(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0077423
(43) 공개일자 2014년06월24일

(51) 국제특허분류(Int. Cl.)
G10L 15/14 (2006.01)
(21) 출원번호 10-2012-0146228
(22) 출원일자 2012년12월14일
심사청구일자 없음

(71) 출원인
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
(72) 발명자
정의석
대전 유성구 은구비남로 55, 705동 1301호 (지족동, 열매마을7단지)
박전규
대전 유성구 대덕대로541번길 68, 103동 205호 (도룡동, 현대아파트)
(뒷면에 계속)
(74) 대리인
특허법인이지

전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **음성인식기의 언어모델 저장방법**

(57) 요 약

실시 예에 따른 음성인식기의 언어모델 저장방법은, 발화된 음성 신호에 대응된 압축 언어모델을 생성 및 저장하기 용이하도록, 실시 예는, 발화된 음성 신호에 대응하는 텍스트 말뭉치(text corpus)를 생성하는 단계, 상기 텍스트 말뭉치에 기초된 공동체 표준 ARPA 포맷의 ARPA 언어모델을 생성하는 단계, 상기 ARPA 언어모델을 어휘 언어모델 및 ngram 언어 모델로 분리하는 단계 및 상기 어휘 언어모델 및 상기 ngram 언어 모델을 설정된 언어모델 스코어 연산 알고리즘에 따라 인코딩하여 압축 언어모델을 생성 및 저장하는 단계를 포함하는 음성인식기의 언어모델 저장방법을 제공한다.

대 표 도 - 도2

```
score_type { prob, backoff, inx[MAX_INX] }

score_type  mst[MAX_NGRAM][MAX_TOK_NUM];

function compute_lm_score( input_str[], lm_score )
    for (int i = 0; i < MAX_NGRAM; i++) {
        for (int j = 0; j < num of input_str; j++) {
            if (i == 0) {
                if (LM_DB.find( input_str[j] ))
                    mst[i][j] <- result of LM_DB search
            } else {
                if (j < i)  mst[i][j] <- mst[i-1][j]
                else {
                    if (LM_DB.find("mst[i-1][j-1].inx mst[0][j].inx"))
                        mst[i][j] <- result of LM_DB search
                    else if (mst[i-1][j-1].inx exists in LM_DB)
                        mst[i][j].prob <- mst[i-1][j].prob + mst[i-1][j-1].backoff
                    else
                        mst[i][j].prob <- mst[i-1][j].prob
                }
            }
        }

        if (LM_DB no result) { lm_ngram <- i; break; }
    }

    lm_score <- 0
    for (int k = 0; k < num of input_str; k++)
        lm_score += mst[lm_ngram][k].prob;
```

(72) 발명자

전형배

대전 서구 청사로 65, 109동 605호 (월평동, 황실타운)

이윤근

대전 서구 청사서로 11, 103동 1406호 (월평동, 무지개아파트)

이 발명을 지원한 국가연구개발사업

과제고유번호	11921-03001
부처명	KCC
연구사업명	방송통신연구개발사업[기술개발부문]
연구과제명	Beyond 스마트TV 기술 개발
기 여 율	1/1
주관기관	ETRI
연구기간	2011.03.01 ~ 2015.02.28

특허청구의 범위

청구항 1

발화된 음성 신호에 대응하는 텍스트 말뭉치(text corpus)를 생성하는 단계;

상기 텍스트 말뭉치에 기초된 공동체 표준 ARPA 포맷의 ARPA 언어모델을 생성하는 단계;

상기 ARPA 언어모델을 어휘 언어모델 및 ngram 언어 모델로 분리하는 단계; 및

상기 어휘 언어모델 및 상기 ngram 언어 모델을 설정된 언어모델 스코어 연산 알고리즘에 따라 인코딩된 압축 언어모델을 생성 및 저장하는 단계;를 포함하는 음성인식기의 언어모델 저장방법.

명 세 서

기 술 분 야

[0001] 실시 예는 음성인식기의 언어모델 저장방법에 관한 것으로서, 더욱 상세하게는 발화된 음성 신호에 대응된 압축 언어모델을 생성 및 저장하기 용이한 음성인식기의 언어모델 저장방법에 관한 것이다.

배 경 기 술

[0002] 음성인식은 음성에 포함된 언어적인 정보를 추출하여 인간이 해독할 수 있는 표현방법으로 변환하는 과정을 말하는 것으로서, 음향학, 음운학, 언어학 등의 단계적인 처리를 필요로 한다. 이를 위한 음향모델은 입력음성과의 음향학적 조합을 통해 음향학적 우도(Likelihood)로 주어진 모델을 말하고, 언어모델은 이웃하는 단어 사이의 연관성을 나타내는 정보를 포함하는 것으로서, 유한상태 네크워크(finite-state network) 언어 모델, 문맥의존(context-sensitive grammar) 언어모델 등이 있다. 음성인식은 연속음성인식(continuous speech recognition)과 단어음성인식(word recognition)으로 나뉠 수 있다.

[0003] 통상적인 음성인식시스템은 음향분석부, 음운인식부, 단어인식부, 언어처리부 등의 요소로 구성되어 있다. 음향분석부에서는 음성신호에 대해 20~30ms의 짧은 구간마다 주파수분석 또는 선형예측분석이라 부르는 수학적인 변환처리를 하고, 이것으로 십수차원의 특징벡터(feature vector)계열로 변환한다. 음운인식부에서는 음성의 대략적인 특징을 이용하여 음성신호를 일정한 물리적 성질을 지닌 부분으로 분할하는 조작, 즉 세그먼테이션(segmentation)을 하고, 각 구간을 각각 하나의 단위로 하여 모음과 자음을 인식한다. 음운인식의 결과를 1차원적인 음운기호열로 나타내기는 곤란하므로, 몇몇 가능성을 남긴 음운래티스(phoneme lattice) 꼴로 주어진다. 단어인식부에서는 단어의 음형이 기술되어 있는 단어사전을 참조하여 음운래티스를 단어래티스(word lattice)로 변환한다. 언어처리부에서는 단어래티스 안의 단어들로부터 구문적 제약을 만족시키고 의미적으로도 정합이 이루어진 단어열을 선택한다.

[0004] 음성인식을 위해서는 많은 다양한 기술들이 사용된다. 전형적인 음성인식은 음성의 디지털 샘플링(digital sampling)에서 시작되어, 음향 신호처리에 의한 분석이 수행된다. 이러한 방법에는 LPC analysis (Linear Predictive Coding): 선형예측코딩, MFCC(Mel Frequency Cepstral Coefficients), cochlea modeling 등이다.

[0005] 다음은 음소의 인식이다(recognition of phonemes). 음소들의 그룹과 단어를 포함하고, 이에 사용되는 기술들은 DTW(Dynamic Time Warping), HMM(hidden Markov modeling), Neural Networks, expert systems and combinations of technique 등이 있다. 최근까지 음성 인식에 가장 많이 사용되며 성공적이었던 알고리즘은 HMM이며, HMM은 이중 통계적 모델로서, 기본이 되는 음소열의 생성과 프레임 단위의 표면적 음향학적인 표현을 markov 과정과 같이 확률로서 나타낸다.

[0006] 일반적인 음성인식 분야의 언어모델 저장 방법은 일반적 ASR(자동음성인식)의 언어모델은 음향모델과 통합하여 wFST(weighted finite state transducer) 기반의 정보구조로 저장되나, 후처리 방식의 ASR 리스코링에서는 방대한 양의 언어모델 활용이 필요하다. 현재 wFST기술은 방대한 양의 언어모델을 통합하기에는 한계가 있다.

발명의 내용

해결하려는 과제

[0007] 실시 예의 목적은, 발화된 음성 신호에 대응된 압축 언어모델을 생성 및 저장하기 용이한 음성인식기의 언어모델 저장방법을 제공함에 있다.

과제의 해결 수단

[0008] 실시 예에 따른 음성인식기의 언어모델 저장방법은, 발화된 음성 신호에 대응하는 텍스트 말뭉치(text corpus)를 생성하는 단계, 상기 텍스트 말뭉치에 기초된 공동체 표준 ARPA 포맷의 ARPA 언어모델을 생성하는 단계, 상기 ARPA 언어모델을 어휘 언어모델 및 ngram 언어 모델로 분리하는 단계 및 상기 어휘 언어모델 및 상기 ngram 언어 모델을 설정된 언어모델 스코어 연산 알고리즘에 따라 인코딩하여 압축 언어모델을 생성 및 저장하는 단계를 포함한다.

[0009] 또한, 실시 예에 따른 음성인식의 언어모델 저장방법은, 상기 텍스트 말뭉치를 이진 배열 구조로 변환된 언어모델 동적 차트를 생성하고, 상기 언어모델 동적 차트에 상기 어휘 언어모델 및 상기 ngram 언어 모델을 결합하여 인코딩하는 언어모델 스코어 연산 알고리즘이 설정된다.

발명의 효과

[0010] 실시 예에 따른 음성인식기의 언어모델 저장방법은, 대용량의 ARPA 언어모델을 고압축된 압축 언어모델로 생성 및 저장함으로써, 저장 공간을 효율적으로 이용할 수 있는 이점이 있다.

[0011] 또한, 실시 예에 따른 음성인식기의 언어모델 저장방법은, 대용량의 ARPA 언어모델을 어휘 언어모델 및 ngram 언어 모델로 분리한 후 재 구성된 압축 언어모델을 생성 및 저장함으로써, 활용도 측면이 향상되는 이점이 있다.

도면의 간단한 설명

[0012] 도 1은 실시 예에 따른 음성인식기의 언어모델 저장방법을 나타낸 순서도이다.

도 2는 실시 예에 따른 언어모델 동적 차트를 생성하기 위한 알고리즘을 나타낸 예시도이다.

발명을 실시하기 위한 구체적인 내용

[0013] 이하의 내용은 단지 발명의 원리를 예시한다. 그러므로 당업자는 비록 본 명세서에 명확히 설명되거나 도시되지 않았지만 발명의 원리를 구현하고 발명의 개념과 범위에 포함된 다양한 장치를 발명할 수 있는 것이다. 또한, 본 명세서에 열거된 모든 조건부 용어 및 실시예들은 원칙적으로, 발명의 개념이 이해되도록 하기 위한 목적으로만 명백히 의도되고, 이와 같이 특별히 열거된 실시예들 및 상태들에 제한적이지 않는 것으로 이해되어야 한다.

[0014] 또한, 발명의 원리, 관점 및 실시예들 뿐만 아니라 특정 실시예를 열거하는 모든 상세한 설명은 이러한 사항의 구조적 및 기능적 균등물을 포함하도록 의도되는 것으로 이해되어야 한다. 또한 이러한 균등물들은 현재 공지된 균등물뿐만 아니라 장래에 개발될 균등물 즉 구조와 무관하게 동일한 기능을 수행하도록 발명된 모든 소자를 포함하는 것으로 이해되어야 한다.

[0015] 따라서, 예를 들어, 본 명세서의 블럭도는 발명의 원리를 구체화하는 예시적인 개념적 관점을 나타내는 것으로 이해되어야 한다. 이와 유사하게, 모든 흐름도, 상태 변환도, 의사 코드 등은 컴퓨터가 판독 가능한 매체에 실질적으로 나타낼 수 있고 컴퓨터 또는 프로세서가 명백히 도시되었는지 여부를 불문하고 컴퓨터 또는 프로세서에 의해 수행되는 다양한 프로세스를 나타내는 것으로 이해되어야 한다.

[0016] 프로세서 또는 이와 유사한 개념으로 표시된 기능 블럭을 포함하는 도면에 도시된 다양한 소자의 기능은 전용 하드웨어뿐만 아니라 적절한 소프트웨어와 관련하여 소프트웨어를 실행할 능력을 가진 하드웨어의 사용으로 제공될 수 있다. 프로세서에 의해 제공될 때, 상기 기능은 단일 전용 프로세서, 단일 공유 프로세서 또는 복수의 개별적 프로세서에 의해 제공될 수 있고, 이들 중 일부는 공유될 수 있다.

[0017] 또한 프로세서, 제어 또는 이와 유사한 개념으로 제시되는 용어의 명확한 사용은 소프트웨어를 실행할 능력을 가진 하드웨어를 배타적으로 인용하여 해석되어서는 아니되고, 제한 없이 디지털 신호 프로세서(DSP) 하드웨어, 소프트웨어를 저장하기 위한 롬(ROM), 램(RAM) 및 비 휘발성 메모리를 암시적으로 포함하는 것으로 이해되어야 한다. 주지관용의 다른 하드웨어도 포함될 수 있다.

[0018] 본 명세서의 특허청구범위에서, 상세한 설명에 기재된 기능을 수행하기 위한 수단으로 표현된 구성요소는 예를 들어 상기 기능을 수행하는 회로 소자의 조합 또는 펌웨어/마이크로 코드 등을 포함하는 모든 형식의 소프트웨어를 포함하는 기능을 수행하는 모든 방법을 포함하는 것으로 의도되었으며, 상기 기능을 수행하도록 상기 소프트웨어를 실행하기 위한 적절한 회로와 결합된다. 이러한 특허청구범위에 의해 정의되는 발명은 다양하게 열거된 수단에 의해 제공되는 기능들이 결합되고 청구항이 요구하는 방식과 결합되기 때문에 상기 기능을 제공할 수 있는 어떠한 수단도 본 명세서로부터 파악되는 것과 균등한 것으로 이해되어야 한다.

[0019] 상술한 목적, 특징 및 장점은 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 또한, 발명을 설명함에 있어서 발명과 관련된 공지 기술에 대한 구체적인 설명이 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에 그 상세한 설명을 생략하기로 한다.

[0020] 이하, 첨부된 도면을 참조하여 발명에 따른 실시 예를 상세히 설명하기로 한다.

[0021] 도 1은 실시 예에 따른 음성인식기의 언어모델 저장방법을 나타낸 순서도이다.

[0022] 도 1을 참조하면, 음성인식기의 언어모델 저장방법은, 발화된 음성 신호에 대응하는 텍스트 말뭉치(text corpus)를 생성하고(S100), 상기 텍스트 말뭉치에 기초된 공동체 표준 ARPA 포맷의 ARPA 언어모델을 생성하며(S110), 상기 ARPA 언어모델을 어휘 언어모델 및 ngram 언어 모델로 분리하고(S120) 및 상기 어휘 언어모델 및 상기 ngram 언어 모델을 설정된 언어모델 스코어 연산 알고리즘에 따라 인코딩된 압축 언어모델을 생성 및 저장한다(S130).

[0023] 여기서, 음성인식기는 사용자로부터 발화된 음성 신호 입력시, 상기 음성 신호에 포함된 잡음(noise) 등과 같은 신호를 제거 및 상기 음성 신호에 대한 음향모델에 따른 연산과 언어모델에 따른 연산 과정을 실행한다.

[0024] 이때, 상기 음성인식기는 상기 언어모델에 따른 연산 과정 시, 상기 음성 신호에 대응하는 텍스트 말뭉치를 생성한다.

[0025] 상기 텍스트 말뭉치는 언제든지 재사용이 가능하도록 부가적인 정보화 다 큐먼트가 갖추어져 있으며, 컴퓨터로 읽을 수 있는 형태로 구성된 음성자료의 모음을 말한다.

[0026] 그리고, 상기 음성인식기는 상기 텍스트 말뭉치를 기초로 설정된 공동체 표준 ARPA 포맷의 ARPA 언어모델을 생성한다.

[0027] 예를 들어, 상기 ARPA 언어모델은 하기의 [예 1]과 같은 ARPA 언어모델 형식에 의해 생성된다.

[0028] [예 1]

[0029] \data\

[0030] ngram 1=4989

[0031] ngram 2=835668

[0032] ngram 3=12345678

[0033] \1-grams:

[0034] ...

[0035] -0.9792 ABC -2.2031

[0036] ...

[0037] \2-grams:

[0038] ...

[0039] -0.8328 ABC DEFG -3.1234

[0040] ...

[0041] \3-grams:

[0042] ...

[0043] -0.234 ABCD EFGHI JKL

[0044] ...

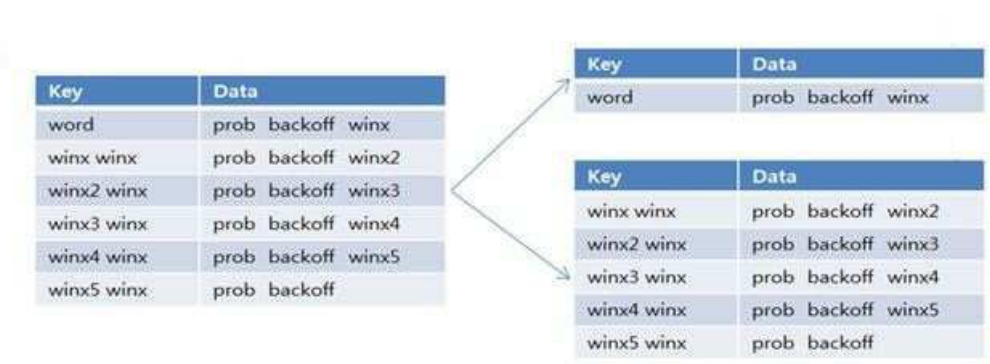
[0045] \end\

[0046] 상술한 [예 1]은 3gramdp 대한 기술을 나타낸다.

[0047] 즉, 1gram은 4,989개, 2gram은 835,669개, 3gram은 12,345,678개가 존재하면 각 해당 ngram 섹션에 해당 정보가 라인당 존재하게 된다.

[0048] 다시 말하면, "1-grams:"의 "-0.9792 ABC -2.2031"는 정보가 4,989개가 존재하는 것이다. 여기서 -0.9792는 probability(prob) 이고, -2.2031은 backoff값이 된다.

[0049] [예 2]



[0050]

[0051] [예 2]의 테이블은 상기 ARPA 언어모델을 KEY, 어휘 언어모델 및 DATA, nga형태로 표현한 것이다. 1gram의 경우 "word"가 KEY가 되고, "prob backoff winx"가 DATA가 된다. 여기서 winx는 해당 엔트리의 정수 인덱스 값이 된다.

[0052] 2gram의 경우 1gram에 존재하는 어떤 winx와 어떤 winx의 결합으로 KEY가 구성되며 "prob backoff winx2"가 DATA가 된다. 여기서 winx2는 2gram의 인덱스가 된다. ngram에서 n값이 증가할 때마다 일정한 두개의 winx인덱스로만 표현이 가능하다.

[0053] 여기서, 상기 음성인식기는 어휘 테이블과 2gram 이상으로 구성된 ngram 언어모델을 분리 및 재구성된 압축 언어모델을 생성 및 저장하게 된다.

[0054] 이때, 상기 음성인식기는 상기 압축 언어모델을 압축하기 위하여 winx 정수를 이진 배열 구조를 갖도록 63진수로 변환한 후 이를 문자열로 변환한다.

[0055] 여기서, 상기 음성인식기는 최종적으로 생성된 인덱스용 문자열이 실제 언어모델 데이터 베이스에 저장하여 정수의 winx를 대체하여 스트링 형태로 표현되게 된다. 이는 일종의 큰 스트링으로 표현되는 10진수 정수를 63진수로 변환하여 작은 스트링으로 변환하여 ARPA 언어모델을 압축 및 축소하여 상기 압축 언어모델을 생성한다.

[0056] 즉, 상기 음성인식기는 상기 ARPA 언어모델로부터 분리된 상기 어휘 언어모델 및 상기 ngram 언어모델을 설정된 언어모델 스코어 연산 알고리즘에 따라 인코딩된 압축 언어모델을 생성 및 저장한다.

[0057] 상술한 상기 어휘 모델은 상기 텍스트 말뭉치에 포함된 단어의 어휘에 대한 언어모델이며, 상기 ngram 언어모델은 과거의 N-1개의 단어로부터 다음에 나타날 단어의 확률을 정의하는 문법이 되는 것으로서, 충분한 학습 데이터가 존재할 경우 매우 좋은 성능을 보이고 있다. 그러나, 인식 어휘가 점점 증가하게 되면, 통계적 언어 모델

은 대용량의 훈련 데이터가 필요하게 되며, 시간 복잡도 및 공간 복잡도의 영향으로 형태소나 어절의 바이그램(bigram) 또는 트라이그램(trigram) 정도의 간단한 언어 모델만이 적용 가능하다. 따라서, 통계적 언어모델의 개선이 요구되고 있는 실정이다. 또한, 상기 언어모델은 대어휘 연속 음성 인식을 위한 기술이 요구되는 분야로, 단어 단위로 인식된 결과를 문장으로 재구성하는 작업에 사용되며, 음향학적인 모호함 때문에 정확히 인식하지 못하는 부분을 언어 정보를 이용하여 탐색공간을 줄이는 역할을 하는 것으로, 이러한 대어휘 연속 음성인식 성능 향상은 주로 언어 모델의 성능 향상을 통하여 구현되고 있으므로, 음성인식에 있어서 언어 모델의 개선은 매우 큰 비중을 차지한다.

[0058] 여기서, 상기 언어모델 스코어 연산 알고리즘은 상기 텍스트 말뭉치를 이진 배열 구조로 변환된 언어모델 동적 차트를 생성하고, 상기 언어모델 동적 차트에 상기 어휘 언어모델 및 상기 ngram 언어 모델을 결합하여 인코딩하여 상기 압축 언어모델을 생성하는 알고리즘이다.

[0059] 도 2는 실시 예에 따른 언어모델 동적 차트를 생성하기 위한 알고리즘을 나타낸 예시도이다.

[0060] 도 2을 참조하면, 알고리즘의 1행에서 score_type은 prob, backoff, inx를 갖고 있다. 이는 상술한 [예 2]의 DATA 정보와 일치한다. 3행은 mst로 명명된 이진 배열 구조를 갖고 이는 연산시 필요한 동적 차트의 선언부이다.

[0061] 5행은 함수 명칭을 보여 주고, 텍스트 말뭉치 input_str과 연산 결과를 돌려줄 lm_score를 선언한다.

[0062] 6행은 mst의 행 크기만큼 for문을 실행하며 이는 MAX_NGRAM(최대 ngram)까지 수행된다.

[0063] 7행은 입력 스트링의 어휘수까지 진행된다.

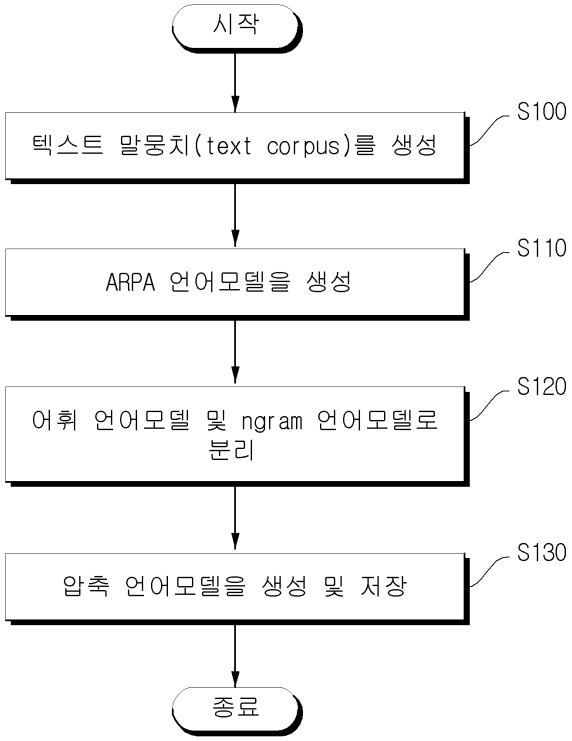
[0064] 행8은 1gram일 때, 즉 i가 0일 때, LM_DB로부터 입력 스트링의 정보를 찾아 mst[i][j]값을 채우는 기능을 수행한다. 여기서 LM_DB는 [예 2]의 어휘 언어모델에 해당한다. 찾아진 결과물은 해당 언어모델의 DATA 정보가 되고 mst슬롯에 할당된다.

[0065] 행12는 mst에서 ngram차수를 충족하지 못할 때 하위 mst[i-1][j]이 그대로 상위 mst[i][j]에 전이되는 것을 보여 준다. 해당 i, j에서 ngram이 존재할 수 있는 상태일 때 ngram 테이블을 찾게 된다. 여기서 키는 mst[i-1][j-1].inx와 mst[0][j].inx를 결합한 문자열이 된다. 이는 행 14에 표현되어 있고, 행 15는 LM_DB의 검색결과가 저장된다. 여기서 mst[i-1][j-1].inx만 DB에 존재할 때 언어 모델의 backoff연산이 진행되는데 이 과정은 행17에 기술되어 있다. 여기서 mst[i-1][j-1].inx의 존재여부 확인은 기존 연산과정에서만 DB 탐색이 발생한다. 즉, 동적 연산(dynamic programming)방식의 고효율적인 연산이라고 할 수 있다. 최종적으로 mst[i-1][j-1].inx 값도 존재하지 않을 때 이전 단계의 값이 그대로 전이된다. 이는 행19에 기술되어 있다. 전체과정을 거치면 최종적으로 lm_ngram값이 결정되고, 이는 DB의 존재 ngram수를 의미하며, lm_score값이 계산된다. 이는 행 28, 29에 기술되어 있다.

[0066] 이상에서 실시 예를 중심으로 설명하였으나 이는 단지 예시일 뿐 본 발명을 한정하는 것이 아니며, 본 발명이 속하는 분야의 통상의 지식을 가진 자라면 본 실시예의 본질적인 특성을 벗어나지 않는 범위에서 이상에 예시되지 않은 여러 가지의 변형과 응용이 가능함을 알 수 있을 것이다. 예를 들어, 실시 예에 구체적으로 나타난 각 구성 요소는 변형하여 실시할 수 있는 것이다. 그리고 이러한 변형과 응용에 관계된 차이점들은 첨부된 청구 범위에서 규정하는 본 발명의 범위에 포함되는 것으로 해석되어야 할 것이다.

도면

도면1



도면2

```
score_type { prob, backoff, inx[MAX_INX] }

score_type  mst[MAX_NGRAM][MAX_TOK_NUM];

function compute_lm_score( input_str[], lm_score )
    for (int i = 0; i < MAX_NGRAM; i++) {
        for (int j = 0; j < num of input_str; j++) {
            if (i == 0) {
                if (LM_DB.find( input_str[j] ))
                    mst[i][j] <- result of LM_DB search
            } else {
                if (j < i)  mst[i][j] <- mst[i-1][j]
                else {
                    if (LM_DB.find("mst[i-1][j-1].inx mst[0][j].inx"))
                        mst[i][j] <- result of LM_DB search
                    else if (mst[i-1][j-1].inx exists in LM_DB)
                        mst[i][j].prob <- mst[i-1][j].prob + mst[i-1][j-1].backoff
                    else
                        mst[i][j].prob <- mst[i-1][j].prob
                }
            }
        }

        if (LM_DB no result) { lm_ngram <- i; break; }
    }

    lm_score <- 0
    for (int k = 0; k < num of input_str; k++)
        lm_score += mst[lm_ngram][k].prob;
```