



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년11월24일
(11) 등록번호 10-1679445
(24) 등록일자 2016년11월18일

(51) 국제특허분류(Int. Cl.)
G10L 15/06 (2006.01) G10L 15/14 (2006.01)
(21) 출원번호 10-2011-7022845
(22) 출원일자(국제) 2010년03월26일
심사청구일자 2015년03월04일
(85) 번역문제출일자 2011년09월29일
(65) 공개번호 10-2012-0018114
(43) 공개일자 2012년02월29일
(86) 국제출원번호 PCT/US2010/028932
(87) 국제공개번호 WO 2010/117688
국제공개일자 2010년10월14일
(30) 우선권주장
12/413,606 2009년03월30일 미국(US)
(56) 선행기술조사문헌
JP10240288 A
JP2002366190 A
US20050060138 A1

(73) 특허권자
마이크로소프트 테크놀로지 라이선싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
(72) 발명자
오츠키 가츠토시
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 어텐션 - 인터내셔널 패이턴츠 마이
크로소프트 코포레이션
우메오카 다카시
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 어텐션 - 인터내셔널 패이턴츠 마이
크로소프트 코포레이션
(74) 대리인
제일특허법인

전체 청구항 수 : 총 20 항

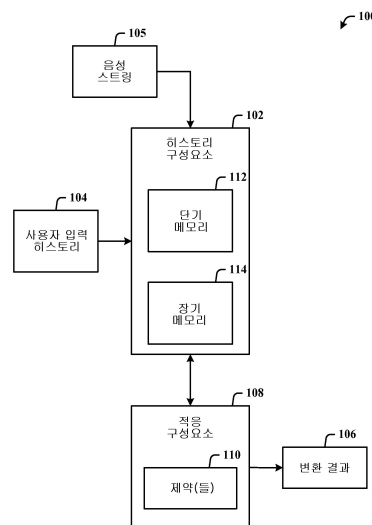
심사관 : 정성윤

(54) 발명의 명칭 컴퓨터구현 음성 방법 및 시스템

(57) 요약

장기 메모리와 단기 메모리에 적절한 제약을 적용함으로써 예상치 못한 단어의 출현을 억제하는 구조를 개시한다. 또한 제약을 활용함으로써 신속한 적응을 실현한다. 구조는 변환 결과를 출력하는 변환 프로세스에 의한 음성 스트링의 변환을 위해서 사용자 입력 히스토리를 처리하는 히스토리 구성요소, 그리고 변환 프로세스 동안에 단어 출현에 영향을 주는 단기 메모리에 적용되는 제약(들)을 기반으로 사용자 입력 히스토리에 변환 프로세스를 적응시키기 위한 적응 구성요소를 포함한다. 구조는 문맥의존 확률 차이(단기 메모리)를 기반으로 확률 증대를 수행하고, 그리고 선행한 단어 문맥의 빈도수(장기 메모리)를 기반으로 한 기준 언어 모델과 장기 메모리 사이에 동적 선행-보간을 수행한다.

대표도



명세서

청구범위

청구항 1

컴퓨터로 구현된 음성 시스템으로서,

변환 프로세스에 의한 음성 스트링(phonetic string)의 변환을 위해 사용자 입력 히스토리(user input history)를 처리하도록 구성된 히스토리 구성요소와,

문맥의존 확률 차이(context-dependent probability differences)에 기초하여 확률을 증대(boosting)시킴으로써 제약(restriction)을 적용하도록 구성된 제약 구성요소와,

상기 사용자 입력 히스토리에 적용된 하나 이상의 제약에 기초하여 상기 변환 프로세스를 적응(adapt)시키도록 구성된 적응 구성요소- 상기 사용자 입력 히스토리는 문맥의존 확률 차이를 단기 기억 정보(short-term memory)로서 포함하고 또한 단어의 선행 문맥의 빈도수(frequency of preceding context of a word)를 장기 기억 정보(long-term memory)로서 포함하며, 상기 적응 구성요소는 예상치 못한 단어 출현을 억제하도록 상기 변환 프로세스 동안 상기 변환 프로세스를 상기 사용자 입력 히스토리에 적응시킴 -와,

상기 히스토리 구성요소, 상기 제약 구성요소 및 상기 적응 구성요소와 연관된 컴퓨터 실행가능 명령어들을 실행하도록 구성된 마이크로프로세서

를 포함하는 컴퓨터로 구현된 음성 시스템.

청구항 2

제1항에 있어서,

상기 제약 구성요소는 문맥의존 확률 차이에 기초하여 확률을 증대시킴으로써 장기 기억 정보에 제약을 적용하는

컴퓨터로 구현된 음성 시스템.

청구항 3

제1항에 있어서,

상기 적응 구성요소는 장기 기억 정보와 기준 언어 모델(baseline language model) 사이에 동적 선형 보간(dynamic linear interpolation)을 수행하는

컴퓨터로 구현된 음성 시스템.

청구항 4

제1항에 있어서,

상기 제약은 단어가 후보 목록의 제1 후보와 다를 때에 상기 단어의 확률을 증대시키는

컴퓨터로 구현된 음성 시스템.

청구항 5

제1항에 있어서,

상기 단기 기억 정보에 적용되는 상기 제약은 문맥인식 단기 기억 정보 바이그램 확률(context-sensitive short-term memory bigram probability)을 이용하는

컴퓨터로 구현된 음성 시스템.

청구항 6

제1항에 있어서,

상기 단기 기억 정보에 적용되는 상기 제약은 단어와 문장 내의 상기 단어의 문맥에 기초하여 확률을 증대시키는

컴퓨터로 구현된 음성 시스템.

청구항 7

제6항에 있어서,

상기 문맥은 상기 문장에서 상기 단어에 대한 선행 문맥과 후행 문맥을 포함하는

컴퓨터로 구현된 음성 시스템.

청구항 8

제1항에 있어서,

상기 적응 구성요소는, 후보 목록의 제1 후보와 상기 후보 목록의 선택된 후보 사이의 차이에 기초하여 플래그-학습(flag-learning)을 수행하고 다음 변환 프로세스에서 상기 선택 후보를 제1 변환 결과 위치로 이동시키는 학습 알고리즘을 포함하는

컴퓨터로 구현된 음성 시스템.

청구항 9

컴퓨터로 구현된 음성 시스템으로서,

변환 프로세스 동안 음성 스트링의 변환을 위해 사용자 입력 히스토리를 처리하도록 구성된 히스토리 구성요소와,

예상치 못한 단어의 출현을 억제하도록 상기 변환 프로세스 동안 상기 사용자 입력 히스토리에 제약을 적용하도록 구성된 제약 구성요소- 상기 제약 구성요소는 문맥의존 확률 차이에 기초하여 확률을 증대시키는 제약을 적용하고, 상기 히스토리는 문맥의존 확률 차이를 단기 기억 정보로서 포함하고 또한 단어의 선행 문맥의 빈도수를 장기 기억 정보로서 포함함 -와,

상기 제약에 기초하여 상기 변환 프로세스를 상기 사용자 입력 히스토리에 적응시키도록 구성된 적응 구성요소와,

상기 히스토리 구성요소, 상기 제약 구성요소 및 상기 적응 구성요소와 연관된 컴퓨터 실행가능 명령어들을 실행하는 마이크로프로세서

를 포함하는 컴퓨터로 구현된 음성 시스템.

청구항 10

제9항에 있어서,

상기 제약 구성요소는 문맥인식 단기 기억 정보 바이트맵 확률을 이용하는 제약을 상기 단기 기억 정보에 적용하고, 문맥의존 확률 차이에 기초하여 확률을 증대시키는 제약을 상기 장기 기억 정보에 적용하는

컴퓨터로 구현된 음성 시스템.

청구항 11

제9항에 있어서,

상기 적응 구성요소는 장기 기억 정보와 기준 언어 모델 사이에 동적 선형 보간을 수행하는

컴퓨터로 구현된 음성 시스템.

청구항 12

제9항에 있어서,

상기 제약은 단어가 후보 목록의 제1 후보와 다를 때에 상기 단어의 확률을 증대시키는 컴퓨터로 구현된 음성 시스템.

청구항 13

제9항에 있어서,

상기 단기 기억 정보에 적용되는 상기 제약은 단어와 문장 내의 상기 단어의 문맥에 기초하여 확률을 증대시키고, 상기 문맥은 상기 문장에서 상기 단어에 대한 선행 문맥과 후행 문맥을 포함하는 컴퓨터로 구현된 음성 시스템.

청구항 14

제9항에 있어서,

상기 적응 구성요소는, 후보 목록의 제1 후보와 상기 후보 목록의 선택된 후보 사이의 차이에 기초하여 플래그-학습을 수행하고 다음 변환 프로세스에서 제1 변환 결과 위치로 상기 선택된 후보를 이동시키는 학습 알고리즘을 포함하는 컴퓨터로 구현된 음성 시스템.

청구항 15

머신 관독가능한 명령어들을 실행하는 컴퓨터 시스템에 의해 수행되는 컴퓨터로 구현된 음성 방법으로서, 변환 프로세스 동안 음성 스트링의 변환을 위해 사용자 입력 히스토리를 처리하는 단계와,

예상치 못한 단어의 출현을 억제하도록 상기 변환 프로세스 동안 상기 사용자 입력 히스토리에 제약을 적용하는 단계- 상기 히스토리는 문맥의존 확률 차이를 단기 기억 정보로서 포함하고 또한 단어의 선행 문맥의 빈도수를 장기 기억 정보로서 포함하고, 상기 문맥의존 확률 차이에 기초하여 확률을 증대시키는 제약이 적용됨 -와, 상기 제약에 기초하여 상기 변환 프로세스를 상기 사용자 입력 히스토리에 적응시키는 단계를 포함하는 컴퓨터로 구현된 음성 방법.

청구항 16

제15항에 있어서,

장기 기억 정보와 기준 언어 모델 사이에 동적 선형 보간을 수행하는 단계를 더 포함하는 컴퓨터로 구현된 음성 방법.

청구항 17

제15항에 있어서,

단어가 후보 목록의 제1 후보와 다를 때에 상기 단어의 확률을 증대시키는 단계를 더 포함하는 컴퓨터로 구현된 음성 방법.

청구항 18

제15항에 있어서,

단어와 문장 내의 상기 단어의 문맥에 기초하여 확률을 증대시키는 제약을 상기 단기 기억 정보에 적용하는 단계를 더 포함하는 컴퓨터로 구현된 음성 방법.

청구항 19

제18항에 있어서,

상기 문맥은 상기 문장에서 상기 단어에 대한 선행 문맥과 후행 문맥을 포함하는 컴퓨터로 구현된 음성 방법.

청구항 20

제15항에 있어서,

후보 목록의 제1 후보와 상기 후보 목록의 선택된 후보 사이의 차이에 기초하여 플래그-학습을 수행하는 단계와,

다음 변환 프로세스에서 제1 변환 결과 위치로 상기 선택된 후보를 이동시키는 단계를 더 포함하는 컴퓨터로 구현된 음성 방법.

발명의 설명

배경 기술

[0001] 입력 방법은 예를 들어 음성 스트링(phonetic string)(판독)을 중국어, 한국어 및 일본어와 같은 동아시아 언어를 위한 디스플레이 문자로 변환하고, 또한 중국어 변체와 같은 획을 처리하도록 사용될 수 있다. 동음이의어와 가능한 다양한 단어 분절로 인하여 변환시에 모호성이 존재한다. 입력 방법은 일반(예를 들어 기준(baseline), 디폴트) 언어 모델과 사용자 입력 히스토리(user input history)를 기반으로 이 모호성을 해결하려고 한다. 사용자 입력 히스토리에 대한 적응(adaptation)은 몇몇 방식으로, 예를 들면 단기 메모리(short-term memory)와 장기 메모리(long-term memory)로 수행될 수 있다. 단기 메모리는 적응의 신속성에 대응하고, 장기 메모리는 적응의 안정성에 대응한다. 변환 결과는 단기 및 장기 메모리로부터의 정보를 일반 언어 모델에 추가함으로써 결정된다.

[0002] 단기 메모리는 단어의 이전 사용자 선택(사용자 입력 히스토리)을 기반으로 단어 순위를 변경하거나 단어 점수를 증대(boosting)시킴으로서 구현될 수 있다. 그러나 일부 단어는 사용된 후에 충분히 빨리 나타나지 않으며, 일부 단어는 사용된 후에 수용불가한 문맥에서 갑자기 나타난다. 장기 메모리는 사용자 입력 히스토리를 누적함으로써 구현될 수 있다. 그러나 일부 단어는 여전히 장기 메모리의 사용에도 불구하고 수용불가한 문맥에서 갑자기 나타난다.

발명의 내용

과제의 해결 수단

[0003] 다음은 여기에 기술된 신규의 소정 실시예에 대한 기본적인 이해를 위하여 간단한 요약を提供한다. 이 요약은 폭넓은 개요가 아니며, 핵심적인/중대한 요소를 확인하거나 또는 그의 범주를 기술하려는 것이 아니다. 이의 유일한 목적은 차후에 제공되는 보다 상세한 설명에 대한 서두로서 간단한 형태의 일부 개념을 제공하기 위한 것이다.

[0004] 개시된 구조는 장기 및 단기 메모리에 적절한 제약(restriction)을 적용함으로써 단어의 예상치 않은 출현을 억제시킨다. 또한 제약을 활용하여 적응을 신속히 실현시킬 수 있다.

[0005] 이 구조는 변환 결과를 출력하는 변환 프로세스에 의한 음성 스트링의 변환을 위해 사용자 입력 히스토리를 처리하기 위한 히스토리 구성요소, 그리고 변환 프로세스 동안에 단어 출현에 영향을 주는 단기 메모리에 적용되는 제약(들)을 기반으로 사용자 입력 히스토리에 변환 프로세스를 적응시키기 위한 적응 구성요소를 포함한다. 구조는 문맥의존 확률 차이(context-dependent probability differences)(단기 메모리)를 기반으로 확률 증대(probability boosting)를 수행하고, 그리고 단어의 선행 문맥의 빈도수(장기 메모리)를 기반으로 한 기준 언어 모델과 장기 메모리 사이에 동적 선형 보간(dynamic linear-interpolation)을 수행한다.

[0006] 진술한 관련 목적을 성취하기 위하여, 다음의 설명과 첨부된 도면과 관련하여 소정의 설명 양상을 여기에 기술한다. 이들 양상은 여기에 개시된 원리를 실행할 수 있는 다양한 방법을 나타내며, 모든 양상과 이의 등가물은

청구 주제의 범주내에 있다. 다른 장점과 신규 특징은 도면과 함께 고려시에 다음의 상세한 설명으로부터 명백해질 것이다.

도면의 간단한 설명

- [0007] 도 1은 개시된 구조에 따르는 컴퓨터 구현 음성 시스템을 도시하는 도면.
 도 2는 도 1의 음성 시스템의 추가 양상을 포함한 시스템을 도시하는 도면.
 도 3은 가중치 변화에 대한 그래프를 도시하는 도면.
 도 4는 캐시 가중치 변화에 대한 그래프를 도시하는 도면.
 도 5는 컴퓨터 구현 음성 방법을 도시하는 도면.
 도 6은 도 5의 방법의 추가 양상을 도시하는 도면.
 도 7은 도 5의 방법의 추가 양상을 도시하는 도면.
 도 8은 개시된 구조에 따라서 통계 언어 모델을 위한 신속하고 안정된 적응을 실행하기 위해 동작가능한 컴퓨팅 시스템의 블록도.

발명을 실시하기 위한 구체적인 내용

- [0008] 일반 시나리오에서 기존의 음성 시스템의 변환 정확성이 높을 수 있겠지만, 사용자는 특정 사용자의 언어 공간이 일반적 공간과 다르므로 여전히 실망한다. 이것은 특히 개인 이름인 경우에 그러하며, 표현 선호도는 당연히 사용자에 따라 변하며, 따라서 일반 언어 모델에 의해 다루어질 수 없다.
- [0009] 개시된 구조는 사용자가 짧은 시간 주기(예를 들어 2주 내지 3 주) 동안에 제품을 사용한 후에 더 이상 후보 목록을 개방할 필요없는 자체조정(self-tuning) 기법이다. 또한 개시된 자체조정 기법은 사용자의 작업 능력을 개선한다. 이 구조는 문맥의존 확률 차이(단기 메모리)를 기반으로 확률 증대를 수행하고, 그리고 단어의 선행 문맥의 빈도(장기 메모리)를 기반으로 장기 메모리와 기준 언어 모델 간에 동적 선행 보간을 수행한다.
- [0010] 이제 도면을 참조하며, 여기서 동일한 참조번호는 도면 모두에서 동일한 요소를 참조하는데 사용된다. 다음의 서술에서 설명을 위하여, 다수의 특정한 상세사항은 이의 철저한 이해를 돕기 위하여 제시된다. 그러나 신규의 실시예는 이들 특정한 상세사항 없이도 실행될 수 있다는 것은 분명할 수 있다. 다른 경우, 잘 알려진 구조 및 장치는 이의 설명을 용이하게 하기 위하여 블록도 형태로 도시된다. 본 발명의 청구 주제의 사상 및 범주 내에 있는 모든 변경, 등가물 및 대안들을 포함하려 한다.
- [0011] 도 1은 개시된 구조에 따라서 컴퓨터 구현 음성 시스템(100)을 도시한다. 시스템(100)은 변환 결과(106)를 출력하는 변환 프로세스에 의한 음성 스트링(105)의 변환을 위해 사용자 입력 히스토리(104)를 처리하는 히스토리 구성요소(102), 그리고 변환 프로세스 동안에 단어 출현에 영향을 주는 단기 메모리(112)에 적용되는 제약(들)을 기반으로 사용자 입력 히스토리(104)에 변환 프로세스를 적응시키기 위한 적응 구성요소(108)를 포함한다.
- [0012] 적응 구성요소(108)는 장기 메모리(114)를 기반으로 한 기준 언어 모델과 장기 메모리(114) 사이에 동적 선행 보간을 수행한다. 제약(들)(110)은 단어가 후보 목록의 제1 후보와 다를 때에 그 단어의 확률을 증대시킨다. 단기 메모리(112)에 적용되는 제약(들)(110)은 문맥인식 단기 메모리 바이그램 확률(context-sensitive short-term memory bigram probability)을 이용한다. 단기 메모리(112)에 적용되는 제약(들)(110)은 문장에서 단어와 단어의 문맥을 기반으로 확률을 증대시킨다. 문맥은 문장내의 단어에 대한 선행 문맥 및 후행 문맥을 포함한다. 적응 구성요소(108)는 후보 목록의 제1 후보와 후보 목록의 선택된 후보 간의 차이를 기반으로 플래그-학습(flag-learning)을 수행하고, 다음 변환 프로세스에서 선택된 후보를 제1 변환 결과 위치로 이동시키는 학습 알고리즘을 포함한다.
- [0013] 도 2는 도 1의 음성 시스템(100)의 추가 양상을 포함한 시스템(200)을 도시한다. 시스템(200)은 변환 프로세스(204)에 의한 음성 스트링(105)의 변환을 위해 사용자 입력 히스토리(104)를 처리하는 히스토리 구성요소(102), 그리고 변환 프로세스(204) 동안에 단어 출현에 영향을 주는 단기 메모리(112)에 적용되는 제약(들)(110)을 기반으로 사용자 입력 히스토리(104)에 변환 프로세스(204)를 적응시키는 적응 구성요소(108)를 포함한다.

- [0014] 적응 구성요소(108)는 장기 메모리(114)를 기반으로 한 기준 언어 모델(208)과 장기 메모리(114)사이에 동적 선행 보간을 수행한다. 제약(들)(110)은 단어가 후보 목록의 제1 후보와 다를 때에 단어의 확률을 증대시킨다. 단기 메모리(112)에 적용되는 제약(들)(110)은 문맥인식 단기 메모리 바이그램 확률을 사용한다. 단기 메모리(112)에 적용되는 제약(들)(110)은 문장에서 단어와 단어의 문맥을 기반으로 확률을 증대시킨다. 문맥은 문장 내의 단어에 대한 선행 문맥 및 후행 문맥을 포함한다. 적응 구성요소(108)는 후보 목록의 제1 후보와, 후보 목록의 선택된 후보 사이의 차이를 기반으로 플래그 학습을 수행하고, 다음 변환 프로세스에서 선택된 후보를 제1 변환 결과로 이동시키는 학습 알고리즘을 포함한다.
- [0015] 시스템(200)은 문맥의존 확률 차이를 기반으로 확률을 증대시킴으로써 제약(들)(110)을 적용시키기 위한 제약 구성요소(206)를 더 포함한다. 또한 제약 구성요소(206)는 문맥의존 확률 차이를 기반으로 확률을 증대시킴으로써 하나 이상의 제약(들)(110)을 장기 메모리(114)에 적용할 수 있다.
- [0016] 또 다른 방식에서, 음성 시스템(200)은 변환 프로세스(204) 동안에 음성 스트링(105)의 변환을 위해 사용자 입력 히스토리(104)를 처리하기 위한 히스토리 구성요소(102), 그리고 변환 프로세스(204) 동안에 하나 이상의 제약(들)(110)을 사용자 입력 히스토리(104)에 적용하기 위한 제약 구성요소(206)를 포함한다. 히스토리(104)는 단기 메모리(112)와 장기 메모리(114)를 포함한다. 또한 시스템(200)은 제약(들)(110)을 기반으로 사용자 입력 히스토리(104)에 변환 프로세스(204)를 적용시키기 위한 적응 구성요소(108)를 포함한다.
- [0017] 제약 구성요소(206)는 하나 이상의 제약(들)(110)을 단기 메모리(112)에 적용한다. 적용된 제약(들)(110)은 문맥인식 단기 메모리 바이그램 확률, 그리고 문맥의존 확률 차이를 기반으로 확률을 증대시키는 장기 메모리(114)에 대한 하나 이상의 제약(들)을 사용한다. 적응 구성요소(108)는 장기 메모리(114)를 기반으로 한 기준 언어 모델(208)과 장기 메모리(114)사이에 동적 선행 보간을 수행한다. 제약(들)(110)은 단어가 후보 목록의 제1 후보와 다를 때에 단어의 확률을 증대시킨다. 단기 메모리(112)에 적용되는 제약(들)(110)은 단어, 그리고 문장 내의 그 단어의 문맥을 기반으로 확률을 증대시킨다. 문맥은 문장 내의 단어에 대한 선행 문맥 및 후행 문맥을 포함한다.
- [0018] 다음은 통계 언어 모델을 위한 신속하고 안정된 적응을 위해 사용되는 계산을 상세히 설명한다.
- [0019] 입력 음성 스트링에 대한 입력 방법 변환 결과는 다음의 확률에 의해 결정될 수 있다:
- [0020]
$$P(W) = P(w_1 | < s >) \cdot P(w_2 | w_1) \cdot P(w_3 | w_2) \dots P(w_N | w_{N-1}) \cdot P(< /s > | w_N)$$
- [0021] 여기서 $\mathbb{W}$ 는 단어열 $\{w_1, w_2, w_3, \dots, w_{N-1}, w_N\}$ 을 포함한 문장이고, $< s >$ 와 $< /s >$ 는 각각 문장 시작과 문장 끝을 위한 기호이다. 수학적식은 바이그램 모델을 위한 것이며, 트리그램(trigram) 또는 보다 높은 차수 n그램 모델로써 표현될 수 있다.
- [0022] 동음이의어와 단어 분절의 모호성으로 인하여 입력 음성 스트링에 대한 가능한 다수의 단어열 $\mathbb{W}$ 가 있을 수 있다.
- [0023] 가장 확률있는 후보 문장을 변환 결과로서 선택한다.
- [0024]
$$\hat{W} = \arg \max_W P(W)$$
- [0025] 각 단어에 대한 확률은 다음과 같이 정의될 수 있다:
- [0026]
$$P(w_n | w_{n-1}) = \alpha \cdot P_{\text{baseline}}(w_n | w_{n-1}) + \beta \cdot P_{\text{ltm}}(w_n | w_{n-1}) + \delta \cdot P_{\text{stm}}(w_n | w_{n-1})$$
- [0027] 여기서 α, β 및 δ 는 1로 합산되는 선형 보간 계수이고 ($\alpha + \beta + \delta = 1$), $P_{\text{baseline}}(w_k | w_{k-1})$ 는 훈련 텍스트 데이터베이스로부터 추정되는 기준 바이그램 확률이고(처음 입력 방법을 사용시에, 오직 이 확률만 값을 가짐), $P_{\text{ltm}}(w_n | w_{n-1})$ 은 장기 메모리에 대한 바이그램 확률이고, 그리고 $P_{\text{stm}}(w_n | w_{n-1})$ 는 단기 메모리에 대한 바이그램 확률이다. 장기 메모리에 대한 바이그램 확률은 다음과 같이 사용자 입력 히스토리로부터 계산될 수 있다:
- [0028]
$$P_{\text{ltm}}(w_n | w_{n-1}) = \frac{C_{\text{user}}(w_{n-1}, w_n)}{C_{\text{user}}(w_n)}$$
- [0029] 여기서 $C_{\text{user}}(w_n)$ 은 사용자가 단어 w_n 를 사용한 횟수이고, $C_{\text{user}}(w_{n-1}, w_n)$ 은 사용자가 단어열 w_{n-1}, w_n 을 사용하는

횃수이다.

[0030] 단기 메모리에 대한 바이그램 확률, $P_{stm}(w_n|w_{n-1})$ 는 단어가 결과의 제1 후보가 아니지만, 사용자가 후보 리스트로부터 그 단어를 선택할 때, 그 단어에 대한 확률을 증대시킨다.

[0031]
$$P_{stm}(w_n|w_{n-1}) = \frac{C_{user-sel}(w_{n-1}, w_n)}{M}$$

[0032] 여기서 $C_{user-sel}(w_{n-1}, w_n)$ 은 사용자가 후보 목록으로부터 단어열 w_{n-1}, w_n 을 선택하는 횃수이고, M은 선택을 위한 최대 카운트이다. $C_{user-sel}(\cdot)$ 은 M을 초과할 수 없다는 데에 주목한다.

[0033] 전술한 수확식은 다음과 같이 표현되는 먹승에 의해 일반화될 수 있다:

[0034]
$$P_{stm}(w_n|w_{n-1}) = \left(\frac{C_{user-sel}(w_{n-1}, w_n)}{M} \right)^k$$

[0035] 다음은 장기 메모리에 대한 추가 설명이다. 단어 w_n 에 대한 선형 보간 가중치 α 및 β 는 $C_{user}(w_{n-1})$ 에 따라 변한다. 이것은 가중치가 이전 단어에 따라 다르다는 것을 의미한다.

[0036] 목표 가중치 α_{target} 및 β_{target} 는 $C_{user}(w_{n-1})$ 가 충분히 클 때 정의되고 사용된다. w_n 에 대한 실제 가중치 α 및 β 는 다음과 같이 계산될 수 있다:

[0037]
$$\beta = \omega \cdot \beta_{target}$$

$$\alpha = \alpha_{target} + \beta_{target} - \beta = \alpha_{target} + (1 - \omega) \cdot \beta_{target}$$

$$\omega = \begin{cases} \frac{1}{t} \cdot C_{user}(w_{n-1}) & \text{if } C_{user}(w_{n-1}) < t \\ 1 & \text{그 외에} \end{cases}$$

[0038] 도 3은 가중치 변화에 대한 그래프(300)를 도시한다. 그래프(300)는 단기 메모리 δ , 장기 메모리 β 및 기준 α 에 대한 상대적 수직 범위 세그먼트를 도시하는데, 장기 메모리 β 는 β_{target} 를 지정하고, 기준은 α_{target} 를 지정한다. 그래프(300)는 단어가 사용되는 횃수가 증가함에 따라, 시간 t에서 장기 메모리에 대한 가중치는 β_{target} 에 도달하는 것을 나타낸다.

[0039] $C_{user}(w_{n-1})$ 가 작을 때, 장기 바이그램 확률이 높아지는 경향이 있으며, 예상치 못한 단어 출현이 일어난다. 그러나 이 가중치 조정은 이들 종류의 부작용을 억제할 수 있다.

[0040] 다음은 단기 메모리에 대한 추가 설명이다. 두 접근방안을 개별적으로 또는 결합하여 사용할 수 있다: 단기 메모리 바이그램 확률의 문맥인식 사용, 그리고 확률 차이에 따라 증대되는 확률.

[0041] 제1 접근방안, 단기 메모리 바이그램 확률의 문맥인식 사용에서, 선택된 카운트가 후행 단어열에 대해 0일 때에 확률은 0으로 간주된다.

[0042]
$$P_{stm}(w_n|w_{n-1}) = \begin{cases} \frac{C_{user-sel}(w_{n-1}, w_n)}{M}, & \text{if } C_{user-sel}(w_n, w_{n+1}) > 0 \\ 0 & \text{그 외에} \end{cases}$$

[0043] 유사한 결과는 선행 단어열을 사용하여 얻을 수 있다.

[0044]
$$P_{stm}(w_n|w_{n-1}) = \begin{cases} \frac{C_{user-sel}(w_{n-1}, w_n)}{M}, & \text{if } C_{user-sel}(w_{n-2}, w_{n-1}) > 0 \\ 0 & \text{그 외에} \end{cases}$$

[0045] 이들 조건은 단어의 POS(part-of-speech)에 따라 변경될 수 있다. 확률 증대가 이들 조건을 기반으로 문맥에 의존하고, 이전에 선택한 단어의 예상치 못한 출현을 억제할 수 있다.

[0046] 제2 접근방안, 확률 차이에 의존하는 확률 증대에서, $C_{user-sel}(w_{n-1}, w_n)$ 을 1 씩 증분하는 것은 일부 단어에 불충분할 수 있고 다른 단어에는 너무 과할 수 있다. $C_{user-sel}(w_{n-1}, w_n)$ 의 적절한 증분 수는 단어 및 단어의 문맥에 의존한다.

[0047] 사용자가 후보 목록으로부터 단어를 선택하는 이유는 문장이 단어를 포함하는 확률이 다른 문장이 그 단어를 포

합하는 확률보다 낮기 때문이다. 따라서 다음에 단어를 얻기 위하여, 문장이 그 단어를 포함한 확률은 다른 문장이 그 단어를 포함하는 확률보다 더 높아야 한다(이전 변환에서 첫 문장).

도 4는 캐시 가중치 변화를 위한 그래프(400)를 도시한다. 다른 실시예에서, 캐시 가중치 변화는 선형 함수를 사용하여 제공되고, 캐시 가중치는 단지 바이그램 캐시(바이캐시:bicache)를 위한 것이다. 바이그램 캐시 가중치는 선행 단어의 유니그램 캐시(unicache) 양에 의존한다. 이것은 바이그램 캐시 확률에 대한 가중치 $P_{bicache}(w_i|w_{i-1})$ 는 $C_{unicache}(w_{i-1})$ 에 의존한다는 것을 의미한다.

$$\omega = \begin{cases} \frac{1}{t} \cdot \text{CacheAmount}(w_{i-1}) & \text{if CacheAmount} < t \\ 1 & \text{그 외에} \end{cases}$$

플래그 가중치 $\delta + \varepsilon$ 는 일정하다. 유니그램 캐시에 대한 가중치도 일정하지만, 오프셋 값은 이전의 캐시에 의한 부작용을 감소시키기 위하여 총 유니그램 캐시 카운트에 추가된다.

$$P_{unicache}(w_i) = P_{unicache-smooth}(w_i) = \frac{C(w_i)}{\sum_k C(w_k) + \text{UniCacheOffset}}$$

$$P_{unicache-raw}(w_i) = \frac{C(w_i)}{\sum_k C(w_k)}$$

플래그 학습은 확률 차이에 의존한다. 바이그램 플래그의 증가 레벨은 제1 후보와 선택된 후보 사이에 추정되는 차이량에 따라 변한다. 주위 문맥이 동일한 경우에, 선택된 후보가 다음 제1 후속 변환 결과가 된다.

다음 경우를 고려할 수 있고, 아래의 알고리즘은 모든 경우를 포함한다.

Case #1: $\{w_a, w_b, w_c\}$ after conversion $\rightarrow \{w_a, w_x, w_c\}$ after editing

Case #2: $\{w_a, w_{b1}...w_{bm}, w_c\}$ after conversion $\rightarrow \{w_a, w_x, w_c\}$ after editing

Case #3: $\{w_a, w_b, w_c\}$ after conversion $\rightarrow \{w_a, w_{x1}...w_{xn}, w_c\}$ after editing

Case #4: $\{w_a, w_{b1}...w_{bm}, w_c\}$ after conversion $\rightarrow \{w_a, w_{x1}...w_{xn}, w_c\}$ after editing

다음 정의를 제공한다.

$P(w_b|w_a)$ 는 기준, 캐시 및 플래그 확률을 포함한 학습 전의 단어 바이그램 확률이다.

$$P(w_b|w_a) = \alpha \cdot P_{baseline}(w_b|w_a) + \beta \cdot P_{cache}(w_b|w_a) + \gamma \cdot P_{cache}(w_b) + \delta \cdot P_{flag}(w_b|w_a) + \varepsilon \cdot P_{flag}(w_b)$$

$P_L(w_b|w_a)$ 는 학습 후의 단어 바이그램 확률이다. 단순화를 위해 여기서 캐시 확률의 변경을 무시하며, 플래그 확률만이 학습후에 변경된다.

$$P_L(w_b|w_a) = \alpha \cdot P_{baseline}(w_b|w_a) + \beta \cdot P_{cache}(w_b|w_a) + \gamma \cdot P_{cache}(w_b) + \delta \cdot P_{flag}(w_b|w_a)_{-1} + \varepsilon \cdot P_{flag}(w_b)_{-1}$$

사용자가 후보 목록으로부터 다른 후보를 선택할 때에 제1 후보인 후보 단어에 대한 플래그 카운트는 학습 후에 1만큼 감분된다.

$$P_L(w_x|w_a) = \alpha \cdot P_{baseline}(w_x|w_a) + \beta \cdot P_{cache}(w_x|w_a) + \gamma \cdot P_{cache}(w_x) + \delta \cdot P_{flag}(w_x|w_a)_{+d} + \varepsilon \cdot P_{flag}(w_x)_{+1}$$

후보 단어는 후보 목록으로부터 선택된 대응되는 후보 단어에 대한 유니그램 플래그 카운트는 1만큼 증분된다. 후보 목록으로부터 선택된 대응되는 후보 단어에 대한 바이그램 플래그 카운트가 증분되고, 증분량은 추후에 결정한다.

알고리즘에 대해, 학습 전에, 제1 후보와 선택된 후보 사이의 크기 관계는 다음과 같다.

$$P(w_b|w_a) \cdot P(w_c|w_b) > P(w_x|w_a) \cdot P(w_c|w_x)$$

[0065] 학습 후에 크기는 다음과 같이 된다:

$$[0066] \quad P_L(w_b|w_a) \cdot P_L(w_c|w_b) < P_L(w_x|w_a) \cdot P_L(w_c|w_x)$$

[0067] 확률 $P_L(w_b|w_a) \cdot P_L(w_c|w_b)$ 와 $P(w_x|w_a) \cdot P(w_c|w_x)$ 는 알고 있으며, $P_L(w_x|w_a) \cdot P_L(w_c|w_x)$ 를 계산하는 방법이 요구된다.

[0068] 학습에 의한 확률 변경은 먹승으로서 표현될 수 있다(또는 먹).

$$[0069] \quad P_L(w_x|w_a) \cdot P_L(w_c|w_x) = [P(w_x|w_a) \cdot P(w_c|w_x)]^\varphi$$

[0070] 따라서

$$[0071] \quad P_L(w_b|w_a) \cdot P_L(w_c|w_b) < [P(w_x|w_a) \cdot P(w_c|w_x)]^\varphi$$

[0072] 그러면 ϕ 는 다음과 같이 계산될 수 있다.

$$[0073] \quad \varphi > \frac{\log[P(w_x|w_a) \cdot P(w_c|w_x)]}{\log[P_L(w_b|w_a) \cdot P_L(w_c|w_b)]}$$

[0074] 이제, ϕ 로부터 $P_{\text{flag}}(w_x|w_a)+d$ 를 계산한다.

$$[0075] \quad P_L(w_x|w_a) \cdot P_L(w_c|w_x) = [P(w_x|w_a) \cdot P(w_c|w_x)]^\varphi = P(w_x|w_a)^\varphi \cdot P(w_c|w_x)^\varphi$$

[0076] 아래의 등식이 만족되면, 위의 등식이 만족된다.

$$[0077] \quad P_L(w_x|w_a) = P(w_x|w_a)^\varphi \text{ and } P_L(w_c|w_x) = P(w_c|w_x)^\varphi$$

[0078] $P_L(w_x|w_a) = P(w_x|w_a)^\varphi$ 를 고려한다.

$$\begin{aligned} P(w_x|w_a) &= \alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) \\ &\quad + \delta \cdot P_{\text{flag}}(w_x|w_a) + \varepsilon \cdot P_{\text{flag}}(w_x) \\ P_L(w_x|w_a) &= \alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) \\ &\quad + \delta \cdot P_{\text{flag}}(w_x|w_a)+d + \varepsilon \cdot P_{\text{flag}}(w_x)+1 \\ \alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) + \delta \cdot P_{\text{flag}}(w_x|w_a)+d + \varepsilon \\ &\quad \cdot P_{\text{flag}}(w_x)+1 \\ &= [\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) \\ &\quad + \delta \cdot P_{\text{flag}}(w_x|w_a) + \varepsilon \cdot P_{\text{flag}}(w_x)]^\varphi \\ \delta \cdot P_{\text{flag}}(w_x|w_a)+d \\ &= [\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) \\ &\quad + \delta \cdot P_{\text{flag}}(w_x|w_a) + \varepsilon \cdot P_{\text{flag}}(w_x)]^\varphi \\ &\quad - [\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) + \varepsilon \\ &\quad \cdot P_{\text{flag}}(w_x)+1] \\ P_{\text{flag}}(w_x|w_a)+d \\ &= \frac{[\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) + \delta \cdot P_{\text{flag}}(w_x|w_a) + \varepsilon \cdot P_{\text{flag}}(w_x)]^\varphi}{\delta} \\ &\quad - \frac{[\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) + \varepsilon \cdot P_{\text{flag}}(w_x)+1]}{\delta} \end{aligned}$$

[0079]

[0080] 확률의 차이를 기반으로 증분 d를 계산할 수 있다.

$$[0081] \quad P_{\text{flag}}(w_x|w_a)+d - P_{\text{flag}}(w_x|w_a)$$

[0082] 플래그 학습 확률은 플래그-카운트에 대응하여 준비된다. 플래그-카운트의 범위는 예를 들어 8, 16 또는 32일 수 있다. 카운트의 수가 많을수록, 이 알고리즘이 더 정확하게 작동한다.

- [0083] 개시된 구조의 신규 양상을 수행하기 위한 예시적인 방법론을 나타내는 흐름도의 집합을 여기에 포함시킨다. 설명을 간단하게 하기 위하여, 예를 들어 흐름도의 형태로 여기에 도시된 하나 이상의 방법론은 일련의 액트(acts)로서 도시하고 기술하였지만, 소정 액트는 이에 따라서 상이한 순서로 발생할 수 있고, 및/또는 여기에 도시 및 기술된 다른 액트와 동시에 발생할 수 있으므로, 방법론은 액트 순서에 의해 제한되지 않는다는 것을 이해하고 알 것이다. 예를 들면 당업자는 방법론은 상태도에서와 같이 일련의 상호관련 상태 또는 이벤트로서 대신 표현될 수 있다는 것을 이해하고 알 것이다. 또한 신규의 구현을 위해 방법론에 도시된 모든 액트를 필요로 하지 않을 수도 있다.
- [0084] 도 5는 컴퓨터 구현 음성 방법을 도시한다. 단계(500)에서, 사용자 입력 히스토리는 변환 프로세스 동안에 음성 스트링의 변환을 위해 처리된다. 단계(502)에서, 변환 프로세스 동안에, 단기 메모리와 장기 메모리를 포함하는 사용자 입력 히스토리에 제약을 적용한다. 단계(504)에서, 제약을 기반으로 사용자 입력 히스토리에 변환 프로세스를 적응시킨다.
- [0085] 도 6은 도 5의 방법의 추가 양상을 도시한다. 단계(600)에서, 문맥의존 확률 차이를 기반으로 확률을 증대시키는 제약을 적용한다. 단계(602)에서, 장기 메모리를 기반으로 한 기준 언어 모델과 장기 메모리 사이에 동적 선형 보간을 수행한다. 단계(604)에서, 단어가 후보 목록의 제1 후보와 다를 때에, 단어의 확률을 증대시킨다.
- [0086] 도 7은 도 5의 방법의 추가 양상을 도시한다. 단계(700)에서, 단어와 문장 내에서 단어의 문맥을 기반으로 확률을 증대시키는 단기 메모리에 제약을 적용한다. 단계(702)에서, 후보 목록의 제1 후보와 후보 목록의 선택된 후보 사이의 차이를 기반으로 플래그 학습을 수행한다. 단계(704)에서, 다음 변환 프로세스에서 제1 변환 결과 위치로 선택된 후보를 이동시킨다.
- [0087] 본 출원에서는, 용어 "구성요소"와 "시스템"은 컴퓨터관련 항목인 하드웨어, 하드웨어와 소프트웨어의 조합, 소프트웨어 또는 실행되고 있는 소프트웨어를 지칭하기 위해 사용된다. 예를 들어 구성요소는 프로세서상에서 실행하는 프로세스, 프로세서, 하드 디스크 드라이브, (광, 고체상태 및/또는 자기 저장매체의) 다수의 저장 드라이브, 객체, 실행가능명령, 실행 스레드, 프로그램, 및/또는 컴퓨터일 수 있지만, 이로 제한되지는 않는다. 설명을 위해, 서버상에서 실행중인 애플리케이션과 서버 모두가 구성요소일 수 있다. 하나 이상의 구성요소는 프로세스 및/또는 실행 스레드내에서 상주할 수 있고, 구성요소는 하나의 컴퓨터에 국한될 수 있고, 그리고/또는 둘 이상의 컴퓨터들 간에 분산될 수 있다. 단어 "예시적인(exemplary)"은 예, 경우, 또는 설명으로서의 기능을 하는 것을 의미하기 위해 여기서 사용될 수 있다. "예시적인"으로 여기에 기술된 임의의 양상 또는 설계가 다른 양상 또는 설계에 비해 반드시 선호되거나 또는 유리하게 구성될 필요는 없다.
- [0088] 이제 도 8을 참조하면, 개시된 구조에 따라서 통계 언어 모델을 위해 신속하고 안정된 적응을 실행하기 위해 동작가능한 컴퓨팅 시스템(800)의 블록도를 도시한다. 이의 다양한 양상에 대한 추가 문맥을 제공하기 위하여, 도 8과 다음의 논의는 다양한 양상을 구현할 수 있는 적절한 컴퓨팅 시스템(800)에 대한 간단하고 일반적인 설명을 제공하려는 것이다. 전술한 설명이 하나 이상의 컴퓨터상에서 실행될 수 있는 컴퓨터 실행가능한 인스트럭션의 일반적이지만, 당업자는 다른 프로그램과 결합하여, 및/또는 하드웨어와 소프트웨어의 조합으로서 신규의 실시예를 구현할 수 있다는 것을 알 것이다.
- [0089] 다양한 양상을 구현하기 위한 컴퓨팅 시스템(800)은 처리 유닛(들)(804), 시스템 메모리(806) 및 시스템 버스(808)를 가진 컴퓨터(802)를 포함한다. 처리 유닛(들)(804)은 단일 프로세서, 다중 프로세서, 단일 코어 유닛 및 다중 코어 유닛과 같이 상업적으로 사용가능한 다양한 프로세서들 중의 임의의 프로세서일 수 있다. 또한 당업자는 신규 방법들이 미니컴퓨터, 메인프레임 컴퓨터뿐만 아니라 퍼스널 컴퓨터(예를 들면 데스크탑, 랩탑 등), 핸드헬드 컴퓨팅 장치, 마이크로프로세서기반 또는 프로그램가능 가전제품 등을 포함한 다른 컴퓨터 시스템 구성에서 실행될 수 있다는 것을 알 것이며, 위 예의 각각은 하나 이상의 관련 장치와 동작가능하게 연결될 수 있다.
- [0090] 시스템 메모리(806)는 VOL(volatile memory)(810)(예를 들면 RAM(random access memory))와 NON-VOL(non-volatile memory)(812)(예를 들면 ROM, EPROM, EEPROM 등)을 포함할 수 있다. BIOS(basic input/output system)는 비휘발성 메모리(812)에 저장될 수 있고, 시동 동안 같은 때에 컴퓨터(802)내 구성요소들 간에 데이터 및 신호의 전달을 용이하게 하는 기본 루틴을 포함한다. 또한 휘발성 메모리(810)는 데이터를 캐싱(caching)하기 위한 정적 RAM과 같은 고속 RAM을 포함할 수 있다.
- [0091] 시스템 버스(808)는 메모리 서브시스템(806)과 처리 유닛(들)(804)을 포함하지만 이로 제한되지 않는 시스템 구성요소를 위한 인터페이스를 제공한다. 시스템 버스(808)는 각종 상업적으로 사용가능한 버스 구조 중의 임의

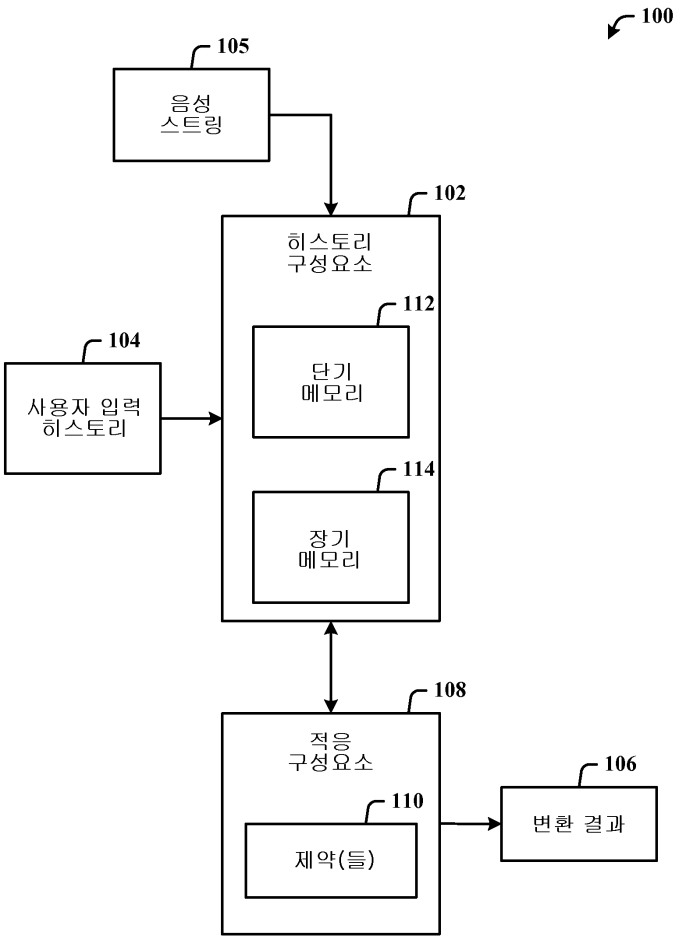
의 버스 구조를 사용하여 (메모리 제어기로서, 또는 없이) 메모리 버스, 그리고 주변 버스(예를 들면 PCI, PCIe, AGP, LPC 등)에 더 상호연결될 수 있는 몇몇 유형의 버스 구조 중의 임의의 버스 구조일 수 있다.

- [0092] 컴퓨터(802)는 저장 서브시스템(들)(814), 그리고 저장 서브시스템(들)(814)을 시스템 버스(808)와 다른 원하는 컴퓨터 구성요소로 인터페이스하기 위한 저장 인터페이스(들)(816)을 더 포함한다. 저장 서브시스템(들)(814)은 예를 들어 HDD(hard disk drive), FDD(magnetic floppy disk drive), 그리고/또는 광 디스크 저장 드라이브(예를 들면 CD-ROM 드라이브, DVD 드라이브)중의 하나 이상을 포함할 수 있다. 저장 인터페이스(들)(816)는 예를 들면 EIDE, ATA, SATA 및 IEEE 1394와 같은 인터페이스 기술을 포함할 수 있다.
- [0093] 운영체제(820), 하나 이상의 애플리케이션 프로그램(822), 다른 프로그램 모듈(824) 및 프로그램 데이터(826)를 포함하는 하나 이상의 프로그램과 데이터는, 메모리 서브시스템(806), 분리식 메모리 서브시스템(818)(예를 들면 플래시 드라이브 폼팩터 기술), 및/또는 저장 서브시스템(들)(814)(예를 들면 광, 자기, 고체상태)에 저장될 수 있다.
- [0094] 하나 이상의 애플리케이션 프로그램(822), 다른 프로그램 모듈(824) 및 프로그램 데이터(826)는 예를 들어 도 1의 시스템(100) 및 구성요소, 도 2의 시스템(200) 및 구성요소, 그래프(300, 400)에 의해 표현된 관계, 그리고 도 5 내지 7의 흐름도에 의해 표현된 방법을 포함할 수 있다.
- [0095] 통상, 프로그램은 특정 작업을 수행하거나, 또는 특정한 추상 데이터 유형을 구현하는 루틴, 방법, 데이터 구조, 다른 소프트웨어 구성요소 등을 포함한다. 운영체제(820), 애플리케이션(822), 모듈(824), 및/또는 데이터(826)의 모두 또는 일부는 예를 들어 휘발성 메모리(810)와 같은 메모리에 캐싱될 수 있다. 개시된 구조는 (예를 들어 가상 머신 같은) 상업적으로 이용가능한 다양한 운영체제 또는 운영체제의 조합으로써 구현될 수 있다.
- [0096] 저장 서브시스템(들)(814)과 메모리 서브시스템(806, 818)은 데이터, 데이터 구조, 컴퓨터 실행가능 인스트럭션 등의 휘발성 및 비휘발성 저장을 위한 컴퓨터 판독가능 매체로서의 기능을 한다. 컴퓨터 판독가능 매체는 컴퓨터(802)에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리식 및 비분리식 매체를 포함한다. 컴퓨터(802)의 경우, 매체는 임의의 적당한 디지털 포맷으로 데이터 저장을 수용한다. 당업자는 개시된 구조의 신규 방법을 수행하기 위한 컴퓨터 실행가능 인스트럭션을 저장하기 위해 zip 드라이브(zip drive), 자기 테이프, 플래시 메모리 카드, 카트리지 등과 같은 다른 유형의 컴퓨터 판독가능 매체를 사용할 수 있다는 것을 알아야 한다.
- [0097] 사용자는 키보드 및 마우스와 같은 외부 사용자 입력 장치(828)를 사용하여 컴퓨터(802), 프로그램 및 데이터와 상호작용할 수 있다. 다른 외부 사용자 입력 장치(828)는 마이크로폰, IR(infrared) 원격 제어부, 조이스틱, 게임패드, 카메라 인식 시스템, 스틸러스 펜, 터치 스크린, 제스처 시스템(예를 들어 눈 움직임, 머리 움직임 등) 및/또는 기타등등을 포함할 수 있다. 사용자는 터치패드, 마이크로폰, 키보드 등과 같은 온보드 사용자 입력장치(830)를 사용하여 컴퓨터(802), 프로그램 및 데이터와 상호작용할 수 있고, 여기서 컴퓨터(802)는 예를 들어 휴대용 컴퓨터이다. 이들 및 다른 입력 장치는 시스템 버스(808)를 통해 I/O(input/output) 장치 인터페이스(들)(832)를 거쳐 처리 유닛(들)(804)으로 연결되지만, 병렬 포트, IEEE 1394 직렬포트, 게임포트, USB 포트, IR 인터페이스 등과 같은 다른 인터페이스에 의해 연결될 수 있다. 또한 I/O 장치 인터페이스(들)(832)는 사운드 카드 및/또는 온보드 오디오 처리 능력과 더불어 프린터, 오디오 장치, 카메라 장치 등과 같은 출력 주변기기(834)의 사용을 용이하게 한다.
- [0098] 하나 이상의 그래픽 인터페이스(들)(836)(또한 그래픽 처리 유닛(GPU)으로 언급)은 컴퓨터(802)와 외부 디스플레이(들)(838)(예를 들면 LCD, 플라즈마) 및/또는 (예를 들어 휴대용 컴퓨터를 위한) 온보드 디스플레이(840) 사이에, 그래픽과 비디오 신호를 제공한다 (그래픽 인터페이스(들)(836)는 컴퓨터 시스템 보드의 일부로 생산될 수도 있다).
- [0099] 컴퓨터(802)는 하나 이상의 네트워크 및/또는 다른 컴퓨터로 유선/무선 통신 서브시스템(842)을 통해 논리적 연결을 사용하여 네트워크 환경(예를 들면 IP)에서 동작할 수 있다. 다른 컴퓨터는 워크스테이션, 서버, 라우터, 퍼스널 컴퓨터, 마이크로프로세서기반 엔터테인먼트 기기, 피어 장치(peer device) 또는 다른 공통 네트워크 노드를 포함할 수 있고, 전형적으로 컴퓨터(802)에 대하여 기술한 모든 또는 다수의 요소를 포함한다. 논리적 연결부는 LAN(local area network), WAN(wide area network), 핫스팟(hotspot) 등으로의 유선/무선 연결을 포함할 수 있다. LAN과 WAN 네트워킹 환경은 사무실과 회사들에 흔하며, 인트라넷과 같은 전사적(enterprise-wide) 컴퓨터 네트워크를 용이하게 하고, 이들 모두는 인터넷과 같은 글로벌 통신망에 연결될 수 있다.

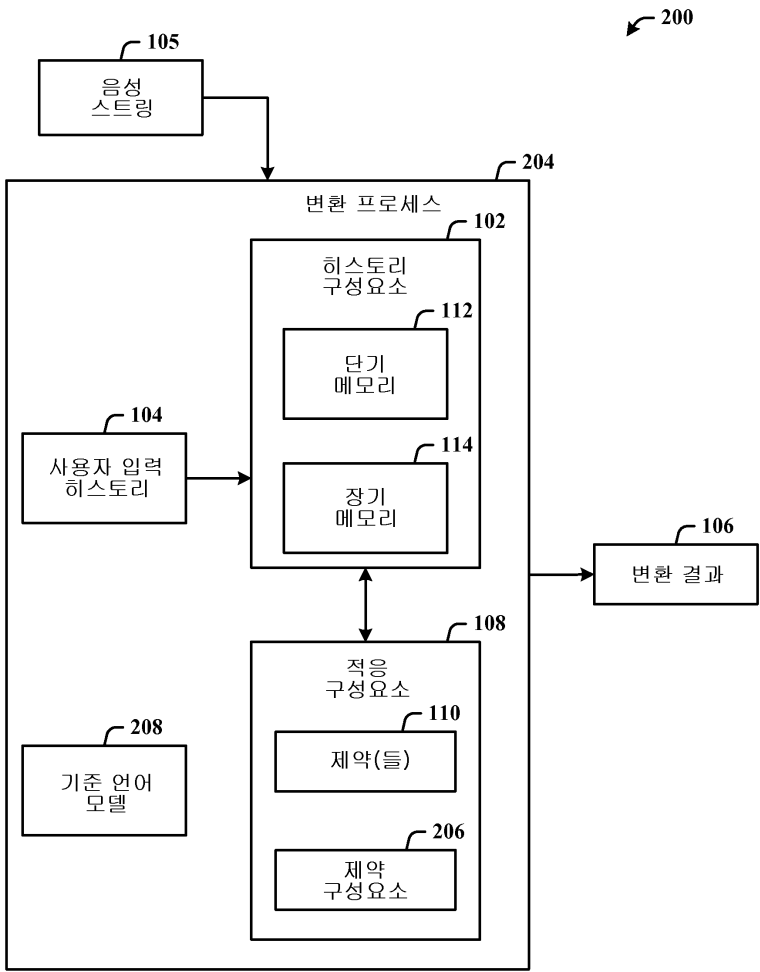
- [0100] 컴퓨터(802)는 네트워킹 환경에서 사용될 때에 유선/무선 네트워크, 유선/무선 프린터, 유선/무선 입력장치(844) 등과 통신하기 위해 유선/무선 통신 서브시스템(842)(예를 들면 네트워크 인터페이스 어댑터, 온보드 트랜스미터 서브시스템 등)을 통해 네트워크로 연결된다. 컴퓨터(802)는 모뎀을 포함하거나, 또는 네트워크를 통해 통신을 설정하기 위한 다른 수단을 가질 수 있다. 네트워크 환경에서, 컴퓨터(802)에 관련된 프로그램과 데이터는 분산 시스템과 관련있으므로 원격 메모리/저장 장치에 저장될 수 있다. 도시된 네트워크 연결은 예시적인 것이며, 컴퓨터들 간에 통신 회선을 설정하기 위한 다른 수단을 사용할 수 있다는 것을 알 것이다.
- [0101] 컴퓨터(802)는 예를 들어 프린터, 스캐너, 데스크탑 및/또는 휴대용 컴퓨터, PDA(personal digital assistant), 통신위성, 무선으로 검출가능한 태그(예를 들어 키오스크, 뉴스탠드, 레스트룸)와 관련된 임의 설비 기구 또는 위치, 그리고 전화와 함께, 무선 통신에 동작가능하게 배치된 무선 장치와 같이 IEEE 802.xx 계열 표준과 같은 무선 기술을 사용하여 유선/무선 장치 또는 항목과 통신하도록 동작할 수 있다(예를 들어 IEEE 802.11 OTA(over-the-air) 변조 기법). 이것은 핫스팟, WiMax 및 블루투스(Bluetooth™) 무선 기술을 위해 적어도 Wi-Fi(Wireless Fidelity)를 포함한다. 따라서 통신은 종래 네트워크처럼 사전정의된 구조이거나, 또는 적어도 두 장치들 간에 애드혹(ad hoc) 통신일 수 있다. Wi-Fi 네트워크는 안전하고 신뢰할만하며 신속한 무선 연결을 제공하기 위하여 IEEE 802.xx(a, b, g 등)로 불리는 무선 기술을 사용한다. Wi-Fi 네트워크는 컴퓨터를 서로, 인터넷으로, 및 (IEEE 802.3-관련 매체와 기능을 사용하는) 유선 네트워크로 연결하는데 사용될 수 있다.
- [0102] 전술한 내용은 개시된 구조의 예를 포함한다. 물론, 생각할 수 있는 모든 구성요소의 조합, 및/또는 방법론을 기술하는 것은 불가능하지만, 당업자는 다수의 다른 조합과 치환이 가능하다는 것을 알 수 있다. 따라서 신규 구조는 첨부된 청구범위의 사상 및 범주내에 있는 모든 이러한 변경, 변형 및 변동을 포함하려 한다. 또한 용어 "포함한다(includes)"가 상세한 설명 또는 청구범위에 사용된 경우, 이러한 용어는 "포함하는(comprising)"이 청구범위에서 접속어로서 사용될 때에 해석되는 바와 같이 용어 "포함하는(comprising)"과 유사한 방식으로 포괄적인 것으로 사용하고자 한다.

도면

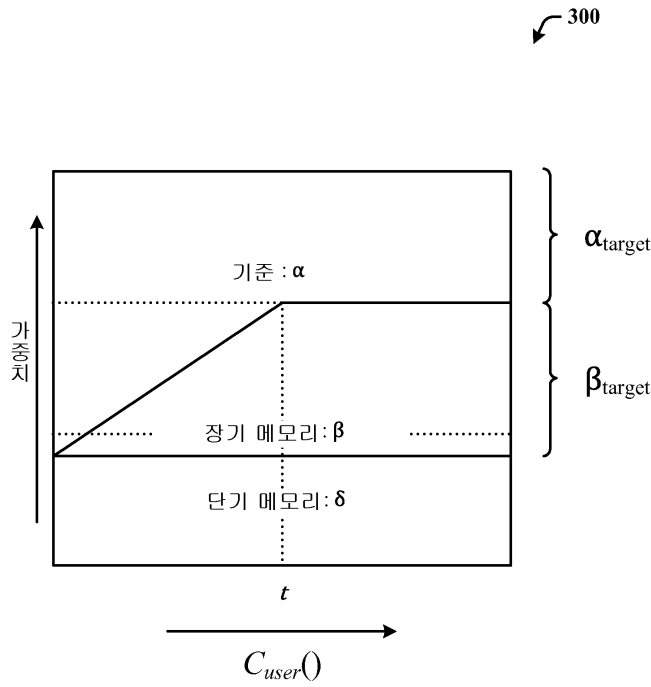
도면1



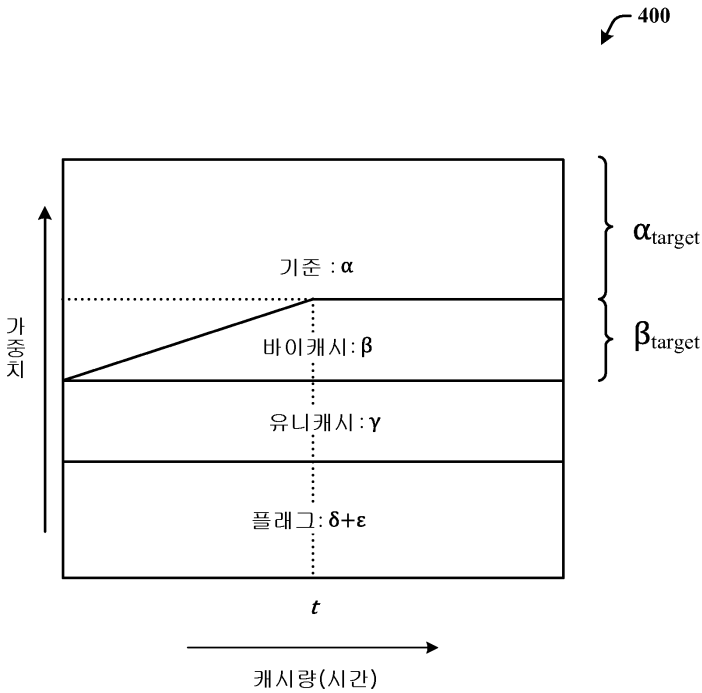
도면2



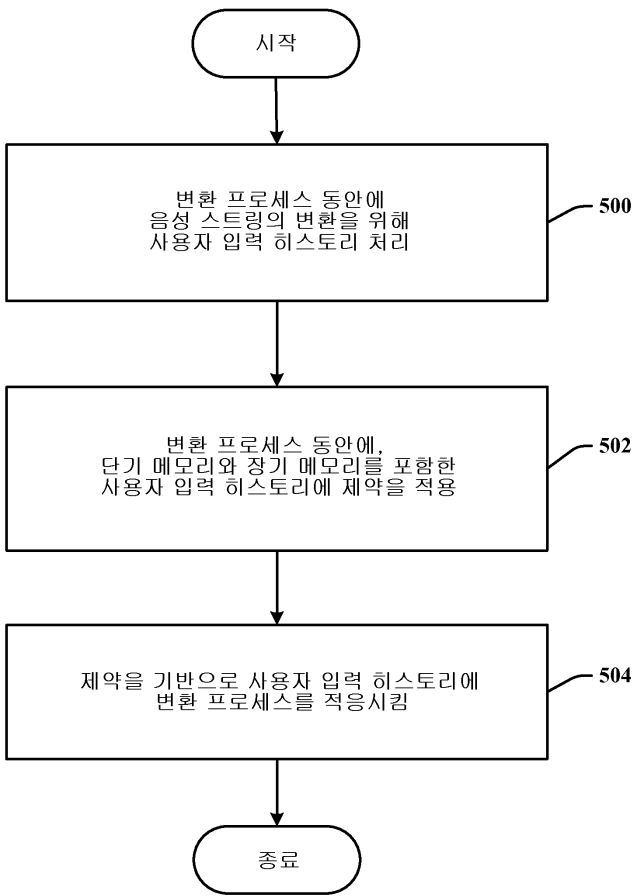
도면3



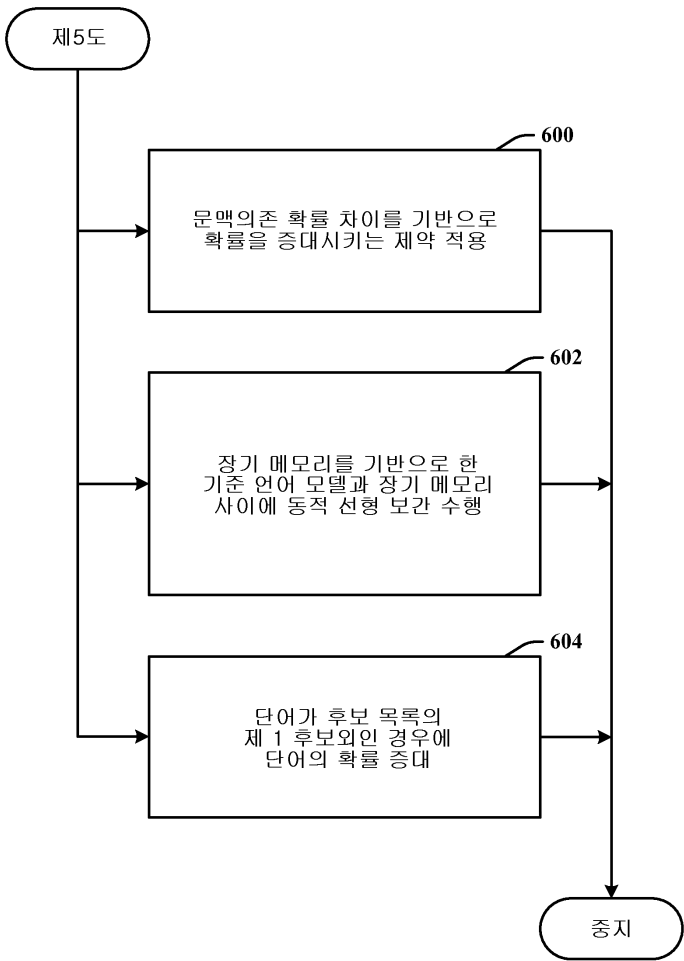
도면4



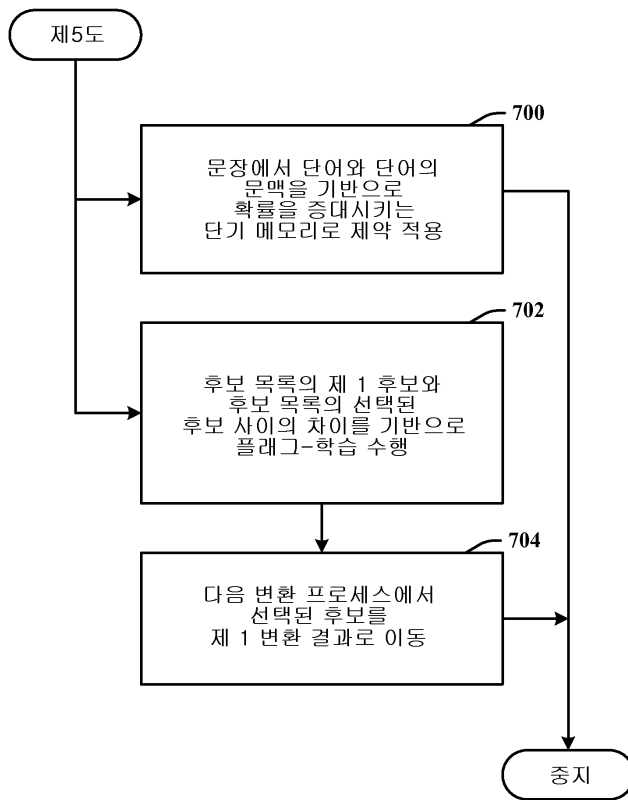
도면5



도면6



도면7



도면8

