시작하여, 소팅된 리스트 내에서 일정한 수치 간격 내에, 즉 문구 번호 간의 차가 임계량, 예를 들면 20000 이내인 다른 후보 문구가 있는지를 결정한다. 후보 문구가 있으면, 질의 내의 가장 좌측에 있는 문구를 유효한 질의 문구 Qp로서 선택한다. 이 질의 문구 및 그의 모든 서브문구를 후보 리스트로부터 제거하고, 리스트를 재소팅하여 프로세스를 반복한다. 이 프로세스의 결과가 유효 질의 문구 Qp의 세트이다.

[0185] 예를 들면, 서치 질의가 "Hillary Rodham Clinton Bill on the Senate Floor"라고 가정한다. 서치 시스템(120)은 다음과 같은 후보 문구, "Hillary Rodham Clinton Bill on", "Hillary Rodham Clinton Bill", 및 "Hillary Rodham Clinton"을 식별하게 된다. 처음 2개는 버리고, 마지막 1개는 유효 질의 문구로서 유지된다. 다음에, 서치 시스템(120)은 "Bill on the Senate Floor", 및 서브 문구 "Bill on the Senate", "Bill on the", "Bill on", "Bill"을 식별하여, "Bill"을 유효 질의 문구 Qp로서 선택하게 된다. 마지막으로, 서치 시스템(120)은 "on the senate floor"를 분석하여 "Senate Floor"를 유효 질의 문구로서 식별했을 것이다.

[0186] 다음에, 서치 시스템(120)은 대문자에 대한 유효 문구 Qp를 조정한다. 질의를 분석할 때, 서치 시스템(120)은 각 유효 문구 내에 잠재적인 대문자를 식별한다. 이는 "United States"로서 대문자화된 "united states" 등의 공지된 대문자 테이블을 사용하거나, 문법 기반 대문자 알고리즘을 사용하여 행해질 수 있다. 이는 적절히 대문자화된 질의 문구의 세트를 생성한다.

[0187] 그 후, 서치 시스템(120)은 대문자화된 문구를 제 2 패스시켜 문구와 그의 서브 문구가 모두 세트 내에 존재하는 경우에 가장 좌측에 있는 문구만을 선택하여 대문자화한다. 예를 들면, "president of the united states"의 서치는 "President of the United States"로 대문자화될 수 있다.

[0188] 다음 스테이지에서, 서치 시스템(120)은 질의 문구 Q와 관련된 문서들을 식별한다(602). 그 후, 서치 시스템(120)은 질의 문구 Q의 포스팅 리스트를 검색하고, 이들 리스트를 교차(intersect)시켜서 문구에 대한 포스팅 리스트의 전부(또는 일부)에 어떤 문서가 나타날 것인지를 판정한다. 질의 내의 문구 Q가 문구 확장 Qe(후술하는 바와 같이)의 세트를 갖는 경우, 서치 시스템(120)은 우선 포스팅 리스트와의 교차를 행하기 전에 문구 확장의 포스팅 리스트를 병합한다. 서치 시스템(120)은 상기한 바와 같이 불완전한 문구 리스트(216) 내에서 각 질의 문구 Q를 찾아내서 문구 확장을 식별한다.

[0189] 교차의 결과는 질의와 관련된 문서의 세트이다. 문구 및 이와 관련된 문구에 의해 문서를 인덱싱하고, 질의 내에서 문구 Q를 식별하고, 그 후 질의와 더욱 관련있는 문서의 세트의 선택 시 문구 확장 결과를 포함하도록 질의를 확대하면, 질의 용어를 포함하는 문서만을 선택하는 종래의 부울대수 기반 서치 시스템과 마찬가지일 것이다.

[0190] 일 실시예에서, 서치 시스템(120)은 최적화된 메카니즘을 사용하여, 질의 문구 Q의 모든 포스팅 리스트를 교차시킬 필요없이 질의에 따른 문서를 식별한다. 인덱스(150) 구조의 결과, 각 문구 g_j마다, 관련 문구 g_k는 공지

되어 있고, 이는 g_k에 대한 관련 문구 비트 벡터로 식별된다. 따라서, 이 정보는 2개 이상의 질의 문구가 서로 관련된 문구이거나, 또는 공통의 관련 문구를 갖는 교차 프로세스를 간단히 하는데 사용될 수 있다. 이 경우, 관련 문구 비트 벡터를 직접 액세스한 다음에 대응하는 문서를 검색하는데 사용할 수 있다. 이 프로세스는 다음과 같이 더욱 충분하게 설명된다.

[0191] 임의의 2개의 질의 문구 Q1 및 Q2가 주어지면, 3가지 가능한 경우의 관계가 있다.

[0192] 1) Q2는 Q1의 관련 문구이다.

[0193] 2) Q2는 Q1의 관련 문구가 아니고 그들 각각의 관련 문구 Qr1 및 Qr2는 교차하지 않는다(즉, 공통의 관련 문구가 없다).

[0194] 3) Q2는 Q1의 관련 문구가 아니지만, 그들 각각의 관련 문구 Qr1 및 Qr2는 교차한다.

[0195] 질의 문구의 각 쌍마다 서치 시스템(120)은 질의 문구 Qp의 관련 문구 비트 벡터를 찾아내서 적합한 경우를 판정한다.

[0196] 서치 시스템(120)은 Q1을 포함하는 문서와, 이들 문서 각각에 대하여 관련 문구 비트 벡터를 포함하는 질의 문구 Q1에 대한 포스팅 리스트를 검색한다. Q1의 관련 문구 비트 벡터는 문구 Q2(및 만일 있다면, 각각의 남아있는 질의 문구)가 Q1의 관련 문구이고 문서 내에 존재하는지의 여부를 표시할 것이다.

[0197] Q2에 제 1의 경우가 적용되면, 서치 시스템(120)은 Q1의 포스팅 리스트 내의 각 문서 d마다 관련 문구 비트 벡터를 스캐닝하여 Q2에 대한 비트를 설정했는지를 판정한다. 이 비트가 Q1의 포스팅 리스트 내의 각 문서 d에 대해 설정되어 있지 않다면, Q2는 그 문서 내에 나타나지 않은 것을 의미한다. 그 결과, 이 문서는 더이상의 고려 대상으로부터 즉시 소거될 수 있다. 그 후, 나머지 문서가 스코어링될 수 있다. 이는 또한 서치 시스템(120)이 Q2의 포스팅 리스트 내에 어떤 문서가 존재하는지 확인하는 처리가 필요 없어, 연산 시간을 절약할 수 있다는 것을 또한 의미한다.

[0198] Q2에 제 2의 경우를 적용하면, 2개의 문구는 서로 관련이 없다. 예를 들면, 질의 "cheap bolt action rifle"은 2개의 문구 "cheap" 및 "bolt action rifle"를 갖는다. 이들 문구는 서로 관련이 없고, 또한 이들의 각 관련 문구는 중복되지 않는다, 즉, "cheap"은 관련 문구 "low cost", "inexpensive", "discount", "bargain basement", 및 "lousy"를 갖는 반면에, "bolt action rifle"은 관련 문구 "gun", "22 caliber", "magazine fed", 및 "Armalite AR-30M"를 갖기 때문에, 리스트는 교차되지 않는다. 이 경우, 서치 시스템(120)은 Q1 및 Q2의 포스팅 리스트의 정규 교차를 행하여, 스코어링(scoring)용 문서를 얻는다.

[0199] 제 3의 경우를 적용하면, 여기서 2개의 문구 Q1 및 Q2는 관련이 없지만, 적어도 하나의 관련된 문구를 공통으로 갖는다. 예를 들면, 문구 "bolt action rifle" 및 "22"는 모두 관련 문구로서 "gun"을 갖을 수 있다. 이 경우, 서치 시스템(120)은 양쪽 문구 Q1 및 Q2의 포스팅 리스트를 검색하고 이 리스트를 교차시켜 양쪽 문구를 포함하는 문서의 리스트를 생성한다.

[0200] 그 후, 서치 시스템(120)은 생성된 문서 각각을 신속하게 스코어링할 수 있다. 우선, 서치 시스템(120)은 각 문서마다 스코어 조정값을 결정한다. 스코어 조정값은 문서의 관련 문구 비트 벡터 내의 질의 문구 Q1 및 Q2에 대응하는 위치에 있는 비트로 형성된 마스크이다. 예를 들면, Q1 및 Q2가 문서 d의 관련 문구 비트 벡터 내의 3번째 및 6번째의 2 비트(bi-bit) 위치에 대응하고, 3번째 위치의 비트값이 (1,1)이고 6번째 쌍의 비트값이 (1,0)이면, 스코어 조정값은 비트 마스크 "00 00 11 00 00 10"이다. 그 후, 스코어 조정값을 사용하여 문서의 관련 문구 비트 벡터를 마스킹한 후, 수정된 문구 비트 벡터가 문서의 본문 스코어(body score)를 산출하는데 사용될 랭킹 함수(후술함)로 패스된다.

[0201] 2. 랭킹(Ranking)

[0202] a) 포함된 문구에 기초한 문서의 랭킹(Ranking Documents Based on Contained Phrases)

[0203] 서치 시스템(120)은 각 문서의 관련 문구 비트 벡터, 및 질의 문구의 클러스터 비트 벡터의 문구 정보를 사용하여 서치 결과 내의 문서가 랭킹되는 랭킹 스테이지(604)를 제공한다. 이러한 접근법은 문서 내에 포함되는 문구, 또는 구어적으로 "본문 히트(body hits)"에 따라 문서를 랭킹한다.

[0204] 상기한 바와 같이, 임의로 주어진 문구 g_j에 대하여, g_j의 포스팅 리스트 내의 각 문서 d는 관련 문구 g_k 및 제 2 관련 문구 g_l이 문서 d 내에 존재하는지를 식별하는 연관된 관련 문구 비트 벡터를 갖는다. 더욱 관련된 문

구 및 제 2 관련 문구가 주어진 문서 내에 존재할 수록, 더욱 많은 비트가 주어진 문구에 대한 문서의 관련 문구 비트 벡터에 설정될 것이다. 더 많은 비트가 설정될 수록, 관련 문구 비트 벡터의 수치는 커진다.

[0205] 따라서, 일 실시예에서, 서치 시스템(120)은 관련 문구 비트 벡터의 값에 따라서 서치 결과 내의 문서들을 소팅한다. 질의 문구 Q에 가장 관련된 문구를 포함하는 문서가 가장 높은 값의 관련 문구 비트 벡터를 가질 것이고, 이들 문서는 서치 결과 내에서 가장 높은 랭킹 문서일 것이다.

[0206] 이러한 접근법은, 의미론적으로 이들 문서가 질의 문구에 가장 주제적으로 관련되기 때문에 바람직하다. 이러한 접근법은, 관련 문구 정보를 사용하여 관련 문서를 식별하고 이들 문서를 랭킹하기 때문에, 문서가 입력 질의 용어 q를 높은 빈도로 포함하지 않는 경우에도 높게 관련된 문서를 제공한다. 입력 질의 용어의 빈도가 낮은 문서가 질의 용어 및 문구에 대해 많은 수의 관련 문구를 아직 갖고 있을 수 있어, 관련 문구는 없지만 빈도가 높은 질의 용어 및 문구를 갖는 문서보다도 더욱 관련될 수 있다.

[0207] 제 2 실시예에서, 서치 시스템(120)은 문서가 포함하는 질의 문구 Q의 관련 문구에 따른 결과 세트 내의 각 문서를 스코어링한다. 이것은 다음과 같이 행해진다.

[0208] 각 질의 문구 Q가 주어지면, 문구 식별 프로세스 중에 식별되는, 질의 문구에 대해 관련된 문구 Qr의 얼마의 개수 N이 있을 것이다. 상기한 바와 같이, 관련 질의 문구 Qr은 질의 문구 Q로부터 그들의 정보 이득에 따라 배열된다. 그 후, 이들 관련 문구는 제 1 관련 문구 Qr1(즉, Q로부터 가장 높은 정보 이득을 갖는 관련 문구 Qr)의 N 포인트부터 시작하여, 다음의 관련 문구 Qr2의 N-1 포인트, 다음에 Qr3의 N-2 포인트 등의 포인트가 할당되어, 최종의 관련 문구 QrN은 1 포인트가 할당된다.

[0209] 그 후, 서치 결과 내의 각 문서는 질의 문구 Q의 관련 문구 Qr이 존재하는지를 판정하여 스코어링되고, 이러한 각 관련 문구 Qr에 할당된 포인트를 문서에 부여한다. 그 후, 문서들이 최상위 스코어로부터 최하위 스코어까지 소팅된다.

[0210] 더욱 개선하여, 서치 시스템(120)은 결과 세트로부터 소정의 문서를 발췌할 수 있다. 일부의 경우에는, 문서들이 다수의 상이한 주제에 관한 것일 수 있고, 이것은 특히 장문의 문서의 경우이다. 대다수의 경우에는, 유저들은 다수의 상이한 주제에 관련되는 문서들보다 질의 내에 표현되는 단일 주제에 대하여 강조되어 있는 문서들을 선호한다.

[0211] 이들 후자의 타입의 문서를 발췌하기 위해서는, 서치 시스템(120)은 질의 문구의 클러스터 비트 벡터 내의 클러스터 정보를 사용하여, 문서 내의 클러스터의 임계치보다도 많이 있는 임의의 문서를 제거한다. 예를 들면, 서치 시스템(120)은 2개의 클러스터보다도 많이 포함하는 임의의 문서를 제거할 수 있다. 이 클러스터 임계치는 미리 정해질 수 있거나, 유저가 서치 파라미터로서 설정할 수 있다.

[0212] b) 앵커 문구에 기초한 문서 랭킹(Ranking Documents Based on Anchor Phrases)

[0213] 일 실시예에서 질의 문구의 본문 히트에 기초하여 서치 결과 내의 문서를 랭킹하는 것 이외에, 서치 시스템(120)은 다른 문서들에 대한 앵커들 내에 질의 문구 Q 및 관련 질의 문구 Qr의 출현에 기초하여 문서들을 랭킹하기도 한다. 일 실시예에서, 서치 시스템(120)은 2개의 스코어, 즉 본문 히트 스코어와 앵커 히트 스코어의 함수(예를 들면, 1차 결합(linear combination))인 스코어를 각 문서마다 산출한다.

[0214] 예를 들면, 주어진 문서의 문서 스코어는 다음과 같이 산출될 수 있다.

[0215] 스코어 = .30*(본문 히트 스코어) + .70*(앵커 히트 스코어)

[0216] .30과 .70의 가중은 원하는 대로 조정될 수 있다. 문서의 본문 히트 스코어는 상기한 바와 마찬가지로 질의 문구 Qp가 주어진 경우에 문서에 대하여 가장 높게 평가된 관련 문구 비트 벡터의 수치이다. 선택적으로, 이 값은 서치 시스템(120)에서 인덱스(150) 내의 각 질의 문구 Q를 찾아내서 질의 문구 Q의 포스팅 리스트로부터 문서를 액세스한 다음에 관련 문구 비트 벡터를 액세스함으로써 직접 구해질 수 있다.

[0217] 문서 d의 앵커 히트 스코어는 질의 문구 Q의 관련 문구 비트 벡터의 함수이고, 여기서 Q는 문서 d를 참조하는 문서 내의 앵커 용어이다. 인덱싱 시스템(110)은 문서 컬렉션 내의 문서들을 인덱싱할 때, 문구가 아웃링크 내의 앵커 텍스트인 문서의 리스트를 각 문구마다 유지하고, 또한 다른 문서로부터 인링크의 리스트(및 연관된 앵커 텍스트)를 각 문서마다 유지한다. 문서의 인링크는 다른 문서(참조 문서들)로부터 주어진 문서로의 참조(하이퍼링크)이다.

[0218] 그 후, 주어진 문서 d에 대한 앵커 히트 스코어를 결정하기 위해서, 서치 시스템(120)은 그들의 앵커 문구 Q에

의해 인덱스 내에 리스팅된 참조할 문서 R(i=1에서 참조 문서의 수까지)의 세트에 걸쳐서 반복하여, 다음과 같은 곱을 합산한다.

[0219] R_i.Q.관련 문구 비트 벡터*D.Q.관련 문구 비트 벡터

[0220] 여기서 곱의 값은 얼마나 주제성이 있는 앵커 문구 Q가 문서 D에 있는지의 스코어이다. 여기서 이 스코어는 "인바운드 스코어 성분(inbound score component)"이라 부른다. 이 곱은 현재 문서 D의 관련 비트 벡터를 참조하는 문서 R 내의 앵커 문구의 관련 비트 벡터에 의해 효율적으로 부과한다. 참조 문서 R 그 자체가 질의 문구 Q에 관련되면(그래서, 보다 높게 값이 매겨진 관련 문구 비트 벡터를 가지면), 현재 문서 D 스코어의 중요성이 커진다. 그 후, 상기한 바와 같이, 본문 히트 스코어 및 앵커 히트 스코어를 조합하여 문서 스코어를 생성한다.

[0221] 다음에, 각각의 참조 문서 R마다, 각 앵커 문구 Q마다의 관련 문구 비트 벡터를 구한다. 이는 앵커 문구 Q가 문서 R에서 얼마나 주제성이 있는지의 측도이다. 여기서 이 값은 아웃바운드 스코어 성분(outbound score component)이라 부른다.

[0222] 그 후, 인덱스(150)로부터, 모든(참조하는 문서, 참조되는 문서) 쌍이 앵커 문구 Q에 대하여 추출된다. 그 후, 이들 쌍은 그들의 연관(아웃바운드 스코어 성분, 인바운드 스코어 성분) 값에 의해 소팅된다. 이러한 수행에 따라서, 이들 성분 중 어느 하나가 제 1 소팅 키일 수 있고, 다른 하나가 제 2 소팅 키일 수 있다. 그 후, 소팅 결과가 유저에게 표시된다. 아웃바운드 스코어 성분으로 문서를 소팅하는 것은, 앵커 히트로서 질의에 대하여 많은 관련 문구를 갖는 문서를 가장 높게 랭킹하게 하여, 이들 문서를 "익스퍼트(expert)" 문서로서 표시한다. 인바운드 문서 스코어로 소팅하는 것은, 앵커 용어에 의해 빈번하게 참조되는 문서를 가장 높게 랭킹하게 한다.

[0223] 3. 서치의 문구 기반 개인화(Phrase Based Personalization of Search)

[0224] 서치 시스템(120)의 다른 형태는 유저가 특별히 관심을 갖는 모델에 따라서 서치 결과의 랭킹을 개인화(606)하거나 맞춤화하는 능력이 있다. 이와 같이, 유저의 관심에 더욱 관련이 있을 것 같은 문서가 서치 결과 내에서 높게 랭킹된다. 서치 결과의 개인화는 다음과 같다.

[0225] 사전에, 문구로 표현될 수 있는 질의 및 문서와 관련하여 유저의 관심(예를 들면, 유저 모델)을 정의하는데 유용하다. 입력되는 서치 질의에 대하여, 질의는 질의 문구 Q, 관련 문구 Qr, 및 질의 문구 Qp의 문구 확장 Qe로 표현된다. 이와 같이, 이러한 용어 및 문구의 세트는 질의의 의미를 표현한다. 다음에, 문서의 의미는 페이지와 연관된 문구로 표현된다. 상기한 바와 같이, 질의 및 문서가 주어지면, 문서의 관련 문구는 문서에 인덱싱된 모든 문구에 대한 본문 스코어(관련 비트 벡터)로부터 결정된다. 마지막으로, 유저는 각각의 이들 요소를 표현하는 문구의 측면에서 질의들의 세트와 문서의 세트를 병합함으로써 표현될 수 있다. 유저를 표현하는 세트 내에 포함되는 특정 문서는 유저의 행동 및 목적지를 감시하는 클라이언트측 툴을 사용하여, 이전의 서치 결과 또는 자료의 일반적인 브라우징(인터넷 상에서의 문서 액세스)에서 유저가 선택하는 어떤 문서로부터 결정될 수 있다.

[0226] 개인화된 랭킹의 유저 모델을 구축 및 사용하는 프로세스는 다음과 같다.

[0227] 우선, 주어진 유저에 대하여, 최종의 K개의 질의 및 액세스된 P개의 문서의 리스트가 유지되고, 여기서 K 및 P는 각각 250 정도가 바람직하다. 이 리스트는 유저 계정 데이터베이스 내에 유지될 수 있고, 여기서 유저는 로그인에 의해 또는 브라우저 쿠키에 의해 인식될 수 있다. 주어진 유저에 대하여, 유저가 질의를 하는 최초에는 리스트가 비어있을 것이다.

[0228] 다음에, 질의 q가 유저로부터 수신된다. 상기한 바와 같이, q의 관련 문구 Qr이 문구 확장과 함께 검색된다. 이것이 질의 모델을 형성한다.

[0229] 1차 패스(예를 들면, 유저에 대해 저장된 질의 정보가 없는 경우)에서, 서치 시스템(120)은 더 맞춤화된 랭킹없이, 유저의 질의에 대한 서치 결과 내의 관련 문서를 단순히 불러오도록 동작한다.

[0230] 클라이언트측 브라우저 툴은 유저가 예를 들어 서치 결과 내의 문서 링크를 클릭함으로써 액세스하는 서치 결과 내의 문서들을 감시한다. 어떤 문구를 선택하는데 기초가 되는 이들 액세스된 문서는 유저 모델의 일부가 될 것이다. 각각의 이러한 액세스된 문서에 대하여, 서치 시스템(120)은 문서에 관련된 문구의 리스트인 문서용 문서 모델을 검색한다. 액세스된 문서에 관련되는 각각의 문구가 유저 모델에 추가된다.

[0231] 다음으로, 액세스된 문서와 관련된 문구가 주어지면, 이들 문구와 관련된 클러스터는 각각의 문구마다 클러스터 비트 벡터로부터 결정될 수 있다. 각각의 클러스터에서, 클러스터의 멤버인 각각의 문구는 상술한 바와 같이, 클러스터 비트 벡터 표시, 또는 클러스터 번호를 포함하는 관련 문구 테이블에서 문구를 찾아서 결정된다. 이 클러스터 번호는 유저 모델에 추가된다. 또한, 각각의 클러스터에서, 카운트는 유지되고 클러스터 내의 문구가 유저 모델에 추가될 때마다, 증가된다. 이 카운터는 후술하는 바와 같이 가중으로서 사용될 수 있다. 따라서, 유저 모델은 유저가 액세스하여 관심을 나타낸 문서에 있는 클러스터에 포함된 문구로부터 구축된다.

[0232] 동일한 일반적인 접근법은, 문서를 단순히 액세스하는 것보다는 유저가 높은 수준의 관심을 나타낸 문구 정보를 획득하기 위해 더욱 정밀하게 집중될 수 있다(유저는 그 문서가 정말로 관련 있는지를 간단하게 판단할 것이다). 예컨대, 유저 모델로의 문구 수집은 유저가 인쇄하거나, 저장하거나, 즐겨찾기 또는 링크(link)로서 등록하거나, 다른 유저에게 이메일을 보내거나, 브라우저 윈도를 장기간(예컨대, 10분)동안 열어놓은 문서에 한정될 수 있다. 이러한 액션 및 다른 액션은 문서의 높은 수준의 관심을 나타낸다.

[0233] 다른 질의가 유저로부터 수신된 경우, 관련 질의 문구 Qr가 검색된다. 관련 질의 문구 Qr은 유저 모델에 리스트된 문구와 교차되어 어떤 문구가 질의와 유저 모델 양쪽에 존재하는지를 결정한다. 마스크 비트 벡터는 질의 Qr의 관련 문구를 위해서 초기화된다. 이러한 비트 벡터는 상술한 바와 같이 2비트 벡터이다. 유저 모델에 또한 존재하는 질의의 각각의 관련 문구 Qr마다, 관련 문구용의 비트 양쪽이 마스크 비트 벡터에 설정된다. 따라서, 마스크 비트 벡터는 질의와 유저 모델 양쪽에 존재하는 관련 문구를 나타낸다.

[0234] 관련 문구 비트 벡터와 마스크 비트 벡터를 AND처리하여, 서치 결과의 현재 세트내의 각각의 문서마다 관련 문구 비트 벡터를 마스크하기 위하여 마스크 비트 벡터가 사용된다. 이는 본문 스코어(body score) 및 앵커 히트 스코어(anchor hit score)를 마스크 비트 벡터에 의해 조정하는 효과를 갖는다. 문서는 이전과 같이 바디 스코어 및 앵커 스코어에 대해 스코어링(scoring)되고 유저에게 제시된다. 이러한 접근법은, 높게 랭크되기 위해 유저 모델에 포함되는 질의 문구를 문서가 포함하는 것을 필수적으로 요구한다.

[0235] 상술하는 엄격한 제약을 부과하지 않는 다른 실시예로서, 마스크 비트 벡터가 어래이(array)로 계산될 수 있어서, 각각의 비트는 유저 모델의 관련 문구용 클러스터 카운트를 가중하는데 사용된다. 따라서, 각각의 클러스터 카운터는 0 또는 1로 승산되어, 효과적으로 카운트(count)를 제로로 만들거나 유지시킨다. 다음으로, 스코어링되는 각각의 문서마다 관련 문구를 곱하기 위해 가중이 또한 사용되는 경우에, 이러한 카운트 자체가 사용된다. 이러한 접근법은 관련 문구로서 질의 문구를 갖지 않는 문서가 여전히 적절하게 스코어링될 수 있게 하는 장점을 갖는다.

[0236] 마지막으로, 유저 모델은 현재 세션(session)에 한정될 수 있고, 일 세션은 서치에서 액티브한 기간의 간격이며, 그 세션 후에 유저 모델이 덤프(dump)된다. 또한, 주어진 유저용 유저 모델은 계속 유지되고 나서, 가중을 줄이거나 에이징(aging)될 수 있다.

[0237] Ⅳ. 결과 프리젠테이션

[0238] 프리젠데이션 시스템(130)은 스코어링되고 소팅된 서치 결과를 서치 시스템(120)으로부터 수신하고, 유저에게 그 결과를 제시하기 이전에 조직화, 주석 달기, 및 클러스터링 작업을 추가로 수행한다. 이러한 작업은 서치 결과의 내용에 대한 유저의 이해를 돕고, 중복을 없애고, 서치 결과의 더욱 대표적인 샘플링을 제공한다. 도 7은 프리젠테이션 시스템(130)의 주요 기능적 작업을 나타낸다.

[0239] 700: 주제 클러스터에 따라 문서를 클러스터링

[0240] 702: 문서 설명(description)을 생성

[0241] 704: 중복 문서를 제거

[0242] 각각의 이들 작업은 서치 결과(701)를 입력으로 받아서 수정된 서치 결과(703)를 출력한다. 도 7에 도시된 바와 같이, 이들 작업의 순서는 독립적이며, 해당 실시예에서 원하는 경우 변화될 수 있어서, 입력은 도시된 바와 같이 병렬로 이루어지는 것 대신에 파이프라인(pipeline)화 될 수 있다.

[0243] 1. 프리젠테이션을 위한 다이내믹 분류(dynamic taxonomy) 생성

[0244] 해당 질의에 대하여, 그 질의를 만족시키는 수백, 심지어 수천의 문서를 리턴시키는 것이 통상적이다. 여러 경우에, 서로 다른 내용을 갖는 특정 문서가 충분히 관련되어서 관련 문서의 의미 있는 그룹, 본질적으로 클러스터를 형성한다. 하지만, 대부분의 유저는 서치 결과에서 처음 30 내지 40 개의 문서 이상은 검토하지 않는다.

그리하여, 예컨대, 세 클러스터로부터 처음 100 개의 문서가 출현하지만, 다음의 100 개의 문서가 추가의 네 클러스터에 해당하면, 추가의 조정이 없는 경우에, 유저는 통상적으로 그와 같은 나중의 문서는 검토하지 않는다. 사실상, 그와 같은 나중의 문서는 질의와 관련된 각종 상이한 주제를 나타내기 때문에, 유저의 질의와 매우 관련적일 수 있다. 따라서, 각 클러스터로부터의 문서의 샘플을 유저에게 제공함으로써, 서치 결과로부터의 상이한 문서의 폭넓은 선택을 유저에게 부여하는 것이 소망된다. 프리젠테이션 시스템(130)은 이것을 다음과 같이 행한다.

[0245] 시스템(100)의 다른 태양으로서, 프리젠테이션 시스템(130)은 서치 결과의 각각의 문서 d마다 관련 문구 비트 벡터를 이용한다. 더욱 상세하게는, 각각의 질의 문구 Q 및 Q의 포스팅 리스트(posting list) 내의 각각의 문서 d마다, 관련 문구 비트 벡터는 어느 관련 문구 Qr가 문서에 존재하는지를 나타낸다. 서치 결과의 문서 세트에 대하여, 각각의 관련 문구 Qr마다, 얼마나 많은 문서가 관련 문구 Qr을 포함하는지를 나타내기 위해 Qr에 대응하는 비트 위치의 비트 값을 합계하여 카운트가 결정된다. 서치 결과에 대하여 합계되고 소팅될 때, 가장 자주 출현하는 관련 문구 Qr가 지시되고, 그 문구 각각이 문서의 클러스터가 될 것이다. 가장 자주 출현하는 관련 문구는 명칭으로서 그것의 관련 문구 Qr을 취하는 제 1 클러스터이고, 기타 상위 3개 내지 상위 5개 클러스터이다. 따라서, 각각의 상위 클러스터는 클러스터의 표제 또는 명칭으로서 문구 Qr과 함께 식별된다.

[0246] 각각의 클러스터로부터의 문서는 여러 방법으로 유저에게 제시된다. 일 적용에서, 각각의 클러스터로부터의 고정된 수의 문서, 예컨대, 각각의 클러스터에서의 상위 스코어 10개 문서가 제시될 수 있다. 다른 적용에서, 각각의 클러스터로부터의 비례수의 문서가 나타날 수 있다. 따라서, 클러스터 1에 50개, 클러스터 2에 30개, 클러스터 3에 10개, 클러스터 4에 7개, 및 클러스터 5에 3개로 하여 서치 결과에 100개 문서가 있고, 20개의 문서만을 나타내고자 하는 경우에, 문서는 다음과 같이 선택된다: 클러스터 1로부터 10개의 문서, 클러스터 2로부터 7개의 문서, 클러스터 3으로부터 2개의 문서, 및 클러스터 4로부터 1개의 문서. 그후, 문서는 유저에게 나타내어지고, 표제로서의 적절한 클러스터 명칭에 따라서 그룹화될 수 있다.

[0247] 예컨대, 검색 질의를 "blue merle agility training"라 하면, 서치 시스템(120)은 100개의 문서를 검색한다. 서치 시스템(120)은 "blue merle" 및 "agility training"을 질의 문구로서 미리 식별할 것이다. 이들 질의 문구의 관련 문구는 다음과 같다:

[0248] "blue merle"::"Australian Shepherd", "red merle", "tricolor", "aussie";

[0249] "agility training"::"weave poles", "teeter", "tunnel", "obstacle", "border collie".

[0250] 그 후, 프리젠테이션 시스템(130)은 각각의 질의 문구의 상기 관련 문구의 각각에 대하여, 그와 같은 문구를 포함하는 문서 수의 카운트를 결정한다. 예컨대, 문구 "weave poles"가 100개의 문서 중에서 75개의 문서에 나타나고, "teeter"가 60개의 문서에 나타나고, "red merle"가 50개의 문서에 나타난다고 가정한다. 그때, 제 1 클러스터는 "weave poles"라 칭하고, 그 클러스터로부터 선택된 문서의 수가 나타내지며, 제 2 클러스터는 "teeter"라 칭하고, 선택된 수가 역시 나타내어진다. 고정적 프리젠테이션의 경우, 각각의 클러스터로부터 10개의 문서가 선택될 수 있다. 비례적 프리젠테이션은 문서의 전체 수에 대한 각각의 클러스터로부터의 문서의 비례수를 사용할 것이다.

[0251] 2. 주제 관련 문서 설명

[0252] 프리젠테이션 시스템(130)의 제 2 기능은 각각의 문서마다 서치 결과 프리젠테이션에 삽입될 수 있는 문서 설명의 생성이다(702). 이 설명은 각각의 문서에 나타난 관련 문구에 기초하며, 그리하여 문서가 무엇에 관한 것인지를 유저가 서치와 문맥적으로 관련되어 있는 방식으로 이해할 수 있도록 도운다. 문서 설명은 일반적이거나 유저에게 개인화될 수 있다.

[0253] a) 일반적 주제 문서 설명

[0254] 상술한 바와 같이, 질의가 주어지면, 서치 시스템(120)은 관련 질의 문구 Qr 및 질의 문구의 문구 확장을 또한 결정하고, 질의에 대한 관련 문서를 식별한다. 프리젠테이션 시스템(130)은 서치 결과 내의 각각의 문서를 액세스하고 다음의 작업을 수행한다.

[0255] 먼저, 프리젠테이션 시스템(130)은 문서의 문장을 질의 문구 Q, 관련 질의 문구 Qr, 및 문구 확장 Qp의 인스턴스의 수에 의해서 랭킹하여, 문서의 각각의 문장에 대하여 세 태양의 카운트를 유지한다.

[0256] 그 후, 문장은 그러한 카운트에 의해서 소팅되고, 제 1 소팅 키는 질의 문구 Q의 카운트이며, 제 2 소팅 키는

관련 질의 문구 Qr의 카운트이고, 마지막 소팅 키는 문구 확장 Qp의 카운트이다.

[0257] 최종적으로, 소팅 후의 상위 N(예컨대, 5) 문장이 문서의 설명으로서 사용된다. 이러한 문장의 세트는 포맷되어 수정된 서치 결과(703) 내의 문서의 프리젠테이션에 포함될 수 있다. 이러한 프로세스는 서치 결과 내의 몇 개의 문서에 반복되고, 유저가 결과의 다음 페이지를 요구할 때마다 요구에 따라 행해질 수 있다.

[0258] b) 개인화된 주제 기반 문서 설명

[0259] 서치 결과의 개인화가 제공된 실시예에서, 문서 설명 또한 유저 모델에서 표현된 바와 같이 유저 관심을 반영하기 위해서 개인화될 수 있다. 프리젠테이션 시스템(130)이 이것을 다음과 같이 수행한다.

[0260] 먼저, 상술한 바와 같이, 프리젠테이션 시스템(130)은 질의 관련 문구 Qr을 유저 모델과 교차시킴으로써 유저에 해당되는 관련 문구를 결정한다(유저에 의해서 액세스된 문서 내의 문구를 리스팅함).

[0261] 그 후, 프리젠테이션 시스템(130)은 비트 벡터 자체의 값에 따라서 유저 관련 문구 Ur의 세트를 안정적으로 소팅하고, 소팅된 리스트를 질의 관련 문구 Qr의 리스트에 추가하고, 임의의 중복 문구를 제거한다. 안정적인 소팅은 동등하게 랭크된 문구의 기존의 순서를 유지한다. 이는 질의 또는 유저에 관련된 관련 문구의 세트로 귀착하고 세트 Qu라 칭한다.

[0262] 프리젠테이션 시스템(130)은, 상술한 바와 같은 일반적인 문서 설명 프로세스와 유사한 방식으로, 서치 결과 내의 각각의 문서 내의 문장을 랭킹하기 위한 근거(base)로서 문구의 순위 리스트를 사용한다. 따라서, 해당 문서에 대하여, 프리젠테이션 시스템(130)은 유저 관련 문구 및 질의 관련 문구(Qu) 각각의 인스턴스의 수에 의해서 문서의 문장을 랭킹하고, 질의 카운트에 따라서 랭킹된 문장을 소팅하며, 최종적으로 그와 같은 각각의 문구마다 문구 확장의 수에 기초하여 소팅한다. 앞에서는, 소팅 키의 순서가 질의 문구 Q, 관련 질의 문구 Qr, 및 문구 확장 Qp 순이였지만, 여기서는 소팅 키는 가장 높이 랭킹된 유저 관련 문구 Ur로부터 가장 낮게 랭크된 유저 관련 문구 Ur의 순이다.

[0263] 또한, 이러한 프로세스는 서치 결과 내의 문서에 대하여 반복된다(요구시 또는 미리). 그와 같은 각각의 문서에서, 결과적인 문서 설명은 문서로부터의 N개의 상위 랭크 문장을 포함한다. 여기서, 이들 문장은 유저 관련 문구 Ur의 최대 수를 갖는 것들이며, 그리하여 (적어도 유저 모델에 포착된 정보에 따른) 유저와 가장 관련 있는 개념 및 주제를 나타내는 문서의 중요 문장을 나타낸다.

[0264] 3. 중복 문서 검출 및 제거

[0265] 인터넷과 같은 대규모 자료에서, 동일 문서의 복합적인 예가 존재하고, 문서의 일부가 여러 상이한 위치에 존재하는 것은 매우 통상적이다. 예컨대, "Associated Press"와 같은 뉴스 지국에서 작성한 해당 뉴스 기사는 개별적인 신문의 여러 웹사이트에서 반복된다. 서치 질의에 대응하여 이들 중복 문서를 모두 포함시키면, 중복 문서로 인하여 유저에 부담만 주며, 질의에 대응하는데 유용하지 않다. 따라서, 프리젠테이션 시스템(130)은 중복되거나 서로 거의 중복될 것 같은 문서를 식별하기 위한 추가의 능력(704)을 제공하여, 서치 결과에는 이것들 중에서 하나만을 포함시킨다. 따라서, 유저는 더욱 다양화되고 확실한 결과의 세트를 수신하고, 서로 중복되는 문서를 리뷰하는데 시간을 소모하지 않아도 된다. 프리젠테이션 시스템(130)은 다음과 같은 기능을 제공한다.

[0266] 프리젠테이션 시스템(130)은 서치 결과 세트(701) 내의 각각의 문서를 처리한다. 각각의 문서 d에서, 프리젠테이션 시스템(130)은 문서와 관련된 관련 문구 R의 리스트를 먼저 결정한다. 이들 관련 문구의 각각에 대하여, 프리젠테이션 시스템(130)은 각각의 문구의 출현 빈도수에 따라 문서의 문장을 랭킹하고, 상위 N(예컨대, 5 내지 10)개의 문장을 선택한다. 이러한 문장 세트는 문서와 관련하여 저장된다. 이를 행하는 하나의 방식은 선택된 문장을 연결한 후, 문서 식별자(identifier)를 저장하기 위하여 해시 테이블(hash table)을 사용하는 것이다.

[0267] 그 후, 프리젠테이션 시스템(130)은 각각의 문서 d의 선택된 문장을 서치 결과(701) 내의 다른 문서의 선택된 문장과 비교하여, 선택된 문장이 (허용 오차 내에서)일치하면, 문서는 중복된 것으로 추정되고, 그것들 중 하나가 서치 결과로부터 제거된다. 예컨대, 프리젠테이션 시스템(130)은 연결된 문장을 해시할 수 있고, 해시 테이블이 해시 값의 엔트리(entry)를 미리 갖고 있다면, 이는 현재의 문서와 현재 해시된 문서가 중복된다는 것을 지시한다. 프리젠테이션 시스템(130)은 문서 중의 하나의 문서 ID로 테이블을 갱신할 수 있다. 바람직하게는, 프리젠테이션 시스템(130)은 높은 페이지 랭크 또는 문서 중요도의 다른 질의 독립 측도를 유지한다. 또한, 프리젠테이션 시스템(130)은 인덱스(150)를 수정하여, 임의의 질의에 대한 향후 서치 결과에 나타나지 않도록 중복 문서를 제거한다.

[0268] 동일한 중복 제거 프로세스는 인덱싱 시스템(110)에 의해서 직접 적용될 수 있다. 문서가 크롤링되는 경우, 선택된 문장 및 문장의 해쉬를 얻기 위하여 상술한 문서 설명 프로세스가 수행된다. 해쉬 테이블이 채워지면, 새롭게 크롤링된 문서는 이전 문서의 중복물로 여겨진다. 그러면, 다시, 인덱싱 시스템(110)은 높은 페이지 랭크 또는 다른 질의 독립 측도를 갖는 문서를 유지할 수 있다.

[0269] 본 발명은 하나의 가능한 실시예에 대하여 특히 상세하게 설명하였다. 당업자는 본 발명이 다른 실시예로 실시될 수 있다는 것을 이해할 것이다. 먼저, 구성 요소의 특정 호칭, 용어의 대문자화, 어트리뷰트(attribute), 데이터 구조, 또는 다른 프로그래밍 또는 구조적 태양은 의무적이거나 중요한 것이 아니며, 본 발명을 실시하는 메커니즘 또는 그것의 특징은 다른 명칭, 포맷, 또는 프로토콜을 가질 수 있다. 또한, 시스템은 기술한 바와 같이, 하드웨어와 소프트웨어의 조합을 통하여, 또는 하드웨어 요소만으로 실시될 수 있다. 또한, 본 명세서에서 기재된 각종 시스템 요소들 사이의 기능의 특정 분할은 단순히 예시적이며, 의무적이지 않고, 단일 시스템 요소에 의해서 수행되는 기능은 복합 요소에 의해서 수행될 수 있으며, 복합 요소에 의해서 수행되는 기능은 단일 요소에 의해서 대신 수행될 수 있다.

[0270] 상술한 설명의 일부는 본 발명의 특징을 정보에 대한 동작의 알고리즘 및 기호 표시 관점으로 나타낸다. 이러한 알고리즘 설명 및 표시는 데이터 처리 분야의 당업자가 그들의 업무의 실체를 다른 당업자에게 가장 효율적으로 전달하기 위해서 사용되는 수단이다. 기능적 또는 논리적으로 기재되는 이러한 동작은 컴퓨터 프로그램에 의해서 실시되는 것으로 이해된다. 또한, 때때로 동작의 이러한 구성을 일반성의 상실 없이 모듈로서 또는 기능적인 명칭으로서 칭하는 것이 편리하다는 것이 입증되었다.

[0271] 상술한 설명으로부터 명백한 바와 같이, 다르게 특정되지 않는 한, 상세한 설명에 걸쳐서, "처리", "산출", "계산", "판정", "표시" 등의 용어 사용은, 컴퓨터 시스템 메모리나 레지스터 또는 다른 정보 스토리지, 전송 또는 디스플레이 장치 내의 물리적(전자) 양으로서 나타내지는 데이터를 조작하고 전송하는 컴퓨터 시스템, 또는 유사한 전자 계산 장치의 작용이나 프로세스를 칭하는 것이다.

[0272] 본 발명의 특정 태양은 알고리즘의 형태로 여기에 기재된 프로세스 스텝 또는 명령을 포함한다. 본 발명의 프로세스 스텝 및 명령은 소프트웨어, 펌웨어, 또는 하드웨어로 실시될 수 있고, 소프트웨어로 실시되는 경우, 실시간 네트워크 오퍼레이팅 시스템에 의해서 사용되는 상이한 플랫폼에 체재하기 위하여 다운 로드 되어 오퍼레이팅 된다.

[0273] 또한, 본 발명은 오퍼레이션을 수행하는 장치에 관한 것이다. 이러한 장치는 요구된 목적을 위해서 특별하게 구성될 수 있거나, 컴퓨터에 의해서 액세스될 수 있는 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램에 의해서 재구성되거나 선택적으로 활성화되는 범용 컴퓨터를 포함할 수 있다. 그러한 컴퓨터 프로그램은 플로피 디스크, 광 디스크, CD-ROM, 자기-광 디스크, ROM(read only memory), RAM(random access memory), EPROM, EEPROM, 자기 또는 광 카드, 특정 용도 집적 회로(ASIC)를 포함하는 임의 종류의 디스크, 또는 전자 명령을 저장하는데 적합하고 컴퓨터 시스템 버스에 각각 접속되는 임의 종류의 매체와 같은 컴퓨터 판독 가능 저장 매체를 포함하지만 그것에 한정되는 것은 아니다. 또한, 명세서 내에서 칭해지는 컴퓨터는 단일 프로세서를 포함하거나 증가된 계산 능력을 위한 복합 프로세서 디자인을 채용하는 아키텍처(architecture)일 수 있다.

[0274] 본 명세서에 기재된 알고리즘 및 오퍼레이션은 임의의 특정 컴퓨터 또는 다른 장치에 본질적으로 관련되지는 않는다. 또한, 다양한 범용 시스템이 교시에 따른 프로그램과 함께 사용될 수 있거나, 요구되는 방법 스텝을 수행하기 위한 더욱 특정된 장치를 구성하는 것이 편리하다는 것이 입증될 수 있다. 각종 시스템의 요구되는 구조는 동등한 변형예와 함께 당업자에게는 자명할 것이다. 또한, 본 발명은 임의의 특정 프로그래밍 언어를 참조하여 기재되지 않는다. 각종 프로그래밍 언어가 명세서에 기재된 본 발명의 교시를 실시하는데 사용될 수 있고, 특정 언어에 대한 임의의 참조가 본 발명의 기능 및 최선의 모드를 개시하기 위해 제공된다.

[0275] 본 발명은 수많은 토폴로지(topology)에 걸쳐있는 다양한 컴퓨터 네트워크 시스템에 매우 적합하다. 이러한 분야에서, 큰 네트워크의 구성 및 관리는 다른 컴퓨터와 통신 가능하게 연결되는 컴퓨터 및 스토리지 장치, 및 인터넷과 같은 네트워크 상의 스토리지 장치를 포함한다.

[0276] 최종적으로, 본 명세서에서 사용된 언어는 본질적으로 읽기 용이성 및 교시적 목적을 위해서 선택되었으며, 발명의 주안점을 정확히 서술하거나 제한하기 위해서 선택된 것은 아니다. 따라서, 본 발명의 개시는 첨부된 특허청구범위에 기재된 본 발명의 범주를 한정하지 않고 예시하기 위하여 의도된 것이다.

발명의 효과

[0277] 본원 발명에서는 대규모 자료에서 문구를 포괄적으로 식별하고, 문구에 따른 문서를 인덱싱하고, 이들 문구에 따라 문서를 서치 및 랭킹하고, 문서에 관한 부가적인 클러스터링 및 설명 정보를 제공할 수 있는 정보 검색 시스템 및 방법이 제공된다.

도면의 간단한 설명

[0001] 도 1은 본 발명의 일 실시예의 소프트웨어 아키텍처의 블록도.

[0002] 도 2는 문서 내의 문구를 식별하는 방법을 나타내는 도면.

[0003] 도 3은 문구 윈도 및 제 2 윈도를 갖는 문서를 나타내는 도면.

[0004] 도 4는 관련 문구를 식별하는 방법을 나타내는 도면.

[0005] 도 5는 관련 문구에 대해 문서를 인덱싱하는 방법을 나타내는 도면.

[0006] 도 6은 문구에 기초하여 문서를 검색하는 방법을 나타내는 도면.

[0007] 도 7은 서치 결과를 제공하는 프리젠테이션 시스템의 동작을 나타내는 도면.

[0008] 도 8의 (a) 및 (b)는 참조하는 문서와 참조되는 문서간의 관계를 나타내는 도면.

[0009] 도면 부호의 설명

[0010] 100 서치 시스템 110 인덱싱 시스템

[0011] 130 프리젠테이션 시스템 140 프론트 엔드 서버

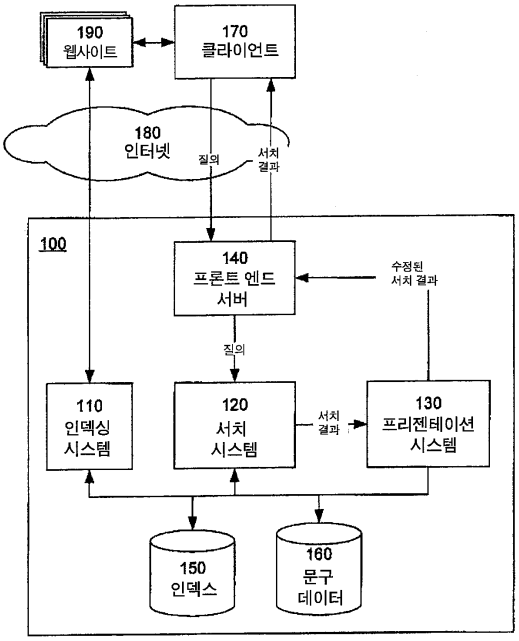
[0012] 150 인덱스 160 문구 데이터

[0013] 170 클라이언트 180 인터넷

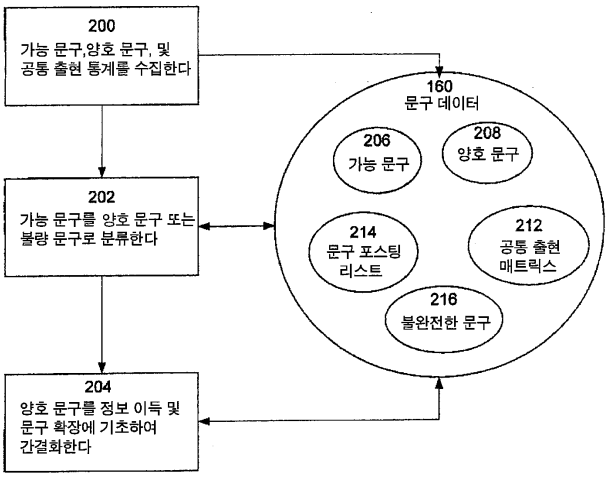
[0014] 190 웹사이트

도면

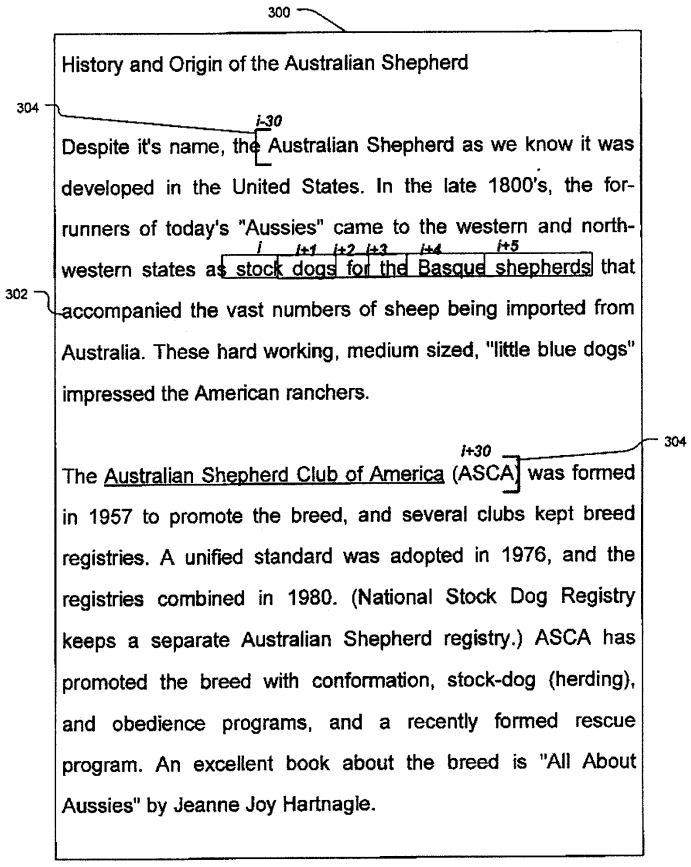
도면1



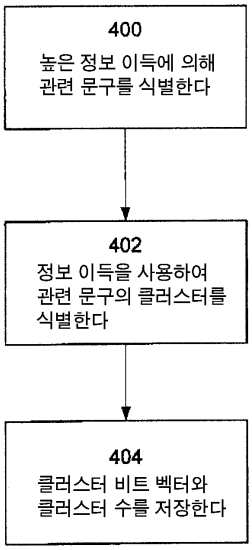
도면2


200
가능 문구,양호 문구, 및
공통 출현 통계를 수집한다
160
문구 데이터
206
가능 문구
208
양호 문구
202
가능 문구를 양호 문구 또는
불량 문구로 분류한다
214
문구 포스팅
리스트
212
공통 출현
매트릭스
216
불완전한 문구
204
양호 문구를 정보 이득 및
문구 확장에 기초하여
간결화한다

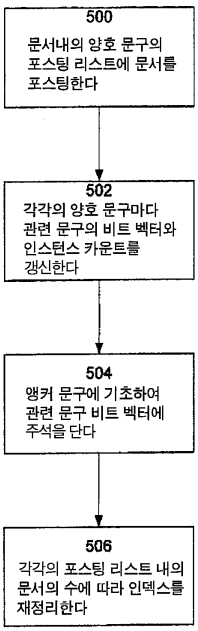
도면3


300
History and Origin of the Australian Shepherd
304
i-30
Despite it's name, the Australian Shepherd as we know it was developed in the United States. In the late 1800's, the forrunners of today's "Aussies" came to the western and northwestern states as stock dogs for the Basque shepherds that
i
i+1
i+2
i+3
i+4
i+5
302
accompanied the vast numbers of sheep being imported from Australia. These hard working, medium sized, "little blue dogs" impressed the American ranchers.
i+30
304
The Australian Shepherd Club of America (ASCA) was formed in 1957 to promote the breed, and several clubs kept breed registries. A unified standard was adopted in 1976, and the registries combined in 1980. (National Stock Dog Registry keeps a separate Australian Shepherd registry.) ASCA has promoted the breed with conformation, stock-dog (herding), and obedience programs, and a recently formed rescue program. An excellent book about the breed is "All About Aussies" by Jeanne Joy Hartnagle.

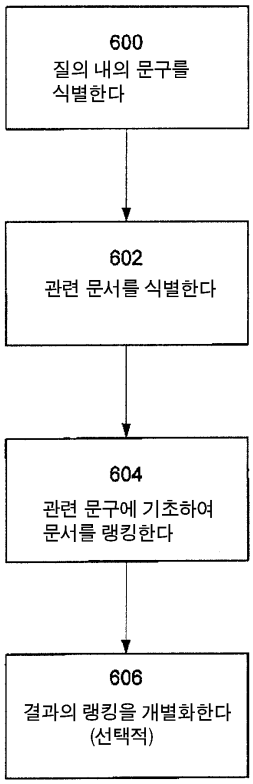
도면4


400
높은 정보 이득에 의해 관련 문구를 식별한다
402
정보 이득을 사용하여 관련 문구의 클러스터를 식별한다
404
클러스터 비트 벡터와 클러스터 수를 저장한다

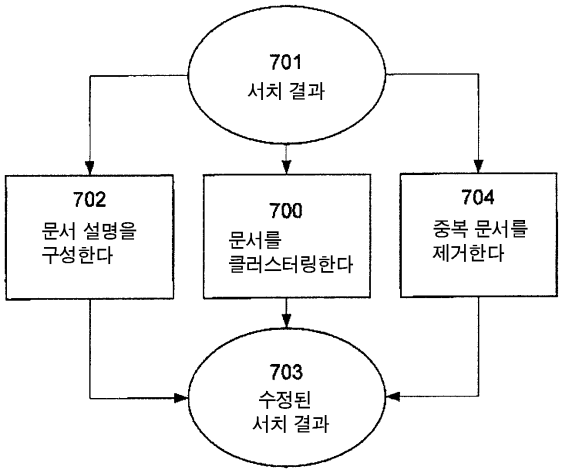
도면5


500
문서내의 앙호 문구의 포스팅 리스트에 문서를 포스팅한다
502
각각의 양호 문구마다 관련 문구의 비트 벡터와 인스턴스 카운트를 갱신한다
504
앵커 문구에 기초하여 관련 문구 비트 벡터에 주석을 단다
506
각각의 포스팅 리스트 내의 문서의 수에 따라 인덱스를 재정리한다

도면6


600
질의 내의 문구를
식별한다
602
관련 문서를 식별한다
604
관련 문구에 기초하여
문서를 랭킹한다
606
결과의 랭킹을 개별화한다
(선택적)

도면7


701
서치 결과
702
문서 설명을
구성한다
700
문서를
클러스터링한다
704
중복 문서를
제거한다
703
수정된
서치 결과

도면8