(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl. [7]
G06F 17/30

(11) 공개번호 특2002-0003701
(43) 공개일자 2002년01월15일

(21) 출원번호 10-2000-0035714
(22) 출원일자 2000년06월27일

(71) 출원인 토팔러지 리서치 인코퍼레이티드
첸 체프
대만 타이페이 세크 4 츙 시아오 이 로드 넘버 563 6플로어
쳉 유엔시엔
대만 타이페이 시엔 팬-시아오 쥬-쿠앙로드 레인 133 넘버 2 7플로어

(72) 발명자 쳉유엔시엔
대만타이페이시엔팬-시아오쥬-쿠앙로드레인133넘버27플로어

(74) 대리인 임창현

심사청구 : 있음

(54) 디지털 문서의 키워드를 자동으로 추출하는 방법

요약

여기에 개시된 본 발명은 디지털 문서의 키워드를 자동으로 추출하는 방법을 제공한다. 우선, 디지털 문서는 적어도 하나의 리스트 엘리먼트를 포함하는 리스트 데이터 구조로 변환되고, 결합 영역은 비어있는 상태로 설정된다. 상기 결합 영역이 비어있는 상태로 설정된 후, 리스트 엘리먼트들은 순차적으로 인출된다. 그리고 나서, 인출된 리스트 엘리먼트들은 스킵(skip)되거나 또는 결합되어 결합 영역 또는 최종 영역에 삽입된다. 다음, 결합 영역은 리스트 데이터 구조로 변환된 후, 종료 조건이 만족될 때까지 상술한 바와 같은 단계들이 반복 수행된다. 마지막으로, 키워드 검색에서 주어진 디스플레이 조건에 따라서 상기 최종 영역 내의 리스트 엘리먼트들은 디스플레이된다.

대표도

도 3

명세서

도면의 간단한 설명

도 1은 종래의 키워드 추출 방법의 의사 코드를 보여주는 도면;

도 2는 본 발명의 바람직한 실시예에 따른 의사 코드의 일부를 보여주는 도면;

도 3은 본 발명의 바람직한 실시예에 따른 해시 기능과 스트링의 발생 빈도를 이용하는 프로세스를 보여주는 도면;

도 4는 본 발명의 바람직한 실시예에 따른 프로세스를 보여주는 도면;

도 5는 본 발명의 바람직한 실시예에 따른 의사 코드를 보여주는 도면;

도 6A 내지 도 6B는 중국어와 영어를 포함하는 환경에 본 발명을 적용하는 것을 보여주는 도면;

도 7A 내지 도 7C는 중국어와 영어를 위해 설계된 본 발명의 바람직한 실시예가 한국어를 추출기 위해 사용된 경우의 결과를 보여주는 도면;

도 8은 OCR 문서로부터 키워드를 추출할 때 본 발명을 적용한 결과를 보여주는 도면;

도 9는 또다른 OCR 문서로부터 키워드를 추출할 때 본 발명을 적용한 결과를 보여주는 도면;

도 10A 내지 도 10B는 다이내믹 사전을 구성하는데 있어서 본 발명의 결과를 이용하는 방법을 보여주는 도면; 그리고

도 11은 내용에 기반을 둔 음악 수정을 위해 본 발명을 적용한 것을 보여주는 도면이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 디지털 문서를 자동으로 색인(indexing)하는 방법에 관한 것으로, 좀 더 구체적으로는 디지털 문서의 키워드를 자동으로 추출하는 방법에 관한 것이다.

인터넷(Internet)이 급속도로 널리 보급되고 성장함에 따라, 검색 시스템(retrieval system)의 데이터와 취급이 이전에 비해 급속히 성장하고 있다. 자동화 기술은 더욱 쉽고 더욱 효과적으로 서비스를 제공하기 위해 검색 시스템 특히, 문서의 내용에 기반을 둔 검색에 적용되었다. 그러나, 자동 인덱싱(automatic indexing), 유사 비교(approximate comparison), 자동 분류(automatic classification), 적절한 피드백(relevant feedback) 등과 같은 대부분의 자동 검색 기능들에 있어서, 문서에서 자동으로 키워드를 추출하는 방법은 그러한 자동화 기술들이 수행되기 전에 수행되어야 한다. 이러한 이유로, 문서가 참고 문헌 데이터, 디지털 텍스트, 잘못된 문서(OCR(Optical Character Recognition) 에러로 인한 부정확한 단어를 포함하고 있는 경우), 또는 MIDI(Musical Instrument Digital Interface) 파일과 같은 디지털 음악인 지의 여부와 무관하게 키워드를 추출하는 방법은 이 기술 분야의 핵심이다.

최종 목표 매체가 무엇이더라도 대부분의 검색 시스템은 검색을 위한 참고 문헌 데이터만을 제공할 수 있다. 비록 음악 데이터가 작곡자의 이름 또는 음악의 제목에 따라 획득될 수 있다하더라도, 음악 데이터의 접근 방법은 실제 적용에 있어서 여전히 한계가 있다. 예를 들어, 누군가 음악의 제목 또는 작곡자의 이름 대신에 멜로디의 일부만으로 음악 데이터를 찾고자할 때, 참고 문헌 데이터를 이용해서는 접근될 수 없다. 또다른 예로, 작곡자가 어떤 영감이 떠올랐을 때 비슷한 멜로디가 이미 작곡되었는 지의 여부를 체크하고자 할 때에도 참고 문헌 데이터를 이용해서는 역시 검색할 수 없다. 디지털 음악 데이터에 대한 요구는 더욱 더 증대되는 요즈음, 내용에 기반을 둔 색인 및 검색 방법은 참고 문헌 데이터 색인 방법에 비해 더욱 더 중요해지고 있다.

고도로 실용적인 내용에 기반을 둔 음악 검색 시스템을 확립하기 이전에 몇가지 문제점을 해결해야만 한다. 첫 번째 문제는 완전히 정확하지 않은 불충분한 멜로디가 질의되어 검색시 정확한 멜로디를 찾을 수 없는 경우이다. 이러한 오류를 " 어휘 불일치(vocabulary mismatch)" 라 한다. 그러므로, 뮤직 검색 시스템과 같은 분야에서는 유사 비교가 중요하다. 그러나, 유사 비교는 계산 시간 또는 저장 매체와 같은 시스템 자원을 많이 소모한다. 더욱이, 음악의 내용은 청취하는데 적합하지만 검색하는데에는 적합하지 않다. 따라서, 검색의 결과를 검사하는데에 많은 시간이 소요된다.

몇몇 키워드 검색 방법들은 모든 종류의 디지털 문서에 적당하도록 개발되어 있으므로, 텍스트 데이터에서 키워드를 추출하는 방법들과 디지털 음악 데이터에서 키 멜로디(key melody)를 추출하는 방법들은 다음에서 각각 분리하여 설명된다. 키워드를 추출하는 방법은 주로 사전 비교 방법, 문법 분석 방법, 그리고 통계적인 분석 방법을 포함한다. 사전 비교 방법은 입력 문서(또는 문장)와 어휘 데이터베이스(phrase database) 모두에 나타나는 어휘를 추출하기 위해 입력 문서를 이미 존재하는 어휘 데이터베이스와 비교한다. 이러한 방법은 어휘 데이터베이스 내의 어휘가 입력 문서에 있는지의 여부만을 체크하면 되므로 쉽다. 이러한 방법에 의해 추출된 어휘는 모든 문서의 키워드라 보장되지 않는다. 이 방법의 또다른 단점은 모든 특수 분야의 어휘를 데이터베이스에 확보하기 위해 데이터베이스를 유지 보수하는데 많은 시간과 노력이 소요된다는 것과, 데이터베이스에 포함되지 않은 어휘에 있어서는 무용지물이 되고, 그리고 데이터베이스의 크기가 크면 비교 속도가 비능률적이라는 것이다.

문법 분석 방법은 자연어 프로세싱 기술을 이용하여 문서에 포함된 명사 어휘들을 우선 추출한다. 몇몇 다른 방법들과 원칙들은 적절치 않은 어휘를 추출하기도 한다. 이러한 방법의 결과는 대부분 의미심장한 명사 어휘이다. 대부분의 문법 분석기는 이미 존재하는 어휘 데이터베이스에 의존하기 때문에, 문법 분석 방법은 사전 비교 방법과 동일한 단점을 갖는다. 더욱이, 몇몇 문법 분석 방법은 문법에 맞는 완전한 문장만을 분석할 수 있기 때문에, OCR 문서 내의 키워드는 추출할 수 없다.

통계적인 분석 방법은 문서를 우선 분석하고, 문서에서 충분한 통계적 파라미터들이 축적된 후 특정 원칙에 따른 통계적인 파라미터의 어휘를 추출한다.

도 1은 통계적인 분석 방법에 따른 키워드 추출 방법의 의사 코드(pseudo code)를 보여주는 도면이다. 통계적인 파라미터를 누적하는 가장 쉬운 방법은 어휘의 발생 빈도를 카운트하고, 어휘의 발생 빈도가 소정 범위 내에 속하면 그 어휘를 추출하는 것이다. 이 방법은 어휘 데이터베이스를 이용하지 않기 때문에, 때때로 무의미하거나 불합리한 어휘를 추출할 수 있다. 더욱이, 통계적인 파라미터들의 요구를 충족하지 않는 키워드는 추출되지 않는다.

몇몇 다른 방법들은 상술한 방법들의 조합 또는 변형된 형태가 대부분이다. 예를 들어, 특정 원리 예를 들면, 타이틀, 리스트 내의 단어, 또는 강조된 어휘(대문자, 이탤릭 타입의 단어, 볼드 타입의 단어, 밑줄친 단어 또는 인용 부호 등)를 이용하여 문서로부터 중요한 어휘를 추출하는 것이다. 각 방법의 단점들은 충분히 유추 가능하며, 독특한 조건들을 고려하여 사용되어야 할 것이다.

최근에는 추출된 키워드를 북 인덱스(book index)로 사용하기 위한 중국어 키워드를 자동으로 추출하는 연구가 수행되고 있다. 이 방법은 어휘를 추출하기 위해 전자 사전을 이용하고, 명사를 분석하기 위해 자연어 처리 기술과 함께 통계치를 이용하고 있다. 이 방법의 정확도는 약 63%이고, 에러의 주원인은 잘못된 판단(42%), 불충분한 통계치 특성(39%), 그리고 복잡한 문법 구성으로 인한 무능력(19%) 등이다.

중국어 키워드를 추출하는 또다른 방법은 우선, PAT-트리로 알려진 데이터 구조를 확립하고, 어휘 빈도 등과 같은 통계적 특성으로 키워드를 추출하는 것이다. 이 방법의 정확도는 약 81%이다. 비록 PAT-트리가 정보를 자동으로 추출하는데에는 훌륭하나, 데이터 구조를 확립하는데 많은 시간이 소요된다. 예컨대, 데이터베이스를 확립하는데 일주일이

소요되고, 그것의 사이즈는 600 메가바이트(Magabytes)에 달한다. 이 방법을 효율적으로 사용하기 위해서는 PAT-트리 데이터베이스를 확립하는 속도가 향상되어야 한다.

상술한 바와 같은 방법들은 OCR 또는 음성 인식과 같은 노이지 문서(noisy documents)에서 효율성이 떨어지고, 실제 환경에 적용하기 어렵기 때문에 그러한 문서에 사용되는 것은 적합하지 않다. 예를 들어, 노이지 문서에 포함된 몇몇 중요한 어휘들은 부정확하게 인식될 수 있고, 그러한 에러의 위치는 미리 알 수 없다. 사전 비교 방법은 에러 위치를 미리 알 수 없을 때 무용지물이 된다. 더욱이, 문법에 따르지 않은 문장과 같은 부정확한 어휘를 포함하는 문장에서 문법 해석 방법은 적절하지 않다. 또한, 인식된 문서는 거의 대부분 텍스트 파일이므로, 레이아웃 원리(layout principles)를 이용한 방법은 이러한 환경에 부적합하다. 비록, 신생어(newborn phrases)는 통계적 해석 방법을 이용해서 추출될 수 있으나, 전체 크기가 큰 데이터 샘플은 적절한 통계적 파라미터를 결정하는 것이 요구된다. 그러한 통계적 파라미터를 만족하지 않는 키워드는 중요하더라도 부정확하게 인식되어 추출할 수 없기 때문에 통계적 해석 방법의 이용은 한계가 있다.

음악 데이터에서 키 멜로디를 자동으로 추출하기 위해서는 주의가 요구된다. 작곡 규칙들 가운데 하나는 반복(repetition)이다. 반복된 부분은 작곡자가 강조하기 위해 그리고 청중이 가능한 한 쉽게 기억하도록 하기 위함이다. 이러한 이유로, 반복된 부분은 검색하는데 있어서 매우 중요하다. 따라서, 반복된 부분이 키 멜로디 또는 음악의 주제라고 가정하자. 텍스트 검색과 비교할 때, 음악의 키 멜로디 검색은 텍스트 검색에서의 요약서 또는 키워드와 동일한 효과를 갖는다. 자동 키 멜로디 검색은 상호 작용하는 환경에서 사용될 때 그리고 질문 유발, 질문 확장, 적절한 피드백 등에서 사용될 때 검색 효율을 향상시킬 수 있다.

음악 선율은 스트링(string)으로 표현될 수 있는 멜로디, 리듬 그리고 화음을 갖는다. 그러므로, 반복 스트링을 추출하는 방법을 이용하여 악곡의 중요한 특징으로서의 악곡의 반복된 부분을 추출하는 것이 가능하다.

종래에 이러한 과제에 관한 연구가 진행되어 왔고, 계산 복잡도와 요구되는 메모리 모두 $O(n^2$, 단, n은 음악 데이터의 길이)이다(Hsu, Liu, Chen, 1998). 계산 복잡도를 감소시킬 수 있는 향상된 방법(Hsu, Liu, Chen, 1999)이 제안되었으나, 중복되어 반복된 스트링은 추출될 수 없었다. 중복되지 않고 반복된 스트링을 빠르게 추출하는 방법은 시간 복잡도가 O(n log n)이다. 그러나,ABCABCAABCAABCABCABCA와 같이, 밑줄친 스트링과 볼드-이태릭체로 표현된 스트링으로 ABCABCA가 두 번 중복되어 반복된 스트링을 추출하기 위한 알고리즘은 여전히 빠르지 않다. 이와 같이 반복된 패턴을 포함하는 음악 데이터는 매우 많으므로, 중복적으로 반복된 패턴을 추출할 수 있는 방법이 음악 데이터의 키 멜로디를 추출하는데 더욱 더 적합하다.

반복적인 스트링들(중복되지 않은 스트링과 중복된 스트링)을 추출하기 위한 방법들은 아직 한계가 있다. 예를 들어, 1972년에 Karp, Miller & Rosenberg에 의해 발표된 방법과 1995년 Soldano, Viari & Champesme에 의해 발표된 방법들은 두 가지 관점 (1) 길이가 K인 스트링을 가지는 반복 스트링을 모두 찾는 것과 (2) 가장 긴 길이 L을 갖는 반복 스트링을 찾는 것에서 한계가 있다. 그러나, 이상적인 방법은 여러 가지 중복적으로 반복된 스트링들 가운데 가장 긴 스트링을 모두 찾을 수 있어야 한다. 예를 들어, 멜로디 스트링 EFGABCABCAEFG에서, 반복된 스트링 EFG, ABC, 그리고 BCA은 길이 3을 갖는다. 여기서, 스트링 ABC, BCA는 두 번 반복되고, 스트링 ABCA도 두 번 반복되는데 ABC, BCA는 스트링 ABCA의 서브 스트링이다. 그러므로, 검색에 있어서, 스트링 ABC와 BCA는 스트링 ABCA에 비해 덜 중요하다. 더욱이, 스트링 ABCA가 길이가 4로서 반복되는 스트링 가운데 가장 긴 길이를 가지므로, 나머지 스트링 EFG는 버려진다. 검색에서 스트링 EFG의 역할은 스트링 ABCA에 의해 대체될 수 없기 때문에, 스트링 ABCA 만을 추출하는 것은 불충분하다.

상술한 바와 같은 종래 기술의 단점을 정리하면 다음과 같다.

1. 사전 비교 방법의 결과가 어휘 데이터베이스 내의 정확한 어휘이더라도, 문서에서의 모든 키워드가 추출되었다고는 보장할 수 없다. 이 방법의 또다른 단점은 사전을 유지보수하고, 특정 분야를 포함시키기 위해 많은 (1) 시간과 (2) 노력이 요구된다는 것과 (3) 사전에 포함되지 않은 분야에 이 방법을 적용할 수 없다는 것이다.

2.어휘 데이터베이스에 의존해야 하는 문법 해석을 위해 사용되는 대부분의 프로그램들은 이미 존재한다. 문법 해석 방법은 사전 비교 방법과 동일한 단점을 갖는다. 게다가, 몇몇 문법 해석 방법들은 문법에 따른 완전한 문장만을 해석할 수 있고, 도서 목록 데이터 또는 OCR 노이지 문서 내의 키워드는 추출할 수 없다.

3. 통계적 해석 방법을 이용하여 신생어는 추출될 수 있지만, 이 방법은 적절한 통계적 파라미터를 결정하기 위해 많은 데이터 샘플을 요구한다. 그러한 통계적 파라미터를 충족하지 않는 키워드는 추출될 수 없기 때문에, 통계적 해석의 적용은 매우 제한적이다.

4. 음악 데이터에 있어서, 키 멜로디는 반복된 스트링을 추출하는 방법에 의해서 추출될 수 있고, 이 방법의 복잡도는 $O(n^2)$이다. 복잡도를 줄일 수 있는 또다른 방법이 있으나, 중복적으로 반복된 스트링은 추출할 수 없다는 단점이 있다.

5. 반복 스트링들(중복되지 않은 스트링과 중복된 스트링)을 추출하기 위한 방법들은 예를 들어, 스트링 길이가 K인 반복 스트링 또는 최대 길이가 L인 반복 스트링을 추출하는데 있어서 한계가 있다.

발명이 이루고자 하는 기술적 과제

따라서, 본 발명의 목적은 상술한 제반 문제점을 해결하기 위해 제안된 것으로, 디지털 문서의 키워드를 자동으로 추출하는 방법을 제공하는데 있다. 이 방법에 따르면, 최대 중복적으로 반복된 최대 패턴들 모두를 추출할 수 있다. 이 방법은 어떠한 제약없이 캐릭터 심벌로 표현된 모든 디지털 문서에 적용될 수 있다.

발명의 구성 및 작용

상술한 바와 같은 본 발명의 목적을 달성하기 위한 본 발명의 특징에 의하면, 디지털 문서의 키워드를 자동으로 추출하는 방법이 제공된다. 이 문서는 엘리먼트들과 적어도 한 바이트로 구성된 복수 개의 엘리먼트들 가운데 하나를 포함한다. 상기 엘리먼트가 나타나는 횟수는 발생 빈도이다. 엘리먼트가 추출될 때, 상기 엘리먼트의 발생 빈도는 드레솔드 값보다 크다. 분리 인자(separator)는 기준 마크(reference mark)로 사용된다. 디지털 문서는 적어도 하나의 리스트 엘리먼트를 포함하는 리스트 데이터 구조로 변환되고, 결합 영역은 비어있는 상태로 설정된다.

상기 결합 영역이 비어있는 상태로 설정된 후, 상기 리스트 엘리먼트들은 순차적으로 인출(taken out)된다. 인출된 엘리먼트가 분리 인자인 경우, 인출된 리스트 엘리먼트는 무시(skip)된다. 인출된 리스트 엘리먼트의 발생 빈도가 상기 드레솔드 값보다 크고, 인출된 리스트 엘리먼트와 근접하게 배열된 연이은 다음 리스트 엘리먼트의 발생 빈도가 상기 드레솔드 값보다 클 때, 결합 프로세스가 수행되어 결합 영역에 저장됨으로써 새로운 엘리먼트가 획득된다. 반면, 인출된 리스트 엘리먼트의 발생 빈도가 드레솔드 값보다 클 때 그리고, 인출된 리스트 엘리먼트 및 인출된 리스트 엘리먼트와 인접한 이전에 인출된 리스트 엘리먼트간의 결합 프로세스가 수행되지 않았을 때, 상기 리스트 엘리먼트는 최종 영역에 저장되고, 결합 영역의 마지막에 위치하는 마지막 새로운 엘리먼트가 상기 분리 인자가 아닐 때 상기 분리 인자가 결합 영역의 마지막에 부가된다.

다음, 상기 결합 영역은 리스트 데이터 구조로 변환되고, 중지 조건이 만족될 때까지 상기 단계들이 반복적으로 수행된다. 마지막으로, 키워드 검색에 주어진 디스플레이 조건에 따라 상기 리스트 엘리먼트를 디스플레이한다.

상술한 바와 같은 방법에 의하면, 본 발명은 다음과 같은 장점들을 갖는다.

1. 신생 어휘, 전문 어휘, 또는 개인, 장소 및 단체의 이름이 추출될 수 있다.

2. 본 발명에 의하면, 문서는 문법에 따르지 않아도 되며, OCR 문서, 음성 인식 문서와 같은 노이지 문서(noisy document)에도 적용될 수 있다.

3. 검색을 위해 사전, 어휘 데이터베이스, 문법 분석기 등과 같은 특별한 자원들을 유지 보수하기 위한 종래의 많은 노력들이 필요치 않다.

4. 추출된 키워드의 길이는 제한되지 않는다. 모든 서브스트링들이 체크되지 않고, 짧은 서브 스트링으로부터 긴 스트링이 결합됨과 동시에 부적합한 서브 스트링은 삭제되므로, 추출 속도가 매우 빠르게 향상된다.

5. 데이터는 결정 원리(deterministic principles)를 이용하여 추출되므로, 비록 통계적 특성이 매우 낮더라도(예를 들면, 발생 빈도가 2인 경우) 키워드가 추출될 수 있다.

6. 본 발명은 텍스트 문서뿐만 아니라, 음악, 음성, 비디오, 이미지 시퀀스, 시간 시퀀스, DNA 시퀀스 등과 같이 스트링 또는 순서적인 셋으로 표현 가능한 데이터에도 적합하다. 디지털 데이터는 추출되기 전에 캐릭터 심볼로 변환된다.

7. 계산 복잡도(computing complexity)의 최상의 이론적인 값은 O(L*N) 이다. 여기서, L은 가장 긴 반복 스트링의 길이이고, N은 입력 스트링의 길이이다. 최상의 이론적인 값은 해시 함수 및 리스트를 이용해서 접근될 수 있다.

8. 추출 결과는 다른 발전된 계산(예를 들면, 자동 인덱싱, 인덱싱 사전의 자동 수립, 자동 발췌, 자동 분류, 관련성 피드백, 자동 필터링, 개념 검색, 용어 제안, DVD 관련성 피드백, 다이내믹 사전 분류, 정보 시각화, 그리고 다른 관련된 적용 분야)에 제공될 수 있다.

(실시예)

이하 본 발명에 따른 실시예를 첨부된 도면 도 2 내지 도 11을 참조하여 상세히 설명한다.

용이하게 설명하기 위하여 아시아어 문자, 서방어 문자, 또는 다른 어떤 문자도 순서적인 리스트에 따라 표시한다. 그러므로, 스트링 " E F G A B C A B C A E F G" 는 음악, 음성, 비디오, 이미지 순서, 시간 순서, DNA 순서 등의 정보를 표현한 것이다.

본 발명은 개별적인 엘리먼트(element)에 의해 시작되고, 중복된 또는 중복되지 않은 여러 가지 최대 반복 스트링들 모두가 추출될 때까지 본 발명의 조건에 따른 엘리먼트들의 결합(combine), 삭제(erase) 및 적용(accept)의 반복적인 프로세스에 의해 진행된다. 그러므로, 추출된 스트링의 최대 길이는 미리 설정되지 않으며, 서브 스트링의 공통 정보는 계산되지 않는다. 더욱이, 프로세스가 진행되는 동안 서브 스트링들 모두가 열거될 필요는 없다. 조건을 충족하지 못하는 모든 서브 스트링들은 프로세스 도중에 삭제되고, 더 이상 처리되지 않는다. 그러므로, 서브 스트링의 검사 횟수는 현저히 감소된다.

본 발명은 적어도 세 개의 메인 프로세싱 단계들을 포함한다. 우선, 입력은 적절한 데이터 구조로 변환된다. 두 번째 단계에서, 중지 조건(stop condition)이 만족될 때까지 스트링은 결합, 소거 및 적용된다. 세 번째 단계에서는 두 번째 단계로부터의 결과를 정렬(sort)하고 필터링(filtering)한다. 처음 두 단계 동안 중복된 또는 중복되지 않은 여러 가지의 최대 반복 스트링들 모두가 추출되고, 이 결과는 적용 분야(예를 들면, 문자, 음악 또는 다른 매체)에 구애되지 않는다. 세 번째 단계는 적용된 분야의 요구를 만족시키기 위해서, 필터링 프로세스를 적용된 분야에 따른 처음 두 단계들로부터 획득된 결과에 적용시킨다.

도 2는 본 발명의 바람직한 실시예에 따른 처음 두 단계가 수행되는 경우 의사 코드(pseudo code)의 일부를 보여주는 도면이다. 실제에 있어서, 상기 방법들을 수행하는데에는 여러 가지 방법들이 있고, 가장 중요한 것은 다양한 방법들의 결과이고, 그 결과는 스트링의 발생 빈도를 기록하고 접근하는 것으로, 도 2에 도시된 바와 같이 함수 Freq()로 표현한다.

스트링의 발생 빈도에 접근하기 위한 가장 빠른 방법은 해시 함수(hash function)를 사용하는 것이다. 많은 키 값 쌍들은 해시 함수에 의해서 저장되고, 그 값들은 대응하는 키에 따라서 접근될 수 있다. 이론적인 평균 접근 시간은 소정 시간 즉, O(1)이고, 키 값 쌍들의 개수와는 무관하다. 그러나, 해시 함수는 매우 긴 키들에 있어서는 부적합하다. 중국 음악에 있어서 가장 긴 키 멜로디는 전체 악곡 길이의 약 20%에 달하고, 서양 클래식 음악의 가장 긴 키 멜로디는 전체 악곡 길이의 약 14%에 달하므로, 해시 함수는 그러한 악곡을 처리하는데에는 부적합하다.

이러한 문제를 해결하기 위하여, 위치 리스트(position list)를 사용해서 긴 스트링의 발생 빈도에 용이하게 접근하는 방법이 있다. 짧은 스트링을 빠르게 다루는데에는 해시 함수가 사용되고, 긴 스트링을 처리하는데에는 위치 리스트를 사용함으로써 해시 함수의 단점을 피할 수 있다. 처리 시간은 두 방법을 스위칭하는데 소요되는 이론적인 시간에 근접한다.

도 3은 본 발명의 바람직한 실시예에 따른 해시 함수와 스트링의 발생 빈도를 이용하는 프로세스를 보여주는 도면이다. 도 3에 있어서, 스트링의 콜론(:) 다음의 숫자는 스트링의 발생 빈도를 나타낸다. 본 발명의 바람직한 실시예에 따른 첫 번째 단계는 문자(엘리먼트)를 입력 데이터로 변환해서 리스트 엘리먼트들(listed elements)을 포함하는 리스트 데이터 구조(listed data structure)에 넣는 것이다. 이 변환 프로세스는 입력 순서에 따라 진행되고, 도 3에 도시된 해시 함수를 이용하여 엘리먼트의 발생 빈도를 계산한다. 두 번째 단계의 루프로 진행하기 전에, 마지막 엘리먼트가 처리될 때 절차(procedure)를 용이하게 하기 위하여, x로 표시된 분리 인자(separator)는 리스트 데이터 구조의 마지막에 부가된다.

비록 본 발명의 바람직한 실시예에 따른 엘리먼트가 단일 문자이나, 이 예는 설명을 용이하기 위한 일 예일 뿐, 본 발명은 단일 문자로 구성된 엘리먼트를 해석하는데 한정되지 않는다.

두 번째 단계에서, 리스트 엘리먼트들은 순서대로 인출(taken-out)된다. 두 개의 인접한 엘리먼트들 가운데 하나는 이미 인출된 하나의 엘리먼트이고, 다른 하나는 연이어 인출될 하나의 엘리먼트 또는 이전에 인출된 엘리먼트이다. 인출된 엘리먼트는, 결합 조건(combination condition)에 따라 다음 루프의 어법을 위하여 연이은 엘리먼트와 결합되어 새로운 엘리먼트를 형성하거나, 적용 조건(acceptance condition)에 따라 추출된 엘리먼트로 받아들여지거나, 또는 삭제 조건(erasure condition)에 따라 삭제된다.

결합 조건은 인접한 두 개의 엘리먼트들의 발생 빈도(즉, Freq(LIST[I]) 및 Freq(LIST[I+1]))로서, 미리 설정된 드레솔드 값보다 커야 한다. 상기 드레솔드 값은 데이터의 크기 또는 총 반복 횟수이다. 이 드레솔드 값은 일반적으로 1로서, 반복 스트링의 발생 빈도가 1 이상이면 추출될 수 있음을 나타낸다.

결합 프로세스는, 연이은 엘리먼트의 마지막 문자가 인출되고, 인출된 엘리먼트의 끝에 결합되어 새로운 엘리먼트가 생기는 것이다. 다음, 새로운 엘리먼트의 발생이 해시 함수에 의해 누적된다.

전체 계산(global computation)은 해시 함수가 수행될 때 요구되고, 로컬 계산(local computation)은 다른 부분의 프로시져가 수행될 때 요구된다. 인출된 엘리먼트와 인접한 로컬 데이터는 거의 모든 계산에 사용된다. 제 1 및 제 2 단계의 시간 복잡도는 다음과 같이 해석된다. 디지털 문서는 문법적으로 분석되고, 리스트 데이터 구조로 변환되며, 발생 빈

도는 제 1 단계에서 계산된다. 해시 함수가 소정 시간을 소모할 때, 이 단계에서는 평균적으로 입력 스트링의 길이인 O(N)가 요구된다. 리스트 데이터 구조는 단계 2에서 반복적으로 처리되고, 리스트된 데이터 구조의 최대 길이는 N이고, 시작할 때에만 나타난다. 그 이후, 리스트 데이터 길이는 급격히 감소된다. 리스트 엘리먼트가 처리될 때 상기 해시 함수가 사용되고, 로컬 데이터가 액세스된다. 그러므로, 리스트 데이터 구조가 처리될 때 요구되는 평균 시간은 O(N) 이다. 길이가 L인 반복 스트링은 상술한 바와 같은 과정을 L번 반복함으로써 추출될 수 있으므로, 반복 스트링의 최대 길이가 L일 때, 단계 2의 계산 복잡도는 O(L*N) 이다. 이는 평균 액세스 시간이 O(1)가 된다고 가정할 때 얻어진 결과이다.

상술한 바와 같이, 해시 함수는 스트링의 길이가 증가할수록 최상의 이론적 시간에 도달하는데 어려움이 따른다. 그러므로, 이러한 종류의 환경에 적합한 또다른 방법이 제공된다.

도 4는 포지션 리스트(position list)를 이용한 프로세스를 보여주는 도면이다. 도 4에서, 포지션 리스트의 엘리먼트들 각각은 리스트 데이터 구조의 엘리먼트들에 대응하는 포지션들의 결합(combination)을 포함한다. 예를 들어, 리스트 데이터 구조(E, F, G, A, B, C, A, B, C, A, E, F, G)의 포지션 리스트는 (1 11, 2, 12, 3 13, 4 7 10, 5 8, 6 9, 4 7 10, 5 8, 6 9, 1 11, 2, 12, 3, 13)이고, 엘리먼트 E는 포지션 1 및 11에 나타나며, 엘리먼트 F는 포지션 2 및 12에 나타난다. 도 3과 같이, 프로세스가 두 번 반복될 때(즉, EFG:2, x, ABC:2, BCA:2, CAB:1, ABC:2, BCA:2, x, EFG:2) 포지션 리스트는 (1 11, x, 4 7, 5 8, 6, 4 7, 5 8, x, 1 11)이 된다.

엘리먼트의 발생 빈도는 리스트 엘리먼트를 포함하는 포지션의 카운팅 수에 의해서 표시될 수 있다. 단계 1에서 발생 빈도가 리스트 엘리먼트에 저장 및 부가될 때, 빈도를 갱신하기 위한 로컬 감소 프로세스(local decrement process)만이 요구된다. 결합 프로세스는 두 개의 서로 인접한 리스트 엘리먼트들의 포지션에서 발견할 수 있다. 예를 들어, 도 4에 도시된 포지션 리스트 " LIST" 내의 " 4 7 10" 과 " 5 8" 두 개의 포지션의 결합은 두 개의 엘리먼트 " A" 및 " B" 의 결합을 나타낸다. 리스트 엘리먼트 " A" 는 포지션 4, 7 및 10에 나타나고, 그리고 리스트 엘리먼트 " B" 는 포지션 5 및 8에 나타난다. 리스트 엘리먼트들 " A" 및 " B" 는 포지션 4 5 및 7 8에서 두 번 서로 인접한다. 그리고, 두 개의 리스트 엘리먼트들 " A" 및 " B" 는 스타팅 포지션 4 및 7에서 순차적으로 두 번 나타난다. 그러므로, 포지션 " 4 7" 의 결합이 저장됨으로써, " A" 및 " B" 를 결합하기 위한 결합 프로세스가 완료된다.

꺼내어진 리스트 엘리먼트가 추출된 후, 꺼내어진 리스트 엘리먼트는 최종 영역에 저장된다. 그러므로, 꺼내어진 리스트 엘리먼트는 루프(loop)가 실행되는 횟수에 따라서 원래의 디지털 문서의 서브 스트링으로 변환된다. 꺼내어진 리스트 엘리먼트에 대응하는 서브 스트링의 길이는 각 루프가 실행된 후 1 만큼 증가되므로, 서브 스트링의 길이는 루프가 실행된 횟수를 통해 표시될 수 있다. 예를 들어, 도 4에 도시된 바와 같이, 루프가 3 번 실행된 후 추출된 리스트 엘리먼트 " 1 11" 은 자신의 원래 서브 스트링 " EFG" 에 재저장될 수 있다. 유사하게, 최종 영역의 리스트 엘리먼트들 각각은 원래 디지털 문서의 서브 스트링으로 다시 변환될 수 있다.

글로벌 계산은 단계 1에서 리스트 데이터 구조가 포지션 리스트로 변환될 때에만 필요하고, 로컬 계산은 다른 프로시져가 실행될 때 필요하다. 로컬 데이터는 모든 계산에서 사용된다. 이 방법의 최대 장점은 발생 빈도가 리스트 엘리먼트 스트링에 구성된 키를 통해서가 아니라 리스트 엘리먼트에 저장된 위치의 번호를 통해 획득될 수 있다는 것이다. 리스트 엘리먼트 내에 저장된 데이터는 자신의 스트링 분만 아니라 리스트 엘리먼트가 나타나는 위치를 갖는다. 그러므로, 여기서는 스트링의 길이는 무의미하게 된다. 최악의 경우는 리스트 엘리먼트들이 여러 위치들에 나타나서 그들의 위치를 저장하기 위한 메모리 공간을 많이 차지하는 것이다. 그러나, 리스트 엘리먼트의 반복 빈도는 반복 스트링의 길이에 따라 감소하고, 그에 따라서 저장되어야 할 위치도 감소된다. 관찰된 음악 데이터에 있어서, 대부분 모든 중요한 멜로디는 2 ~ 3회 반복된다. 예를 들어, 상술한 바와 같이 중국 음악의 가장 긴 키 멜로디는 음악 전체 길이의 20%를 차지하

고, 서약 클래식 음악의 가장 긴 키 멜로디는 음악 전체 길이의 대략 14%를 차지한다. 가장 긴 키 멜로디가 2번 반복되는 경우, 키 멜로디는 음악 전체 길이의 약 40%를 차지하게 된다. 그러므로, 중요한 키 멜로디들 각각이 추출되는 경우, 2 내지 3 발생 위치들을 저장하는 것만으로도 충분하다. 포지션 리스트에 저장되는 포지션의 총합은 대략 2N 또는 3N (즉, O(N))이다.

도 5는 본 발명의 바람직한 실시예에 따른 의사 코드를 보여주는 도면이다. 루프 횟수가 적고 추출된 스트링의 길이가 짧은 경우, 최적의 이론적인 계산 시간을 위해 해시 함수가 적용된다. 길이가 긴 스트링을 추출하기 위해서 루프의 횟수가 증가하는 경우, 최적의 이론적인 계산 시간을 위해 포지션 리스트가 적용된다. 이 실시예에 적용된 방법은 이론적으로 최적의 값 O(LN)에 근접한 계산 시간을 획득하기에 용이한 방법을 제공한다.

도 5에 도시된 바와 같이, 단계 2에서 추출된 키워드들은, 텍스트 데이터의 무의미한 반복 스트링을 포함하거나 또는 음악 데이터의 불필요한 반복 스트링을 포함한다. 그러므로, 적용 분야의 요구를 만족하지 않는 반복 스트링들은 단계 3의 디스플레이 조건에 따라서 필터링되어야 한다.

필터링 방법은 순환적인 스트링의 시작 또는 끝부터 스트링의 시작 또는 끝에 포함되지 않는 정지 단어(stop words; 예를 들어, " of" , " on" , 또는 " the" 와 같은 전치사, 대명사 또는 다른 기능적인 단어)를 삭제한다. 정지 단어는, 문장에서 각 단어가 매우 유용한 중국어에서는 찾을 수 없다. 그러므로, 정지 단어는 몇몇 분야에서만 있을 수 있다. 중국어 환경에서 정지 단어는 " 的" 이라는 단어가 적합하다. 도 6A 및 도 6B는 중국어와 영어를 포함하는 환경에 본 발명을 적용하는 것을 보여주는 도면이다(그러나, 이 실시예에서는 영어 어휘만이 표시되었을 뿐 단일 영어 단어는 사용되지 않았다). 여섯 번째 구 " FGDC 的 Digital Geospatial metadata" 는 중국어와 영어를 모두 포함하는 키워드이다. 그러므로, 이 방법은 여러 종류의 언어들을 포함하는 환경에 유용하게 제공된다.

본 발명은 다양한 언어에 적용될 수 있다. 이것을 보여주기 위하여 중국어와 영역 문서로 디자인된 본 발명의 실시예를 한국어 문서에 직접 적용하는 것을 다음에서 보여준다. 또다른 실시예에서는, 스트링의 첫 번째 바이트를 체크함과 동시에 중국어 문자와 영어 문자를 판별하는 방법이 설명된다. 스트링의 첫 번째 바이트가 십진수 127을 초과할 때, 첫 번째 바이트는 페치(fetch)되고, 2-바이트 코드 형태의 다음 중국어 문자 또는 1-바이트의 영어 문자와 결합된다. 여기서, 상기 중국어 문자와 영어 문자들은 상술한 디지털 문서의 엘리먼트들이다. 그 엘리먼트가 중국어 문자인지 또는 영어 문자인지의 여부가 판별된 후, 그 엘리먼트가 구두점 마크(punctuation mark)인지의 여부가 판별된다. 상기 엘리먼트가 구두점 마크가 아닌 경우, 그 엘리먼트는 전송된다. 이 낱말 입력 스트링의 방법은 다른 언어(예를 들면, 한국어)로 된 문서에서 키워드를 추출하기 위해 직접 적용하는데 있어서 두가지 한계가 있다. 즉, (1) 한국어에서 상기 구두점 마크는 리갈 문자(legal characters)로 중국어의 구두점 마크와는 다르므로, 추출된 스트링은 한국어의 구두점 마크를 포함할 수 있다. (2) 중국어의 구두점 마크는 한국어의 리갈 마크이고, 중국어의 구두점 마크는 삭제되기 때문에, 옳지 않은 한국어의 구가 추출될 수 있다.

도 7A 내지 도 7C는 한국어를 추출할 때 본 발명을 적용한 결과를 보여주는 도면이다. 도 7C는 도 7B의 한국어 키워드를 번역한 것으로, 한국어 키워드의 길이에 따라 소트된 것이다. 그것들로부터 알 수 있는 바와 같이, 상술한 바와 같은 두 가지 한계가 실제 존재하더라도 한국어의 어휘 데이터베이스는 사용되지 않고, 한국어 문서는 미리 분석되지 않으며, 한국어 키워드는 여전히 추출될 수 있다. 그러므로, 본 발명은 어떠한 언어에도 의존적이지 않다는 특징이 있다.

잘못된 키워드는 회피하기 어렵다. 본 발명의 정확도를 판단하기 위하여, 가능한 한 최근의 많은 중국어 책들과 영어 책들을 대상으로 실험하였다. 35만 개의 제목들로부터 약 11만 스트링들이 추출되었고, 추출된 용어로부터 8982 개의 샘플들이 획득되었다. 수작업으로 샘플들을 조사한 결과, 문법에 따른 어휘의 수는 8123 개였고, 문법에 따르지 않은 어휘의 수는 859 개였다. 그러므로, 정확도는 90.4%이다. 또다른 실험은 13035 개의 신문들로부터 용어를 추출하기 위해 수행되었고, 66500 개의 phase들이 추출되었다. 수작업으로 조사한 결과, 문법에 따른 어휘의 수는 57389 개였

고, 문법에 따르지 않은 어휘는 911 개였다. 정확도는 86.3%이다.

상술한 바와 같이, 자동으로 키워드를 추출하는 것은 정보 복구에서의 용어 제안(term suggestion)과 용어 관련성 피드백(term relevance feedback)에 적용될 수 있다. 도 8은 OCR 문서로부터 키워드를 추출할 때 본 발명을 적용한 결과를 보여주는 도면이다. 도 8을 참조하면, 적절한 단어의 데이터베이스를 형성하기 위하여, 1300 개의 OCR 문서로부터 40709 어휘들이 추출되었다. 데이터베이스에서 검색하기 위하여 " 鄕村幹部(이 단어는 중국어이다)" 단어가 사용될 때, " 鄕村幹部" , " 農村幹部" , " 鄕村幹都" , 및 工農幹部" 네 단어들이 발견될 수 있으며, 네 개의 단어들은 OCR 문서에서 검색하기 위해 컴퓨터로 전송된다. 네 개의 단어들 가운데 첫 번째 단어가 매치(match)된다. 두 번째 및 네 번째 단어들의 뜻은 찾는 단어의 뜻과 근접하다. 세 번째 단어의 마지막 글자는 부정확하게 인식된다. 그러나, 세 번째 단어는 OCR 문서에서 두 번 나타났지만 자동으로 추출되었고, 잘못된 데이터의 가능성을 감소시킨다.

도 9는 또다른 OCR 문서로부터 키워드를 추출할 때 본 발명을 적용한 결과를 보여주는 도면이다. 데이터베이스의 내용은 모든 OCR 문서를 조사하지 않고도 대충 이해될 수 있고, 이는 화상 데이터에 의해 달성될 수 있는 기능들 가운데 하나이다.

본 발명에 의해 제공된 키워드 데이터베이스는 용어 제안뿐만 아니라 다이내믹 사전에도 제공될 수 있다. 도 10A 내지 도 10B는 다이내믹 사전을 구성하는데 있어서 본 발명의 결과를 이용하는 방법을 보여주는 도면이다. 결과를 검색하기 위한 키워드로 " DVD" 가 사용되는 경우, 컴퓨터 시스템은 " DVD" 가 문서에서 482회 나타난 것으로 응답한다. 이는 문서들을 하나씩 조사하는데 시간을 소모한다. 그러나, 본 발명에 의해 구성된 키워드 데이터베이스는 " DVD player" 와 같은 몇가지 특정 용어들을 제공할 수 있으므로, 검색 범위를 빠르게 감소시킨다.

도 11은 음악 검색에 본 발명이 적용된 것을 보여주는 도면이다. 간단히 표시한 기호 " 1 7 - 1 3 2 1" 은 아베 마리아(Ave Maria)의 일부 멜로디 " 도 시 도 미 레 도" 를 나타낸다. 이는 음악에서 반복율이 매우 높다. 반복 멜로디가 추출되고 비교에 사용될 때, 비교에 필요한 데이터는 감소된다. 동일한 반복 데이터를 비교하는데 있어서 자원은 소모되지 않고, 더욱 복잡한 계산들이 큰 허용 오차에 적용될 수 있다. 적은 데이터를 포함하는 키 멜로디는 전송되고 청취되기 때문에 검색 결과가 조사될 때, 효율은 증대된다. 그러므로, 자동으로 중요한 요점을 추출하는 것은 음악과 같은 멀티미디어 데이터에 있어서 매우 중요하다.

본 발명의 범위는 개시된 실시예들에 한정되지 않는다. 본 발명은 본 발명에 의해 생성된 키 어휘 데이터베이스에 의해서, 자동 인덱싱, 인덱싱 디렉토리 자동 생성, 자동 발췌, 자동 분류, 관련성 피드백(relevance feedback), 자동 필터링, 개념 검색, 용어 제안, 용어 관련성 피드백과 같은 분야에 적용될 수 있다.

예시적인 바람직한 실시예들을 이용하여 본 발명을 설명하였지만, 본 발명의 범위는 개시된 실시예들에 한정되지 않음이 이해될 것이다. 오히려, 본 발명의 범위에는 다양한 변형 예들 및 그 유사한 구성들을 모두 포함될 수 있도록 하려는 것이다. 따라서, 청구 범위는 그러한 변형 예들 및 그 유사한 구성들 모두를 포함하는 것으로 가능한 폭넓게 해석되어야 한다.

발명의 효과

이상과 같은 본 발명에 의하면, 빠른 속도와 향상된 기술로 디지털 문서의 중요한 요점을 자동으로 추출하는 방법이 제공된다. 본 발명은 실제 환경과 어떤 언어에도 적용될 수 있으며, 어휘 데이터베이스의 뛰어난 품질은 많은 노력과 시간을 소모하지 않는다.

(57) 청구의 범위

청구항 1.

적어도 한 바이트로 구성된 복수 개의 엘리먼트들과 발생 빈도로 알려진 각 엘리먼트들이 나타난 횟수를 포함하는 디지털 문서의 키워드를 자동으로 추출하는 방법에 있어서:

상기 엘리먼트의 발생 빈도가 드레솔드 값보다 클 때 그 엘리먼트는 추출되고, 분리 인자(separator)는 기준 마크(reference mark)로 사용되고;

(A) 상기 디지털 문서를 적어도 하나의 리스트 엘리먼트를 포함하는 리스트 데이터 구조로 변환하는 단계와;

(B) 결합 영역을 비어있는 상태로 설정하는 단계와;

(C) 리스트 엘리먼트를 순차적으로 인출하는 단계와;

인출된 엘리먼트가 분리 인자인 경우, 인출된 리스트 엘리먼트는 무시되고;

인출된 리스트 엘리먼트의 발생 빈도가 상기 드레솔드 값보다 크고, 인출된 리스트 엘리먼트와 근접하게 배열된 연이은 다음 리스트 엘리먼트의 발생 빈도가 상기 드레솔드 값보다 클 때, 결합 프로세스가 수행되어 상기 결합 영역에 저장됨으로써 새로운 엘리먼트가 획득되며;

인출된 리스트 엘리먼트의 발생 빈도가 드레솔드 값보다 클 때 그리고, 인출된 리스트 엘리먼트 및 인출된 리스트 엘리먼트와 인접한 이전에 인출된 리스트 엘리먼트간의 결합 프로세스가 수행되지 않았을 때, 상기 리스트 엘리먼트는 최종 영역에 저장되고, 결합 영역의 마지막에 위치하는 마지막 새로운 엘리먼트가 상기 분리 인자가 아닐 때 상기 분리 인자가 결합 영역의 마지막에 부가되며;

(D) 상기 결합 영역을 리스트 데이터 구조로 변환하는 단계; 및 중지 조건이 만족될 때까지 상기 (B) - (C) 단계를 반복적으로 수행하는 단계; 그리고

(E) 키워드 검색에 주어진 디스플레이 조건에 따라 상기 리스트 엘리먼트를 디스플레이하는 단계를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 2.

제 1 항에 있어서,

상기 결합 단계는 연이은 리스트 엘리먼트의 마지막 문자를 인출하고, 인출된 리스트 엘리먼트의 끝에 상기 마지막 문자를 부가하여 새로운 엘리먼트를 생성하고, 그리고 상기 새로운 엘리먼트는 결합 영역의 끝에 부가되는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 3.

제 1 항에 있어서,

상기 중지 조건은 리스트 데이터 구조 내의 분리 인자를 제외하고, 리스트 엘리먼트들의 개수가 2 보다 작을 때인 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 4.

제 1 항에 있어서,

상기 리스트 엘리먼트들 각각은 대응하는 엘리먼트의 위치의 결합을 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 5.

제 4 항에 있어서,

위치의 결합은 디지털 문서에 나타난 엘리먼트들 각각의 위치를 나타내는 위치들의 개수를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 6.

제 1 항에 있어서,

상기 리스트 엘리먼트들 각각은 엘리먼트들과 대응하는 발생 빈도를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 7.

제 1 항에 있어서,

디스플레이 조건에 따라 상기 리스트 엘리먼트를 디스플레이하는 단계는,

최종 영역내 리스트 엘리먼트의 시작과 끝으로부터 중지 단어를 삭제하는 단계를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 8.

제 7 항에 있어서,

상기 중지 단어는 전치사, 대명사 그리고 기능적인 단어를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 9.

제 1 항에 있어서,

상기 디지털 문서를 리스트 데이터 구조로 변환하는 단계는,

엘리먼트의 첫 번째 바이트에 의해서, 엘리먼트를 아시아 문자와 서양 문자로 구성되는 그룹들 가운데 하나로 판별하는 단계를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 10.

제 9 항에 있어서,

상기 엘리먼트가 아시아 문자와 서양 문자로 구성되는 그룹들 가운데 하나로 판별된 후, 상기 엘리먼트가 구두점 마크인 지의 여부가 판별되는 단계를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 11.

제 10 항에 있어서,

상기 엘리먼트가 구두점 마크로 판별되는 경우, 상기 분리 인자는 리스트 데이터 구조에 부가되는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 12.

제 11 항에 있어서,

디스플레이 조건에 따라서 리스트 엘리먼트를 디스플레이하는 단계는,

상기 최종 영역 내 리스트 엘리먼트의 시작과 끝으로부터 중지 단어를 삭제하는 단계를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 13.

제 12 항에 있어서,

상기 중지 단어는 전치사, 대명사, 그리고 기능적인 단어를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 14.

제 1 항에 있어서,

상기 엘리먼트는 아시아 문자, 서양 문자, 서양 단어, 그리고 음악적 부호를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 15.

제 1 항에 있어서,

상기 분리 인자는 리스트 데이터 구조의 끝에 부가되고, 상기 분리 인자의 발생 빈도는 결합 영역이 비어있는 상태 또는 그 이후 상기 드레솔드 값보다 작도록 설정되는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 16.

제 1 항에 있어서,

상기 디지털 문서는 스트링과 순서적인 세트를 포함하는 정보로 표현될 수 있는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 17.

제 16 항에 있어서,

상기 정보는 문자, 음악, 음성, 비디오, 영상 시퀀스, 시간 시퀀스 그리고 DNA 시퀀스를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

청구항 18.

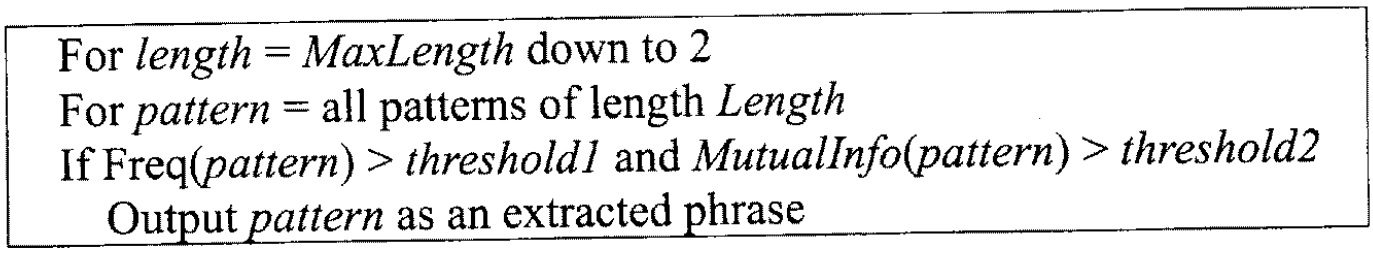
제 1 항에 있어서,

상기 최종 영역내의 상기 리스트 엘리먼트들 가운데 키워드를 검색하는 단계는.

자동 인덱싱, 인덱싱 사전의 자동 수립, 자동 발췌, 자동 분류, 관련성 피드백, 자동 필터링, 개념 검색, 용어 제안, DYD 관련성 피드백, 다이내믹 사전 분류, 그리고 정보 시각화를 포함하는 것을 특징으로 하는 디지털 문서의 키워드를 자동으로 추출하는 방법.

도면

도면 1



도면 2

```
1.    Convert the input into a LIST.
2.    Do Loop
2.1   Set MergeList to empty.
2.2   Put a separator to the end of LIST as a sentinel and
      set the occurring frequency of the separator to 0.
2.3   For I from 1 to NumOf(LIST) – 1 step 1, do
        If LIST[ I ] is the separator, Go to Label 2.3.
        If Freq(LIST[ I ]) > threshold and Freq(LIST[ I +1]) >
threshold
        then
          Merge LIST[ I ] and LIST[ I +1] into Z
          Put Z to the end of MergeList.
        else
          If Freq(LIST[ I ]) > threshold and
            LIST[ I ] did not merge with LIST[ I - 1]
         then
            Save LIST[ I ] in FinalList.
            If the last element of MergeList is not the separator,
            Then
              Put the separator to the end of MergeList.
      End of For loop
2.4   Set LIST to MergeList.
      Until NumOf(LIST) < 2.
```

도면 3

Example 1: Given : E F G A B C A B C A E F G. Let threshold=1, separator=x

Step 1: Create a list of single characters:

LIST = (E:2, F:2, G:2, A:3, B:2, C:2, A:3, B:2, C:2, A:3, E:2, F:2, G:2)

Step 2:

After 1st iteration :

MergeList = (EF:2, FG:2, GA:1, AB:2, BC:2, CA:2, AB:2, BC:2, CA:2, AE:1,EF:2, FG:2)

FinalList = ( )

After 2nd iteration :

MergeList = (EFG:2, x, ABC:2, BCA:2, CAB:1, ABC:2, BCA:2, x, EFG:2)

FinalList = ( )

After 3rd iteration :

MergeList = (x, ABCA:2, x, ABCA:2)

FinalList = (EFG:2)

After 4th iteration :

MergeList = (x)

FinalList = (EFG:2, ABCA:2)

도면 4

Example 2: Given : E F G A B C A B C A E F G. Let threshold=1, separator=x

Step 1: Create a position list :

LIST=(1 11, 2 12, 3 13, 4 7 10, 5 8, 6 9, 4 7 10, 5 8, 6 9, 4 7 10, 1 11, 2 12, 3 13)

Step 2:

After 1st iteration :

MergeList = (1 11, 2 12, 3, 4 7, 5 8, 6 9, 4 7, 5 8, 6 9, 10, 1 11, 2 12)

FinalList = ( )

After 2nd iteration :

MergeList = (1 11, x, 4 7, 5 8, 6, 4 7, 5 8, x, 1 11)

FinalList = ( )

After 3rd iteration :

MergeList = (x, 4 7, x, 4 7)

FinalList = (1 11) = (E F G : 2)

After 4th iteration :

MergeList = (x)

FinalList = (1 11, 4 7) = (E F G : 2, A B C A : 2)

도면 5

1. Convert the input string into a *LIST* of single characters and accumulate their occurring frequencies.

2. Do Loop
2.1 Set *MergeList* to *empty*.
2.2 Put a *separator* to the end of *LIST* as a sentinel and set the occurring frequency of the *separator* to 0.
If the length of the element in *LIST* is less than *SomeNumber*,
Then
Implement step 2.3 by use of hash functions.
Else
Implement step 2.3 with the position list method.
2.4 Set *LIST* to *MergeList*.
Until NumOf(*LIST*) < 2.

3. Filter the *FinalList* and sort the result according to some criteria.

도면 6a

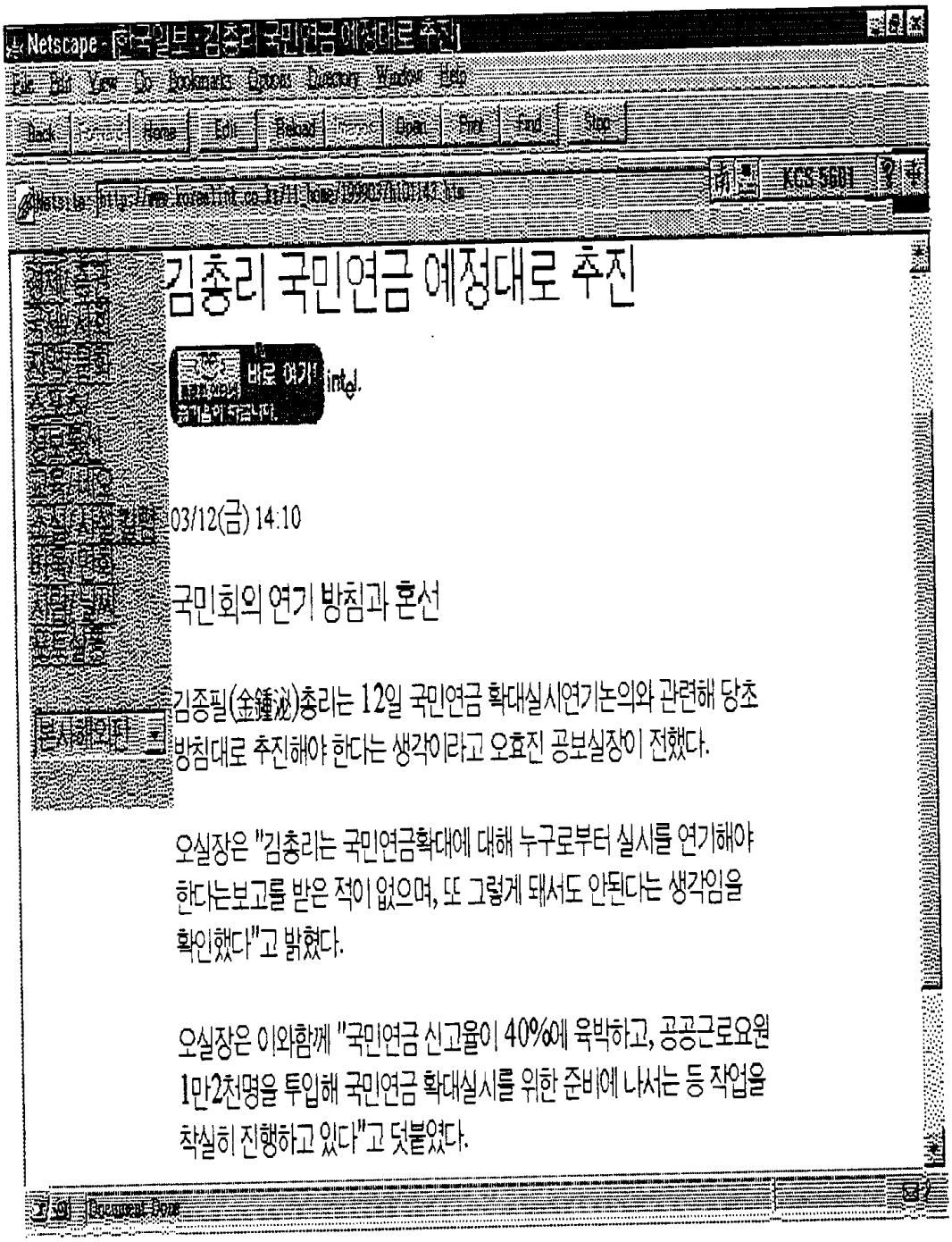
Comparison of Three Metadata Related Standards

在本文中，我們介紹了三個跟 metadata 相關的標準，它們分別是 FGDC 的 Digital Geospatial Metadata、Dublin Core、和 URC。雖然它們各有自己的設計目標和特質，但都是假設其操作環境爲類似網際網路的環境。FGDC 的 Digital Geospatial Metadata 是設計來專門處理地理性資料，由於它有聯邦行政命令的支持，可說是已成爲美國在地理方面的資料著錄國家標準。Dublin Core 則比較像是 USMARC 的網路節縮版，使非專業人士也能在短時間內熟悉和使用此格式來著錄收藏資料，但在現階段祇針對類似傳統印刷品的電子文件。由 IETE 的 URI 工作小組所負責的 URC，其原始的設計目的雖是用來連結 URL 和 URN，但爲因應電子圖書館時代的要求，其內含逐漸擴大，雖然尚在發展中，但由於有 IETE 的支持，未來成爲網際網路上通用標準的可能性極大。在此文中，我們也從幾個不同角度，分析和比較這三個 metadata 格式的異同和優缺點。

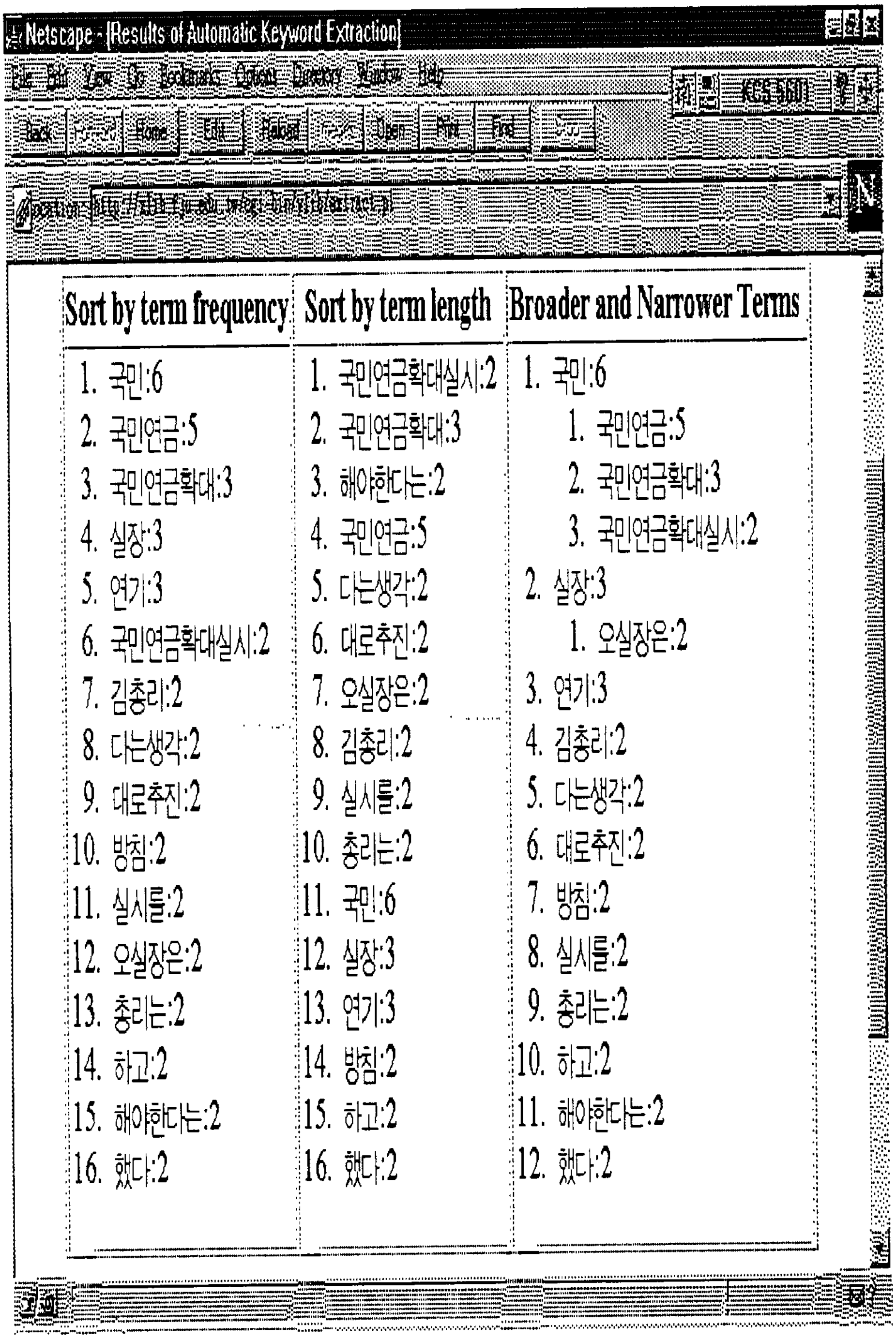
도면 6b

Terms Before Filtering	Terms After Filtering
1. 設計 : 3 (design)	1. 設計 : 3 (design)
2. 資料 : 3 (data)	2. 資料 : 3 (data)
3. 網路 : 3 (network)	3. 網路 : 3 (network)
4. 標準 : 3 (standard)	4. 標準 : 3 (standard)
5. Dublin Core : 2	5. Dublin Core : 2
6. FGDC 的 Digital Geospatial Metadata : 2	6. FGDC 的 Digital Geospatial Metadata : 2
7. IETE的 : 2 (of IETE)	7. IETE : 2
8. 三個 : 2 (three)	8. 三個 : 2 (three)
9. 文中 : 2 (in the article)	9. 文中 : 2 (in the article)
10. 比較 : 2 (comparison)	10. 比較 : 2 (comparison)
11. 它們 : 2 (they)	11. 它們 : 2 (they)
12. 由於 : 2 (owing to)	12. 由於 : 2 (owing to)
13. 地理 : 2 (geography)	13. 地理 : 2 (geography)
14. 成爲 : 2 (become)	14. 成爲 : 2 (become)
15. 我們 : 2 (we)	15. 我們 : 2 (we)
16. 的支持 : 2 (support of)	16. 支持 : 2 (support)
17. 的設計目 : 2 (*incorrect term*)	17. 設計目 : 2 (*incorrect term*)
18. 格式 : 2 (format)	18. 格式 : 2 (format)
19. 著錄 : 2 (record)	19. 著錄 : 2 (record)
20. 電子 : 2 (electronics)	20. 電子 : 2 (electronics)
21. 網際網路 : 2 (Internet)	21. 網際網路 : 2 (Internet)
22. 環境 : 2 (environment)	22. 環境 : 2 (environment)
23. 雖然 : 2 (although)	23. 雖然 : 2 (although)
24. 類似 : 2 (similar)	24. 類似 : 2 (similar)

도면 7a


Netscape - [한국일보:김총리 국민연금 예정대로 추진]
김총리 국민연금 예정대로 추진
바로 여기!
intel
03/12(금) 14:10
국민회의 연기 방침과 혼선
김종필(金鍾泌)총리는 12일 국민연금 확대실시연기논의와 관련해 당초 방침대로 추진해야 한다는 생각이라고 오효진 공보실장이 전했다.
오실장은 "김총리는 국민연금확대에 대해 누구로부터 실시를 연기해야 한다는보고를 받은 적이 없으며, 또 그렇게 돼서도 안된다는 생각임을 확인했다"고 밝혔다.
오실장은 이와함께 "국민연금 신고율이 40%에 육박하고, 공공근로요원 1만2천명을 투입해 국민연금 확대실시를 위한 준비에 나서는 등 작업을 착실히 진행하고 있다"고 덧붙였다.

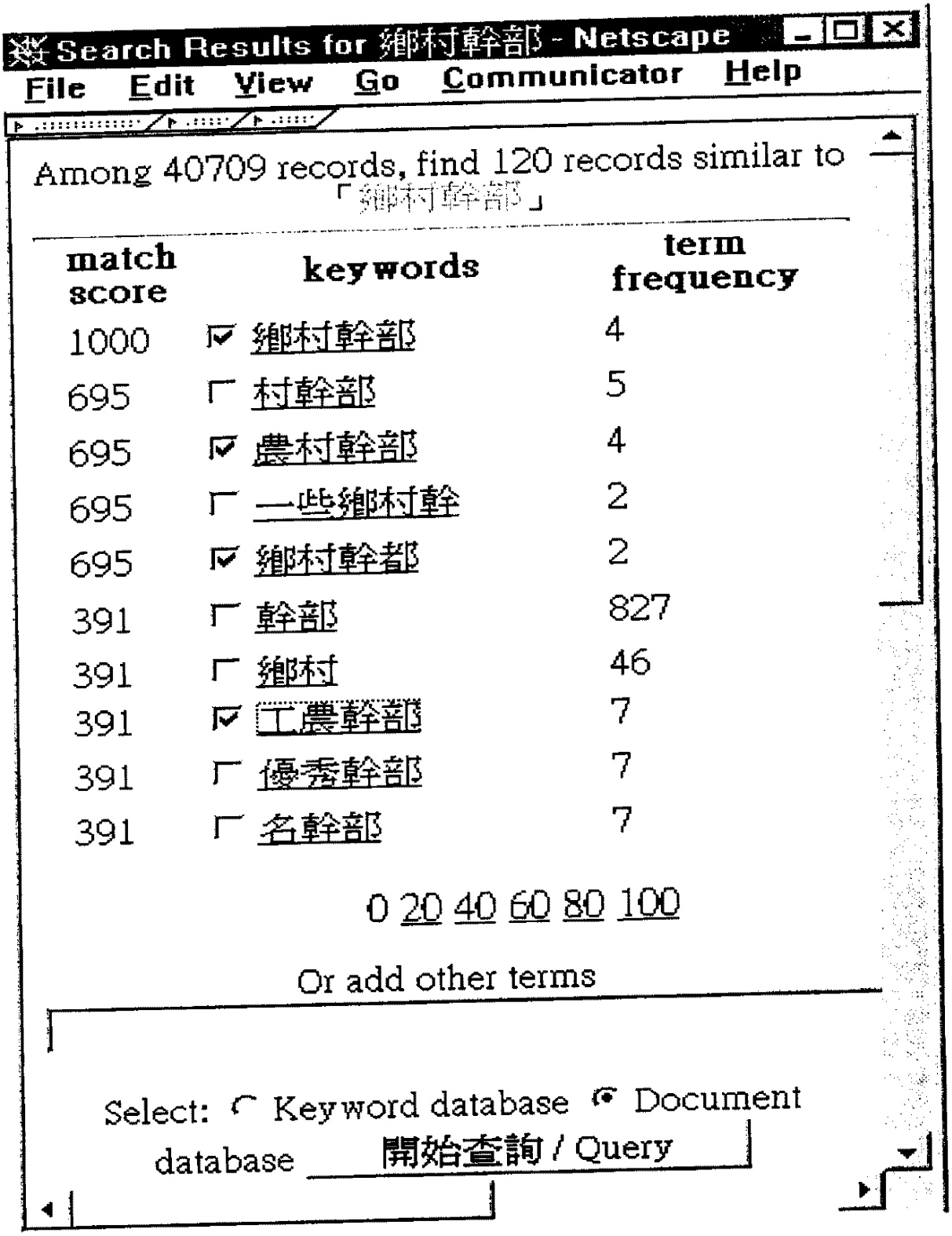
도면 7b


Netscape - [Results of Automatic Keyword Extraction]
Sort by term frequency
1. 국민:6
2. 국민연금:5
3. 국민연금확대:3
4. 실장:3
5. 연기:3
6. 국민연금확대실시:2
7. 김총리:2
8. 다는생각:2
9. 대로추진:2
10. 방침:2
11. 실시를:2
12. 오실장은:2
13. 총리는:2
14. 하고:2
15. 해야한다는:2
16. 했다:2
Sort by term length
1. 국민연금확대실시:2
2. 국민연금확대:3
3. 해야한다는:2
4. 국민연금:5
5. 다는생각:2
6. 대로추진:2
7. 오실장은:2
8. 김총리:2
9. 실시를:2
10. 총리는:2
11. 국민:6
12. 실장:3
13. 연기:3
14. 방침:2
15. 하고:2
16. 했다:2
Broader and Narrower Terms
1. 국민:6
1. 국민연금:5
2. 국민연금확대:3
3. 국민연금확대실시:2
2. 실장:3
1. 오실장은:2
3. 연기:3
4. 김총리:2
5. 다는생각:2
6. 대로추진:2
7. 방침:2
8. 실시를:2
9. 총리는:2
10. 하고:2
11. 해야한다는:2
12. 했다:2

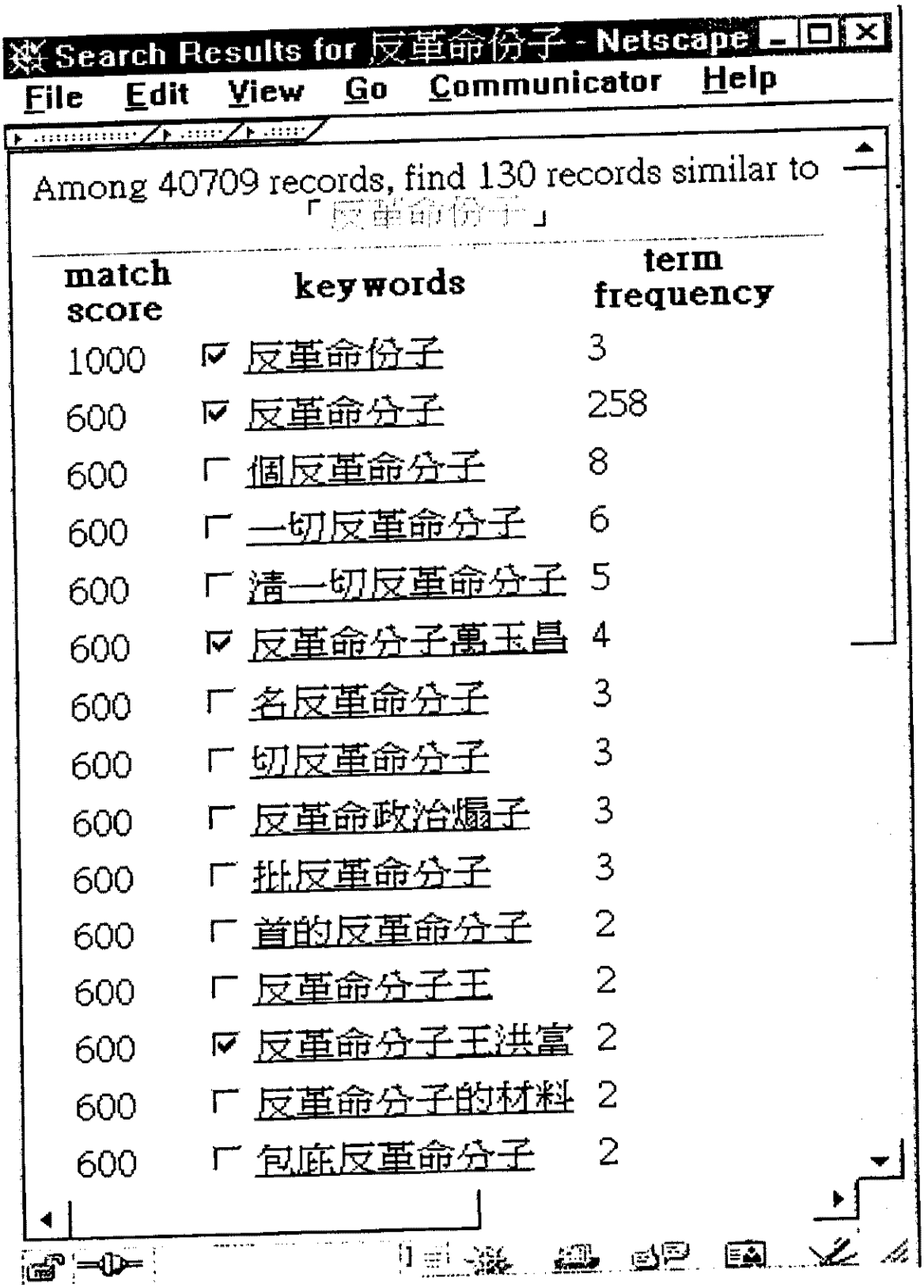
도면 7c

President Kin says:National Annuity is Encouraged to be Executed	
1. National Annuity is Enlarged to be Executed	9. Execution
2. National Annuity is Enlarged	10. President
3. Must be Done	11. National
4. National Annuity	12. Action Officer
5. Thoughts of	13. Delayed
6. Increase	14. Guiding Policy
7. Action Officer Wu	15. Do
8. President Kin	16. Complete

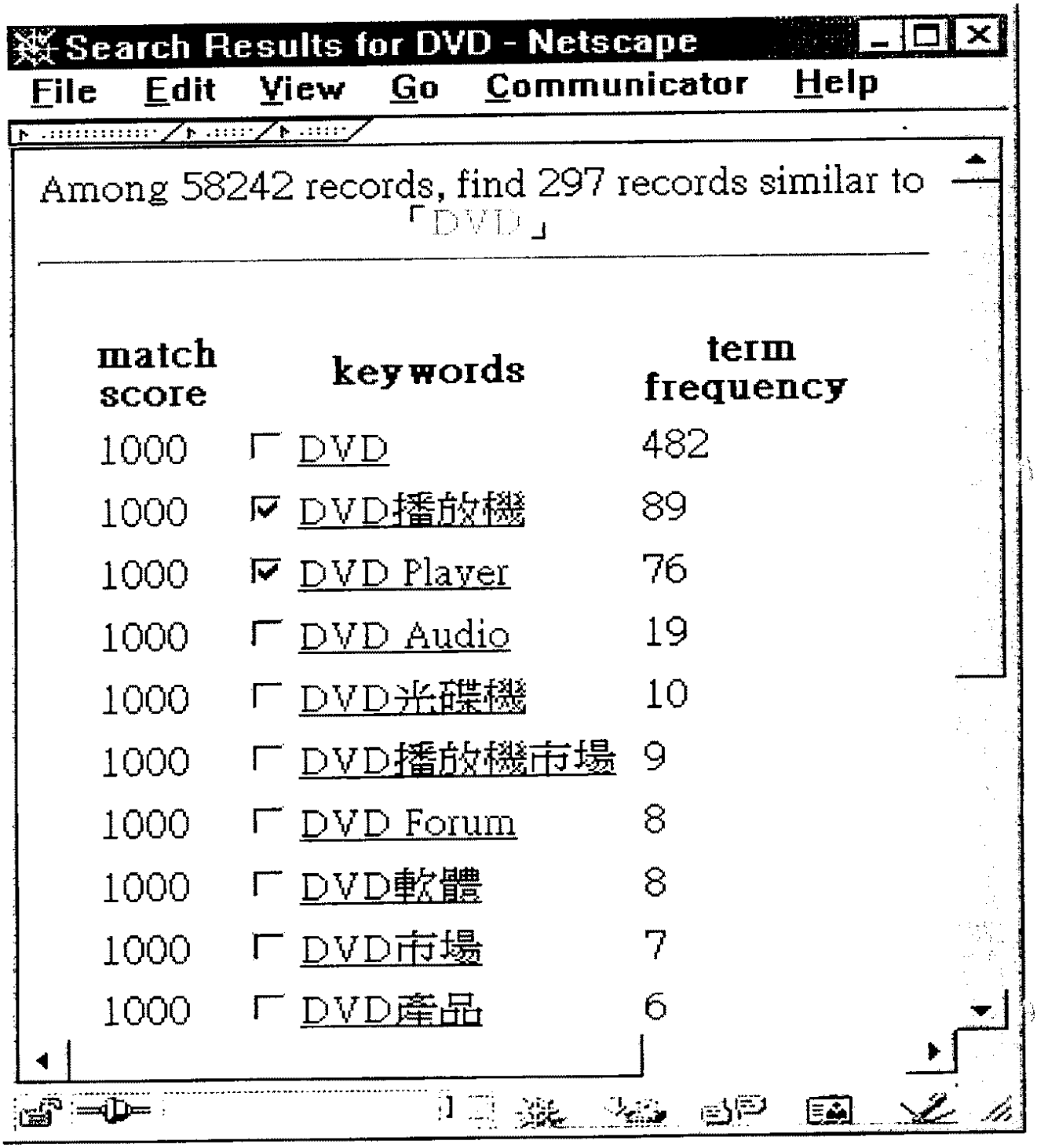
도면 8


Search Results for 鄉村幹部 - Netscape
File Edit View Go Communicator Help
Among 40709 records, find 120 records similar to
「鄉村幹部」
match score
keywords
term frequency
1000 鄉村幹部 4
695 村幹部 5
695 農村幹部 4
695 一些鄉村幹 2
695 鄉村幹都 2
391 幹部 827
391 鄉村 46
391 工農幹部 7
391 優秀幹部 7
391 名幹部 7
0 20 40 60 80 100
Or add other terms
Select: Keyword database Document database
開始查詢 / Query

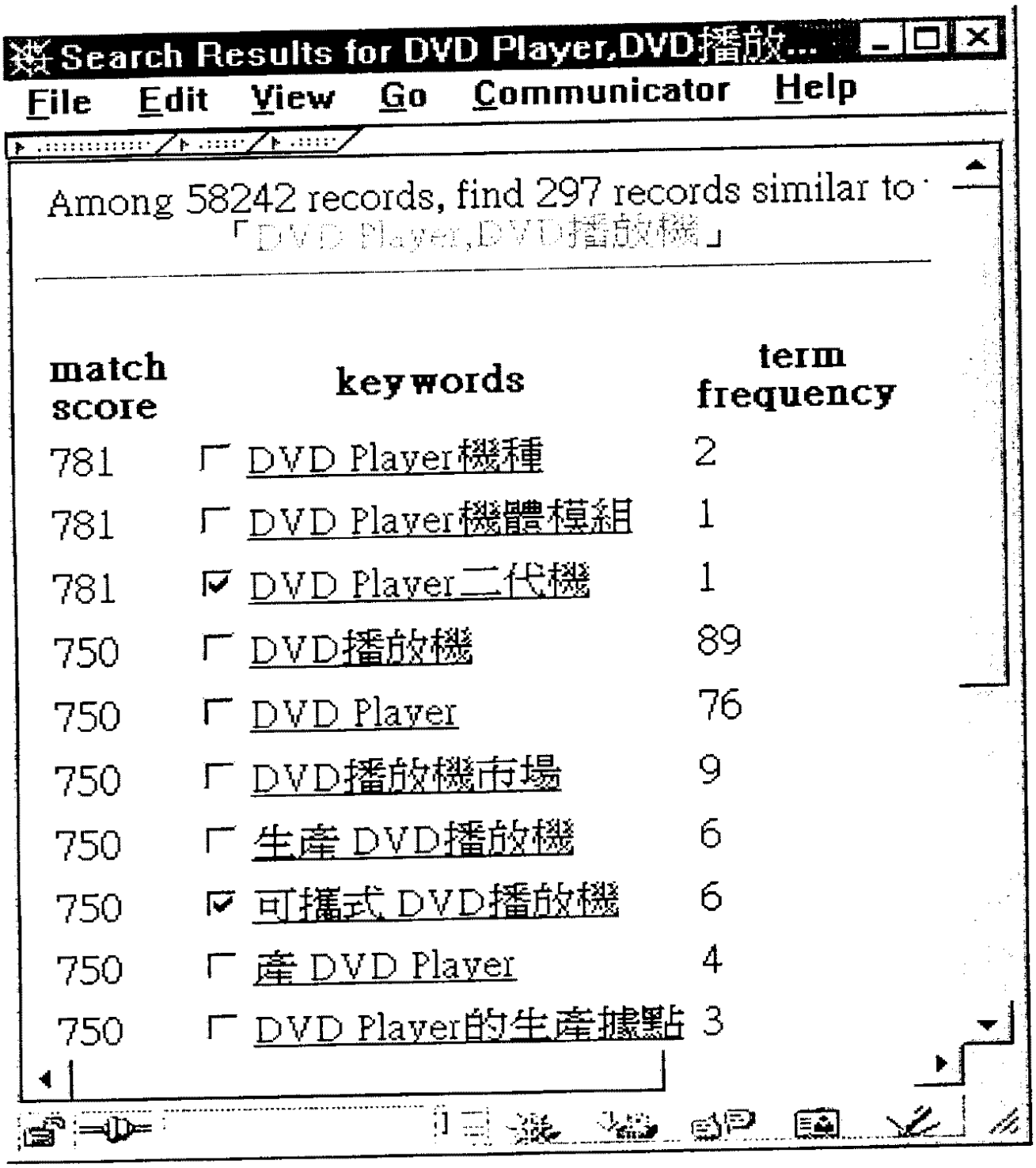
도면 9


Search Results for 反革命份子 - Netscape
File Edit View Go Communicator Help
Among 40709 records, find 130 records similar to
「反革命份子」
match score
keywords
term frequency
1000 反革命份子 3
600 反革命分子 258
600 個反革命分子 8
600 一切反革命分子 6
600 清一切反革命分子 5
600 反革命分子萬玉昌 4
600 名反革命分子 3
600 切反革命分子 3
600 反革命政治煽子 3
600 批反革命分子 3
600 首的反革命分子 2
600 反革命分子王 2
600 反革命分子王洪富 2
600 反革命分子的材料 2
600 包庇反革命分子 2

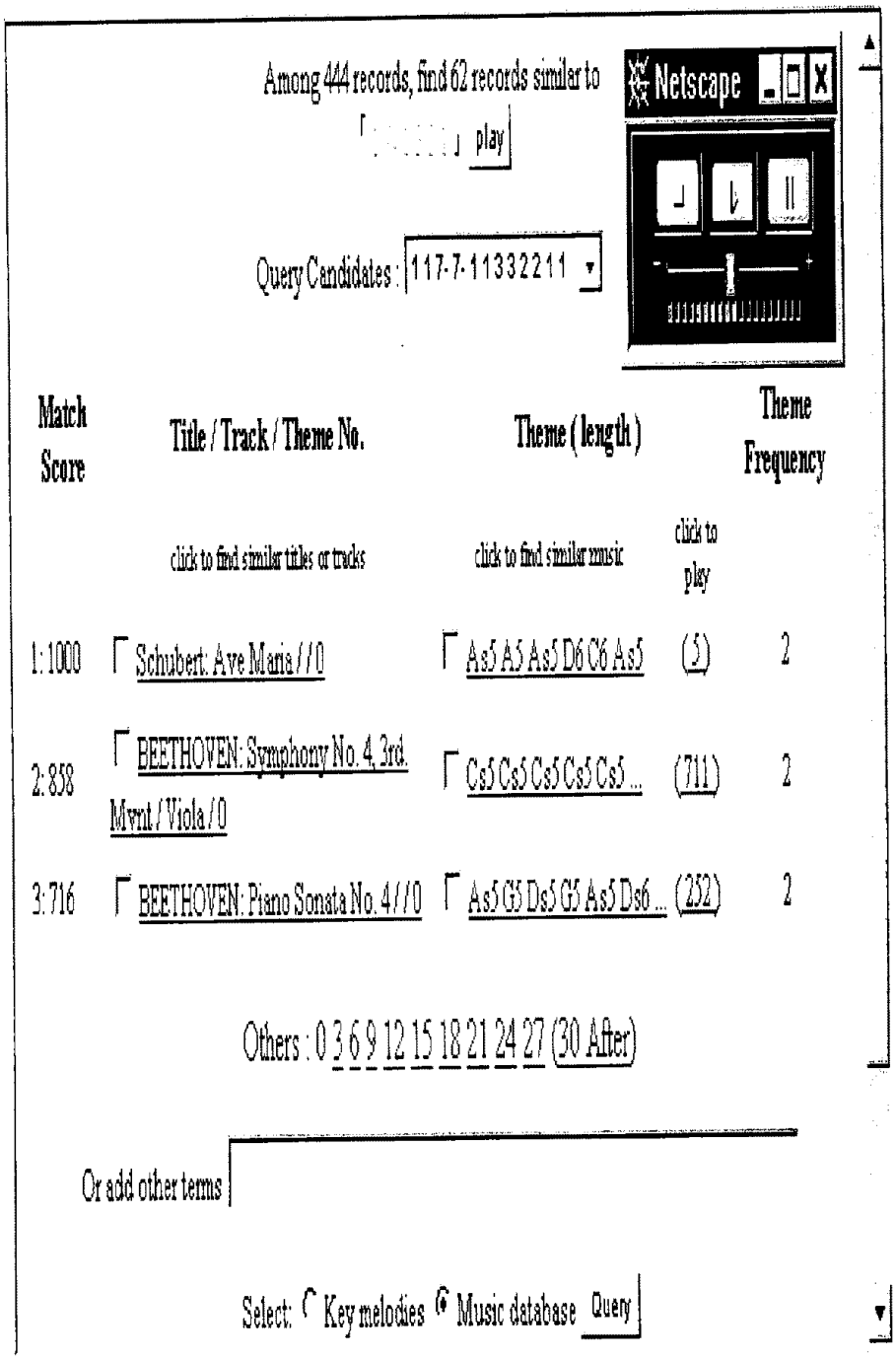
도면 10a


Search Results for DVD - Netscape
File Edit View Go Communicator Help
Among 58242 records, find 297 records similar to 「DVD」
match score
keywords
term frequency
1000 DVD 482
1000 DVD播放機 89
1000 DVD Player 76
1000 DVD Audio 19
1000 DVD光碟機 10
1000 DVD播放機市場 9
1000 DVD Forum 8
1000 DVD軟體 8
1000 DVD市場 7
1000 DVD產品 6

도면 10b


Search Results for DVD Player,DVD播放...
File Edit View Go Communicator Help
Among 58242 records, find 297 records similar to
「DVD Player,DVD播放機」
match score
keywords
term frequency
781 DVD Player機種 2
781 DVD Player機體模組 1
781 DVD Player二代機 1
750 DVD播放機 89
750 DVD Player 76
750 DVD播放機市場 9
750 生產 DVD播放機 6
750 可攜式 DVD播放機 6
750 產 DVD Player 4
750 DVD Player的生產據點 3

도면 11


Among 444 records, find 62 records similar to
play
Query Candidates : 117-7-11332211
Netscape
Match Score
Title / Track / Theme No.
Theme (length)
Theme Frequency
click to find similar titles or tracks
click to find similar music
click to play
1: 1000
Schubert: Ave Maria / / 0
As5 A5 As5 D6 C6 As5
(5)
2
2: 858
BEETHOVEN: Symphony No. 4, 3rd. Mvmt / Viola / 0
Cs5 Cs5 Cs5 Cs5 Cs5 ...
(711)
2
3: 716
BEETHOVEN: Piano Sonata No. 4 / / 0
As5 G5 Ds5 G5 As5 Ds6 ...
(252)
2
Others : 0 3 6 9 12 15 18 21 24 27 (30 After)
Or add other terms
Select: Key melodies Music database Query