(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년10월04일
(11) 등록번호 10-2449831
(24) 등록일자 2022년09월27일

(51) 국제특허분류(Int. Cl.)
G06F 40/20 (2020.01)
(52) CPC특허분류
G06F 40/274 (2020.01)
(21) 출원번호 10-2018-0004433
(22) 출원일자 2018년01월12일
심사청구일자 2020년11월25일
(65) 공개번호 10-2019-0086199
(43) 공개일자 2019년07월22일
(56) 선행기술조사문헌
US20150347383 A1*
US09594741 B1*
KR1020130143080 A
KR101697875 B1
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
하마드 빈 칼리파 유니버시티
카타르, 도하, 피.오.박스 34110
난양 테크놀러지컬 유니버시티
싱가포르 639798, 난양 애비뉴 50
(72) 발명자
신혜진
서울특별시 강서구 공항대로 382 우장산롯데캐슬 아파트
샤오, 샤오쿠이
싱가포르, 639798, 50 난양 에비뉴
(뒷면에 계속)
(74) 대리인
이건주, 김정훈

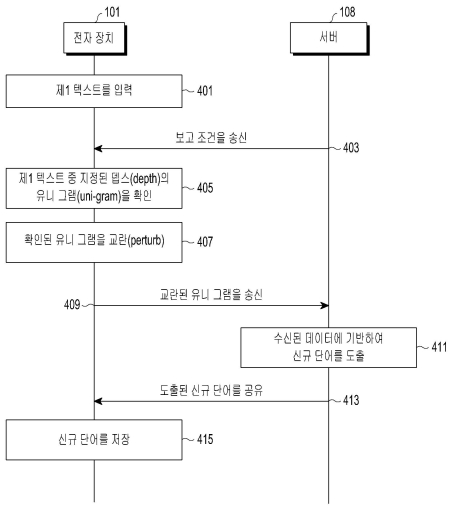
전체 청구항 수 : 총 10 항 심사관 : 김경완

(54) 발명의 명칭 **신규 텍스트에 대한 정보를 제공하는 전자 장치, 신규 텍스트를 확인하는 서버 및 그 동작 방법**

(57) 요 약

다양한 실시예에 따라서, 전자 장치는, 입력 장치, 표시 장치, 통신 회로, 프로세서, 및 메모리를 포함하고, 상기 메모리는, 실행시에 상기 프로세서로 하여금, 상기 입력 장치를 통하여 제 1 텍스트를 입력받으며, 상기 표시 장치를 통하여 상기 제 1 텍스트를 표시하고, 상기 통신 회로를 통하여, 서버로부터 제 1 보고 조건을 수신하고, 상기 제 1 텍스트 중 상기 제 1 보고 조건에 대응하는 뎁스의 제 1 유니 그램(uni gram)을 확인하고, 상기 제 1 유니 그램을 교란(perturb)하여, 제 1 교란된 유니 그램을 획득하고, 상기 통신 회로를 통하여, 상기 제 1 교란된 유니 그램을 상기 서버로 송신하고, 상기 전자 장치로부터 수신한 제 1 교란된 유니 그램 및 다른 전자 장치로부터 수신한 데이터에 기반하여 형성된 트리(trie)로부터 상기 서버에 의하여 도출된 신규 단어를, 상기 통신 회로를 통하여 수신하고, 상기 신규 단어를 상기 메모리에 저장하도록 하는 인스트럭션들을 저장할 수 있다.

대 표 도 - 도4



(72) 발명자

신준범

경기도 수원시 영통구 봉영로 1526 살구골7단지아파트 717동 104호

양, 인

카타르, 도하, 에듀케이션 시티, 라스 빌딩, 룸 에이306에이치

명 세 서

청구범위

청구항 1

전자 장치에 있어서,

입력 장치;

표시 장치;

통신 회로;

프로세서; 및

메모리

를 포함하고,

상기 메모리는, 실행시에 상기 프로세서로 하여금,

상기 입력 장치를 통하여 제 1 텍스트를 입력받으며, 상기 표시 장치를 통하여 상기 제 1 텍스트를 표시하고,

상기 통신 회로를 통하여, 서버로부터 제 1 보고 조건을 수신하고,

상기 제 1 텍스트 중 상기 제 1 보고 조건에 대응하는 뎁스의 제 1 유니 그램(uni gram)을 확인하고,

상기 제 1 유니 그램을 교란(perturb)하여, 제 1 교란된 유니 그램을 획득하고,

상기 통신 회로를 통하여, 상기 제 1 교란된 유니 그램을 상기 서버로 송신하고,

상기 전자 장치로부터 수신한 제 1 교란된 유니 그램 및 다른 전자 장치로부터 수신한 데이터에 기반하여 형성된 트리(trie)로부터 상기 서버에 의하여 도출된 신규 단어를, 상기 통신 회로를 통하여 수신하고,

상기 신규 단어를 상기 메모리에 저장하도록 하는 인스트럭션들을 저장하는 전자 장치.

청구항 2

제 1 항에 있어서,

상기 인스트럭션들은, 상기 프로세서가,

상기 통신 회로를 통하여, 상기 서버로부터 제 2 보고 조건을 수신하고,

상기 전자 장치가 상기 제 2 보고 조건에 대응하는 것으로 확인되면, 상기 제 1 텍스트 중 상기 제 2 보고 조건에 대응하는 뎁스의 제 2 유니 그램을 확인하고,

상기 제 2 유니 그램을 교란하여, 제 2 교란된 유니 그램을 획득하고,

상기 통신 회로를 통하여, 상기 제 2 교란된 유니 그램을 상기 서버로 송신하도록 하는 전자 장치.

청구항 3

제 2 항에 있어서,

상기 제 2 보고 조건은,

상기 전자 장치가, 상기 트리 내에서 상기 제 2 보고 조건에 대응하는 노드의 부모 노드에 대하여 데이터를 제공한지 여부에 대한 것인 전자 장치.

청구항 4

제 3 항에 있어서,

상기 인스트럭션들은, 상기 프로세서가,

상기 통신 회로를 통하여 송신한 상기 제 1 교란된 유니 그램이, 상기 트리 내에서 상기 제 2 보고 조건에 대응하는 노드의 상기 부모 노드에 대한 것이 아닌 것으로 확인되면, 상기 제 2 교란된 유니 그램을 상기 서버로 송신하도록 하는 전자 장치.

청구항 5

제 2 항에 있어서,

상기 인스트럭션들은, 상기 프로세서가,

상기 제 1 교란된 유니 그램을 획득하는 과정에서, 제 1 크기의 제 1 프라이버시 버짓(privacy budget)을 소요하고,

상기 제 2 교란된 유니 그램을 획득하는 과정에서, 제 1 크기의 제 2 프라이버시 버짓(privacy budget)을 소요하도록 하고,

상기 제 1 프라이버시 버짓은, 상기 제 2 프라이버시 버짓으로부터 독립적인 전자 장치.

청구항 6

제 1 항에 있어서,

상기 인스트럭션들은, 상기 프로세서가,

상기 신규 단어를 상기 메모리에 저장한 이후에, 상기 입력 장치를 통하여 텍스트를 입력받고,

상기 입력받은 텍스트의 적어도 일부가, 상기 신규 단어의 적어도 일부와 동일한 것이 확인되면, 상기 신규 단어를 후보 단어로서 상기 표시 장치를 통하여 표시하도록 하는 전자 장치.

청구항 7

전자 장치의 동작 방법에 있어서,

제 1 텍스트를 입력받으며, 상기 제 1 텍스트를 표시하는 동작;

서버로부터 제 1 보고 조건을 수신하는 동작;

상기 제 1 텍스트 중 상기 제 1 보고 조건에 대응하는 뎁스의 제 1 유니 그램(uni gram)을 확인하는 동작;

상기 제 1 유니 그램을 교란(perturb)하여, 제 1 교란된 유니 그램을 획득하는 동작;

상기 제 1 교란된 유니 그램을 상기 서버로 송신하는 동작;

상기 전자 장치로부터 수신한 제 1 교란된 유니 그램 및 다른 전자 장치로부터 수신한 데이터에 기반하여 형성된 트리(trie)로부터 상기 서버에 의하여 도출된 신규 단어를 수신하는 동작; 및

상기 신규 단어를 상기 전자 장치의 메모리에 저장하는 동작을 포함하는 전자 장치의 동작 방법의 동작 방법.

청구항 8

제 7 항에 있어서,

상기 서버로부터 제 2 보고 조건을 수신하는 동작;

상기 전자 장치가 상기 제 2 보고 조건에 대응하는 것으로 확인되면, 상기 제 1 텍스트 중 상기 제 2 보고 조건에 대응하는 뎁스의 제 2 유니 그램을 확인하는 동작;

상기 제 2 유니 그램을 교란하여, 제 2 교란된 유니 그램을 획득하는 동작; 및

상기 제 2 교란된 유니 그램을 상기 서버로 송신하는 동작

을 더 포함하는 전자 장치의 동작 방법의 동작 방법.

청구항 9

◈청구항 9은(는) 설정등록료 납부시 포기되었습니다.◈

제 8 항에 있어서,

상기 제 2 보고 조건은,

상기 전자 장치가, 상기 트리 내에서 상기 제 2 보고 조건에 대응하는 노드의 부모 노드에 대하여 데이터를 제공한지 여부에 대한 것인 전자 장치의 동작 방법.

청구항 10

◈청구항 10은(는) 설정등록료 납부시 포기되었습니다.◈

제 9 항에 있어서,

상기 전자 장치가 상기 제 2 보고 조건에 대응하는 것으로 확인되면, 상기 제 1 텍스트 중 상기 제 2 보고 조건에 대응하는 뎁스의 제 2 유니 그램을 확인하는 동작 및 상기 제 2 교란된 유니 그램을 상기 서버로 송신하는 동작은, 송신한 상기 제 1 교란된 유니 그램이, 상기 트리 내에서 상기 제 2 보고 조건에 대응하는 노드의 상기 부모 노드에 대한 것이 아닌 것으로 확인되면, 상기 제 2 교란된 유니 그램을 상기 서버로 송신하도록 하는 전자 장치의 동작 방법.

청구항 11

◈청구항 11은(는) 설정등록료 납부시 포기되었습니다.◈

제 8 항에 있어서,

상기 제 1 교란된 유니 그램을 획득하는 동작에서, 제 1 크기의 제 1 프라이버시 버짓(privacy budget)을 소요하고,

상기 제 2 교란된 유니 그램을 획득하는 동작에서, 제 1 크기의 제 2 프라이버시 버짓(privacy budget)을 소요하도록 하고,

상기 제 1 프라이버시 버짓은, 상기 제 2 프라이버시 버짓으로부터 독립적인 전자 장치의 동작 방법.

청구항 12

◈청구항 12은(는) 설정등록료 납부시 포기되었습니다.◈

제 7 항에 있어서,

상기 신규 단어를 상기 메모리에 저장한 이후에, 텍스트를 입력받는 동작; 및

상기 입력받은 텍스트의 적어도 일부가, 상기 신규 단어의 적어도 일부와 동일한 것이 확인되면, 상기 신규 단어를 후보 단어로서 표시하는 동작

을 더 포함하는 전자 장치의 동작 방법.

청구항 13

서버에 있어서,

통신 회로;

프로세서; 및

메모리

를 포함하고,

상기 메모리는, 실행 시에 상기 프로세서가,

상기 통신 회로를 통하여, 복수 개의 전자 장치들 중 제1 전자 장치로, 제1 보고 조건을 송신하고,

상기 통신 회로를 통하여, 상기 제1 전자 장치로부터, 상기 제1 전자 장치의 입력 장치를 통하여 입력된 텍스트

중 상기 제1 보고 조건에 대응하는 뎁스의 유니 그램(uni gram)이 교란된(perturb) 유니 그램을 수신하고,

상기 제1 전자 장치로부터 수신한 상기 교란된 유니 그램과 상기 복수 개의 전자 장치들 중 제2 전자 장치로부터 수신한 데이터에 기반하여, 트리를 형성하고,

상기 형성된 트리에 기반하여, 신규 단어를 도출하고,

상기 신규 단어를 상기 통신 회로를 통하여 상기 제1 전자 장치로 송신하도록 하는 인스트럭션들을 저장하는 서버.

청구항 14

◈청구항 14은(는) 설정등록료 납부시 포기되었습니다.◈

제 13 항에 있어서,

상기 인스트럭션들은, 상기 프로세서가,

상기 트리를 구성하는 복수 개의 노드들 중 제 1 노드에 대한 추가 데이터를 송신할 사용자에 대한 제2 보고 조건을, 상기 복수 개의 전자 장치들 중 적어도 하나의 전자 장치에 송신하도록 하는 서버.

청구항 15

◈청구항 15은(는) 설정등록료 납부시 포기되었습니다.◈

제 14 항에 있어서,

상기 제2 보고 조건은, 상기 제 1 노드의 부모 노드에의 부모 노드에 대하여 데이터를 제공하지 않은 상기 적어도 하나의 전자 장치가 데이터를 제공하는 것인 서버.

청구항 16

◈청구항 16은(는) 설정등록료 납부시 포기되었습니다.◈

제 14 항에 있어서,

상기 인스트럭션들은, 상기 프로세서가,

상기 제2 보고 조건을 만족하는 상기 적어도 하나의 전자 장치로부터 상기 추가 데이터를, 상기 통신 회로를 통하여 수신하고,

상기 제 1 노드에 대하여 기존에 수신한 데이터 및 상기 추가 데이터에 기반하여, 상기 제 1 노드에 대한 서포트(support) 값을 확인하고,

상기 서포트 값에 기반하여 상기 트리를 형성하도록 하는 서버.

청구항 17

서버의 동작 방법에 있어서,

복수 개의 전자 장치들 중 제1 전자 장치로, 제1 보고 조건을 송신하는 동작;

상기 제1 전자 장치로부터, 상기 제1 전자 장치의 입력 장치를 통해 입력된 텍스트 중 상기 제1 보고 조건에 대응하는 뎁스의 유니 그램(uni gram)이 교란된(perturb) 유니 그램을 수신하는 동작;

상기 제1 전자 장치로부터 수신한 상기 교란된 유니 그램과 상기 복수 개의 전자 장치들 중 제2 전자 장치로부터 수신한 데이터에 기반하여, 트리를 형성하는 동작;

상기 형성된 트리에 기반하여, 신규 단어를 도출하는 동작; 및

상기 신규 단어를 상기 제1 전자 장치로 송신하는 동작

을 포함하는 서버의 동작 방법.

청구항 18

◈청구항 18은(는) 설정등록료 납부시 포기되었습니다.◈

제 17 항에 있어서,

상기 트리를 구성하는 복수 개의 노드들 중 제 1 노드에 대한 추가 데이터를 송신할 사용자에 대한 제2 보고 조건을, 상기 복수 개의 전자 장치들 중 적어도 하나의 전자 장치에 송신하는 동작

을 더 포함하는 서버의 동작 방법.

청구항 19

◈청구항 19은(는) 설정등록료 납부시 포기되었습니다.◈

제 18 항에 있어서,

상기 제2 보고 조건은, 상기 제 1 노드의 부모 노드에의 부모 노드에 대하여 데이터를 제공하지 않은 상기 적어도 하나의 전자 장치가 데이터를 제공하는 것인 서버의 동작 방법.

청구항 20

◈청구항 20은(는) 설정등록료 납부시 포기되었습니다.◈

제 18 항에 있어서,

상기 제2 보고 조건을 만족하는 상기 적어도 하나의 전자 장치로부터 상기 추가 데이터를 수신하는 동작

을 더 포함하고,

상기 신규 단어를 도출하는 동작은,

상기 제 1 노드에 대하여 기존에 수신한 데이터 및 상기 추가 데이터에 기반하여, 상기 제 1 노드에 대한 서포트(support) 값을 확인하는 동작; 및

상기 서포트 값에 기반하여 상기 트리를 형성하도록 하는 동작

을 포함하는 서버의 동작 방법.

발명의 설명

기 술 분 야

[0001] 다양한 실시예는, 신규 텍스트에 대한 정보를 제공하는 전자 장치, 신규 텍스트를 확인하는 서버 및 그 동작 방법에 관한 것이다.

배 경 기 술

[0002] 대부분의 스마트 폰은 터치스크린 상에 표시되는 가상 키보드를 통하여 텍스트를 입력받는다. 스마트 폰은 휴대를 위하여 상대적으로 소형으로 제작되며, 이에 따라 터치스크린의 크기 또한 제한적이며, 터치스크린에 표시되는 가상 키보드를 구성하는 텍스트 입력을 위한 오브젝트들 또한 소형으로 표시된다. 사용자는 원하는 텍스트를 어려움을 가질 수 있다.

[0003] 전자 장치는, 사용자가 입력한 텍스트의 적어도 일부에 기반하여 예상되는 단어를 추천하여 표시하는 기능을 제공할 수 있다. 예를 들어, 사용자가 "elepha"을 입력한 경우에는, 전자 장치는 해당 텍스트에 대응하는 후보 단어인 "elephant"를 터치스크린 상에 표시할 수 있다. 만약, 사용자가 후보 단어인 "elephant"를 지정한 경우에는, 전자 장치는 지정된 단어를 자동 입력할 수 있다. 이에 따라, 사용자는 상대적으로 빠른 시간 내에 원하는 단어를 입력할 수 있다.

발명의 내용

해결하려는 과제

[0004] 전자 장치는, 사전에 등재된 단어 및 사용자가 자주 입력한 신조어와 같은 신규 텍스트를 후보 단어로 표시할 수 있다. 특히, 신조어와 같은 신규 텍스트는, 전자 장치 내부에서 저장될 수도 있으나, 외부 서버에 의하여

신규 텍스트가 확인될 수도 있다. 외부 서버는, 복수 개의 전자 장치들에 입력된 정보에 대하여 수신할 수 있으며, 이를 분석하여 신규 텍스트를 확인할 수 있다. 외부 서버는, 신규 텍스트를 다른 전자 장치들과 공유할 수 있으며, 이에 따라 전자 장치는 신규 텍스트를 후보 단어로서 제공할 수 있다.

[0005] 전자 장치에서 입력받은 정보를 그대로 외부 서버로 송신하는 경우에는, 전자 장치의 사용자의 프라이버시가 침해될 가능성이 있다. 이에 따라, 로컬 디퍼런셜 프라이버시(local differential privacy) 기법, 예를 들어, RAPPOR 기법 등이 도입되고 있다. 해당 기법에서는, 수신된 텍스트를 n-그램으로 분할하여, 교란을 시킨후, 서버로 전송하는 방식이 이용된다. 하지만, 해당 기법들은 타겟 단어가 상대적으로 긴 길이를 가지는 경우에, 정확도가 현저하게 저하되는 문제를 가진다.

[0006] 다양한 실시예에 따른 전자 장치 및 그 동작 방법은, 신규 스트링 중 지정된 뎁스에서의 유니 그램에 대한 랜덤화된 값을 서버에 제공하는 방식을 이용함으로써, 사용자의 프라이버시가 보장될 수 있다. 다양한 실시예에 따른 서버 및 그 동작 방법은, 수신한 데이터에 기반하여 트리를 형성하고 형성된 트리에 기반하여 신규 텍스트를 확인할 수 있다.

과제의 해결 수단

[0007] 다양한 실시예에 따라서, 전자 장치는, 입력 장치, 표시 장치, 통신 회로, 프로세서, 및 메모리를 포함하고, 상기 메모리는, 실행시에 상기 프로세서로 하여금, 상기 입력 장치를 통하여 제 1 텍스트를 입력받으며, 상기 표시 장치를 통하여 상기 제 1 텍스트를 표시하고, 상기 통신 회로를 통하여, 서버로부터 제 1 보고 조건을 수신하고, 상기 제 1 텍스트 중 상기 제 1 보고 조건에 대응하는 뎁스의 제 1 유니 그램(unigram)을 확인하고, 상기 제 1 유니 그램을 교란(perturb)하여, 제 1 교란된 유니 그램을 획득하고, 상기 통신 회로를 통하여, 상기 제 1 교란된 유니 그램을 상기 서버로 송신하고, 상기 전자 장치로부터 수신한 제 1 교란된 유니 그램 및 다른 전자 장치로부터 수신한 데이터에 기반하여 형성된 트리(trie)로부터 상기 서버에 의하여 도출된 신규 단어를, 상기 통신 회로를 통하여 수신하고, 상기 신규 단어를 상기 메모리에 저장하도록 하는 인스트럭션들을 저장할 수 있다.

[0008] 다양한 실시예에 따라서, 전자 장치의 동작 방법은, 제 1 텍스트를 입력받으며, 상기 제 1 텍스트를 표시하는 동작, 서버로부터 제 1 보고 조건을 수신하는 동작, 상기 제 1 텍스트 중 상기 제 1 보고 조건에 대응하는 뎁스의 제 1 유니 그램(unigram)을 확인하는 동작, 상기 제 1 유니 그램을 교란(perturb)하여, 제 1 교란된 유니 그램을 획득하는 동작, 상기 제 1 교란된 유니 그램을 상기 서버로 송신하는 동작, 상기 전자 장치로부터 수신한 제 1 교란된 유니 그램 및 다른 전자 장치로부터 수신한 데이터에 기반하여 형성된 트리(trie)로부터 상기 서버에 의하여 도출된 신규 단어를 수신하는 동작, 및 상기 신규 단어를 상기 전자 장치의 메모리에 저장하는 동작을 포함할 수 있다.

[0009] 다양한 실시예에 따라서, 서버에 있어서, 통신 회로, 프로세서, 및 메모리를 포함하고, 상기 메모리는, 실행 시에 상기 프로세서가, 상기 통신 회로를 통하여, 복수 개의 전자 장치들 각각으로부터 복수 개의 교란된 유니 그램들을 각각 수신하고, 상기 복수 개의 교란된 유니 그램들 각각의 뎁스에 기반하여, 각각이 상기 복수 개의 교란된 유니 그램들 각각에 대응하는 복수 개의 노드들로 구성되는 트리를 형성하고, 상기 형성된 트리에 기반하여, 신규 단어를 도출하고, 상기 신규 단어를 상기 통신 회로를 통하여 상기 복수 개의 전자 장치들 각각으로 송신하도록 하는 인스트럭션들을 저장할 수 있다.

[0010] 다양한 실시예에 따라서, 서버의 동작 방법은, 복수 개의 전자 장치들 각각으로부터 복수 개의 교란된 유니 그램들을 각각 수신하는 동작, 상기 복수 개의 교란된 유니 그램들 각각의 뎁스에 기반하여, 각각이 상기 복수 개의 교란된 유니 그램들 각각에 대응하는 복수 개의 노드들로 구성되는 트리를 형성하는 동작, 상기 형성된 트리에 기반하여, 신규 단어를 도출하는 동작, 및 상기 신규 단어를 상기 복수 개의 전자 장치들 각각으로 송신하는 동작을 포함할 수 있다.

발명의 효과

[0011] 다양한 실시예에 따라서, 신규 스트링 중 지정된 뎁스의 유니 그램만을 서버에 제공하는 방식을 이용하는 전자 장치 및 그 동작 방법이 제공될 수 있어, 사용자의 프라이버시가 보장될 수 있다. 다양한 실시예에 따라서, 수신한 데이터에 기반하여 트리를 형성하고 형성된 트리에 기반하여 신규 텍스트를 확인하는 서버 및 그 동작 방법이 제공될 수 있다.

도면의 간단한 설명

[0012] 도 1은 다양한 실시예에 따른 복수 개의 전자 장치 및 서버의 도면을 도시한다.

도 2는 다양한 실시예에 따른 전자 장치, 외부 전자 장치 및 서버의 도면이다.

도 3은 다양한 실시예에 따른 트리(trie)를 도시한다.

도 4는 다양한 실시예에 따른 전자 장치 및 서버의 동작 방법을 설명하기 위한 흐름도를 도시한다.

도 5는 다양한 실시예에 따른 전자 장치를 도시한다.

도 6은 다양한 실시예에 따른 전자 장치 및 서버의 동작 방법을 설명하기 위한 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0013] 도 1은 다양한 실시예에 따른 복수 개의 전자 장치 및 서버의 도면을 도시한다.

[0014] 도 1에 도시된 바와 같이, 서버(108)는, 복수 개의 전자 장치들(101,102,103)들과 데이터를 송수신할 수 있다. 도 2를 참조하여 더욱 상세하게 설명할 것으로, 서버(108)는, 복수 개의 전자 장치(101,102,103)들과 무선 통신을 통하여 데이터를 송수신할 수 있다. 예를 들어, 전자 장치들(101,102,103)은 입력 장치를 통하여 획득한 텍스트 중 적어도 일부에 대한 데이터를 서버(108)로 송신할 수 있다. 전자 장치들(101,102,103)은 입력 장치를 통하여 획득한 텍스트 중 일부를 교란(perturb)하여 서버(108)로 송신할 수 있으며, 이에 따라 전자 장치들(101,102,103)로 입력되는 내용이 보호될 수 있다. 다양한 실시예에 따라서, 전자 장치들(101,102,103)은 서버(108)로부터 보고 조건을 수신할 수 있다. 보고 조건에는, 서버(108) 가 형성한 트리(trie) 내에 포함되는 노드에 대응하는 뎁스에 대한 정보가 포함될 수 있다. 전자 장치들(101,102,103) 각각은, 입력받은 텍스트 중 수신한 보고 조건에 대응하는 유니 그램을 확인할 수 있다. 전자 장치들(101,102,103) 각각은 확인한 유니 그램을 교란(perturb)하여, 교란된 유니 그램을 획득하고, 교란된 유니 그램을 서버(108)로 송신할 수 있다. 서버(108)는, 전자 장치들(101,102,103)로부터 수신한 교란된 유니 그램에 기반하여 형성된 트리(trie)로부터 신규 단어를 도출할 수 있다. 서버(108)는, 도출된 신규 단어를 전자 장치들(101,102,103) 각각과 공유할 수 있으며, 상술한 과정에 대하여서는 더욱 상세하게 후술하도록 한다.

[0015] 도 2는 다양한 실시예에 따른 전자 장치, 외부 전자 장치 및 서버의 도면이다.

[0016] 다양한 실시예에 따라서, 전자 장치(101)는 서버(108)와 통신을 수행할 수 있다. 전자 장치(101)는, 예를 들어 원거리 무선 통신 네트워크를 통하여 서버(108)와 통신할 수 있다.

[0017] 본 문서에서 전자 장치(101), 또는 서버(108) 각각이 특정 동작을 수행할 수 있다는 것은, 프로세서(120), 또는 프로세서(122) 각각이 특정 동작을 수행하는 것으로 이해될 수 있다. 또는, 전자 장치(101), 또는 서버(108) 각각이 특정 동작을 수행할 수 있다는 것은, 프로세서(120), 또는 프로세서(122) 각각이, 전자 장치(101), 또는 서버(108) 각각에 포함된 하드웨어 또는 외부의 하드웨어로 하여금 특정 동작을 수행하도록 제어하는 것으로 이해될 수도 있다. 또는, 전자 장치(101), 또는 서버(108) 각각이 특정 동작을 수행할 수 있다는 것은, 메모리(130), 메모리(132) 각각에, 프로세서(120), 또는 프로세서(122) 각각 또는 하드웨어 중 적어도 하나로 하여금 특정 동작을 수행하도록 하는 인스트럭션들이 저장된 것으로 이해될 수도 있다.

[0018] 다양한 실시예에 따라서, 전자 장치(101)는, 프로세서(120), 메모리(130), 입력 장치(150), 표시 장치(160), 및 통신 회로(190)를 포함할 수 있다. 서버(108)는, 프로세서(122), 메모리(132), 및 통신 회로(192)를 포함할 수 있다.

[0019] 다양한 실시예에 따라서, 프로세서(120)는, 예를 들면, 소프트웨어(예: 프로그램)를 실행하여 프로세서(120)에 연결된 전자 장치(101)의 적어도 하나의 다른 구성요소(예: 하드웨어 또는 소프트웨어 구성요소)을 제어할 수 있고, 다양한 데이터 처리 또는 연산을 수행할 수 있다. 일실시예에 따르면, 데이터 처리 또는 연산의 적어도 일부로서, 프로세서(120)는 다른 구성요소(예: 통신 회로(190) 또는 센서 모듈(미도시))로부터 수신된 명령 또는 데이터를 휘발성 메모리에 로드하고, 휘발성 메모리에 저장된 명령 또는 데이터를 처리하고, 결과 데이터를 비휘발성 메모리에 저장할 수 있다. 일실시예에 따르면, 프로세서(120)는 메인 프로세서(예: 중앙 처리 장치 또는 어플리케이션 프로세서), 및 이와는 독립적으로 또는 함께 운영 가능한 보조 프로세서(예: 그래픽 처리 장치, 이미지 시그널 프로세서, 센서 허브 프로세서, 또는 커뮤니케이션 프로세서)를 포함할 수 있다. 추가적으로 또는 대체적으로, 보조 프로세서는 메인 프로세서보다 저전력을 사용하거나, 또는 지정된 기능에 특화되도록

설정될 수 있다. 보조 프로세서는 메인 프로세서와 별개로, 또는 그 일부로서 구현될 수 있다. 보조 프로세서는, 예를 들면, 메인 프로세서가 인액티브(예: 슬립) 상태에 있는 동안 메인 프로세서를 대신하여, 또는 메인 프로세서가 액티브(예: 어플리케이션 실행) 상태에 있는 동안 메인 프로세서와 함께, 전자 장치(101)의 구성요소들 중 적어도 하나의 구성요소(예: 표시 장치(160), 또는 통신 회로(190))와 관련된 기능 또는 상태들의 적어도 일부를 제어할 수 있다. 일실시예에 따르면, 보조 프로세서 (예: 이미지 시그널 프로세서 또는 커뮤니케이션 프로세서)는 기능적으로 관련 있는 다른 구성 요소(예: 통신 회로(190))의 일부로서 구현될 수 있다.

[0020] 메모리(130)는, 전자 장치(101)의 적어도 하나의 구성요소(예: 프로세서(120))에 의해 사용되는 다양한 데이터를 저장할 수 있다. 데이터는, 예를 들어, 소프트웨어(예: 프로그램) 및, 이와 관련된 명령에 대한 입력 데이터 또는 출력 데이터를 포함할 수 있다. 메모리(130)는, 휘발성 메모리 또는 비휘발성 메모리를 포함할 수 있다.

[0021] 입력 장치(150)는, 전자 장치(101)의 구성요소(예: 프로세서(120))에 사용될 명령 또는 데이터를 전자 장치(101)의 외부(예: 사용자)로부터 수신할 수 있다. 입력 장치(150)는, 예를 들면, 마이크, 마우스, 또는 키보드를 포함할 수 있다. 표시 장치(160)는 전자 장치(101)의 외부(예: 사용자)로 정보를 시각적으로 제공할 수 있다. 표시 장치(160)은, 예를 들면, 디스플레이, 홀로그램 장치, 또는 프로젝터 및 해당 장치를 제어하기 위한 제어 회로를 포함할 수 있다. 일실시예에 따르면, 표시 장치(160)는 터치를 감지하도록 설정된 터치 회로(touch circuitry), 또는 상기 터치에 의해 발생되는 힘의 세기를 측정하도록 설정된 센서 회로(예: 압력 센서)를 포함할 수 있다. 이 경우, 입력 장치(150) 및 표시 장치(160)가 터치스크린 장치로 구현될 수도 있다.

[0022] 통신 회로(190)은 전자 장치(101)와 외부 전자 장치(예: 외부 전자 장치(102), 또는 서버(108))간의 직접(예: 유선) 통신 채널 또는 무선 통신 채널의 수립, 및 수립된 통신 채널을 통한 통신 수행을 지원할 수 있다. 통신 회로(190)은 프로세서(120)(예: 어플리케이션 프로세서)와 독립적으로 운영되고, 직접(예: 유선) 통신 또는 무선 통신을 지원하는 하나 이상의 커뮤니케이션 프로세서를 포함할 수 있다. 일실시예에 따르면, 통신 회로(190)은 무선 통신 회로(예: 셀룰러 통신 회로, 근거리 무선 통신 회로, 또는 GNSS(global navigation satellite system) 통신 회로) 또는 유선 통신 회로(예: LAN(local area network) 통신 회로, 또는 전력선 통신 회로)을 포함할 수 있다. 이들 통신 회로 중 해당하는 통신 회로는 제 1 네트워크(예: 블루투스, WiFi direct 또는 IrDA(infrared data association) 같은 근거리 통신 네트워크) 또는 제 2 네트워크(예: 셀룰러 네트워크, 인터넷, 또는 컴퓨터 네트워크(예: LAN 또는 WAN)와 같은 원거리 통신 네트워크)를 통하여 외부 전자 장치와 통신할 수 있다. 이런 여러 종류의 통신 회로들은 하나의 구성 요소(예: 단일 칩)으로 통합되거나, 또는 서로 별도의 복수의 구성 요소들(예: 복수 칩들)로 구현될 수 있다. 무선 통신 회로는 가입자 식별 모듈에 저장된 가입자 정보(예: 국제 모바일 가입자 식별자(IMSI))를 이용하여 제 1 네트워크 또는 제 2 네트워크와 같은 통신 네트워크 내에서 전자 장치(101)를 확인 및 인증할 수 있다.

[0023] 프로세서(122)는, 프로세서(120)와 실질적으로 동일한 구성 요소를 포함하거나, 또는 실질적으로 동일한 기능을 제공할 수 있다. 메모리(132)는 메모리(130)와 실질적으로 동일한 구성 요소를 포함하거나, 또는 실질적으로 동일한 기능을 제공할 수 있다. 통신 회로(192)는 통신 회로(190)와 실질적으로 동일한 구성 요소를 포함하거나, 또는 실질적으로 동일한 기능을 제공할 수 있다.

[0024] 상기 구성요소들 중 일부 구성요소들은 주변 기기들간 통신 방식(예: 버스, GPIO(general purpose input/output), SPI(serial peripheral interface), 또는 MIPI(mobile industry processor interface))를 통해 서로 연결되어 신호(예: 명령 또는 데이터)를 상호간에 교환할 수 있다.

[0025] 일실시예에 따르면, 명령 또는 데이터는 서버(108)를 통해서 전자 장치(101)와 다른 전자 장치(예: 102)간에 송신 또는 수신될 수 있다. 다른 전자 장치(예: 102)는 전자 장치(101)와 동일한 또는 다른 종류의 장치일 수 있다. 일실시예에 따르면, 전자 장치(101)에서 실행되는 동작들의 전부 또는 일부는 다른 하나 또는 복수의 외부 전자 장치에서 실행될 수 있다. 일실시예에 따르면, 전자 장치(101)가 어떤 기능이나 서비스를 자동으로 또는 요청에 의하여 수행해야 할 경우에, 전자 장치(101)는 기능 또는 서비스를 자체적으로 실행시키는 대신에 또는 추가적으로, 그와 연관된 적어도 일부 기능을 외부 전자 장치에게 요청할 수 있다. 상기 요청을 수신한 외부 전자 장치는 요청된 기능 또는 추가 기능을 실행하고, 그 결과를 전자 장치(101)로 전달할 수 있다. 전자 장치(101)는 수신된 결과를 그대로 또는 추가적으로 처리하여 요청된 기능이나 서비스를 제공할 수 있다. 이를 위하여, 예를 들면, 클라우드 컴퓨팅, 분산 컴퓨팅, 또는 클라이언트-서버 컴퓨팅 기술이 이용될 수 있다.

[0026] 로컬 디퍼런셜 프라이버시(local differential privacy: LDP) 기법은, 합산자(예: 서버(108)) 및 복수(예: n개)의 개별 사용자(예: 전자 장치(101)) 사이에서 수행될 수 있으며, 개별 사용자(예: 전자 장치(101))는 사적인 정보를 포함하는 데이터 레코드(data record)를 저장할 수 있다. 본 문서에서, 사용자가 특정 동작을 수행

하는 것은, 전자 장치(101)가 특정 동작을 수행하는 것일 수 있으며, 합산자가 특정 동작을 수행하는 것은 서버(108)가 특정 동작을 수행하는 것일 수 있다. 각각의 사용자(u_i)(i는 1 부터 n까지의 자연수)는 국부적으로(locally) 자신의 레코드(t_i)를 교란(perturb)할 수 있으며, 이에 따라 랜덤화된 레코드($\tilde{t_i}$)를 획득할 수 있다. 전자 장치(101)는, 랜덤화된 레코드($\tilde{t_i}$)를 합산자(예: 서버(108))로 송신할 수 있다. 서버(108)는, 복수의 사용자 장치들로부터 수신된 랜덤화된 레코드($\tilde{t_1}, \tilde{t_2}, \cdots, \tilde{t_n}$)에 기반한 통계 결과를 계산할 수 있다. LDP 기법에서의 교란(perturbation)은, 합산자(예: 서버(108)), 또는 제3자(third party)가 랜덤화된 레코드($\tilde{t_i}$)로부터 정확한 레코드(t_i)를 도출할 수 없도록 하며, 교란의 정도는 파라미터(ε)에 의하여 제어될 수 있다. LDP 기법은 하기와 같이 정의될 수 있다.

[0027] 다양한 실시예에서, 랜덤화 함수(f)가 ε-LDP를 만족하는 명제는, 두 개의 입력 터플(tuple)인 t, t' ∈D(f)이며, 임의의 가능한 f 함수의 출력인 $\tilde{t}$ 에 대하여 수학식 1을 만족하는 명제와 필요충분조건일 수 있다. 여기에서, D(f)는 f의 도메인일 수 있다.

수학식 1

[0028] $$\Pr\left[f(t)=\tilde{t}\right] \leq e^{\epsilon} \cdot \Pr\left[f(t')=\tilde{t}\right]$$

[0029] 상기에서, 파라미터(ε)가 작은 경우에는, ε-LDP에 의한 랜덤화된 레코드($\tilde{t_i}$)에 대하여, 원래의 데이터가 t, t' 중 어떤 것인지를 더 도출하기 어렵게 된다. 즉, 더 작은 파라미터(ε)를 가질수록, 더욱 강력한 프라이버시 보호가 가능할 수 있다. LDP가 디퍼런셜 프라이버시 이론에 기반하였기 때문에, 이는 결합성 속성(composability properties)를 야기하며, 이하에서는 순차적인 결합(sequential composition) 속성을 설명하도록 한다.

[0030] m개의 독립적인 랜덤화 함수(f_1, f_2, , f_m)이 ε-LDP를 만족하는 경우에, g(f_1, f_2, , f_m)의 함수는, $\left(\sum_{i=1}^{m} \epsilon_i\right)$-LDP를 만족할 수 있다. 이에 따라, ε는 프라이버시 버짓(privacy budget)일 수 있다. 다양한 실시예에 따른 전자 장치(101)는, LDP-만족 메커니즘의 시리즈를 적용할 수 있으며, 순차적인 결합마다 프라이버시 버짓(ε)의 부분이 할당되며, LDP-만족 메커니즘의 시리즈는 전체적으로 ε-LDP를 만족할 수 있다.

[0031] ε-LDP를 적용하는 기본적인 메커니즘은 랜덤화된 응답(randomized resposne: RR) 메커니즘일 수 있으며, RR 메커니즘은 각 사용자가 정보의 단일 비트(즉, 0 또는 1)를 소지하며, 합산자가 ε-LDP 하에서 어떤 사용자가 1의 비트를 소지하였는지에 대한 퍼센테이지를 컴퓨팅하는 것에 대한 것이다. 하기는, 알고리즘 1에 관한 것으로, RR 메커니즘의 변형된 예시이다.

Algorithm 1: RandomizedResponse(t, ϵ)

Input : $t \in [0,1]$ and privacy parameter ϵ

Output: $\tilde{t} \in \left\{-\frac{1}{e^{\epsilon}-1}, \frac{e^{\epsilon}}{e^{\epsilon}-1}\right\}$

Sample a *Bernoulli* variable x such that

$$Pr[x=1] = \frac{t \cdot (e^{\epsilon}-1)+1}{e^{\epsilon}+1};$$

if $x = 1$ **then**

  return $\tilde{t} = \frac{e^{\epsilon}}{e^{\epsilon}-1}$;

else

  return $\tilde{t} = -\frac{1}{e^{\epsilon}-1}$;

[0032]

[0033] PR은 확률을 나타낼 수 있다. 각 사용자 u_i는 입력값으로서의 단일 비트인 레코드(ti)에 RR 메커니즘을 적용할 수 있으며, 출력값인 랜덤화된 레코드($\tilde{t_i}$)를 합산자로 보고(report)할 수 있다. 합산자는, 타겟값인 $\frac{1}{n}\sum_i t_i$를 획득하기 위하여 $\frac{1}{n}\sum_i \tilde{t_i}$를 계산할 수 있다. RR 메커니즘의 출력값은 두 가지 가능한 값을 가질 수 있으며, 각각의 값은 프라이버시 침해 없이 합산자에 의하여 계산될 수 있다. 이에 따라, 사용자는 단순히 단일 비트를 합산자로 송신할 수 있으며, 이에 따라 통신 오버헤드가 감소할 수 있다.

[0034] RR 메커니즘은 사용자의 레코드가 단일 비트가 아닌 어플리케이션에도 적용될 수 있다. 그러한 어플리케이션은 카테고리 속성에 대한 히스토그램 예측과 연관될 수 있으며, 각각의 사용자 u_i는 카테고리 값(t_i)을 소지할 수 있다. 이 경우, 합산자의 목적은 도메인 내의 각 값의 빈도를, LDP 하에서 예측할 수 있다. 다양한 실시예에 따른 전자 장치(101)는, 레코드(t_i)를 비트 벡터로 변환할 수 있으며, 벡터 내의 비트 각각에 RR 메커니즘을 적용할 수 있다.

[0035] 다양한 실시예에 따른 전자 장치(101) 및 서버(108)는 RAPPOR 기법에 기반하여 데이터를 처리할 수도 있다. RAPPOR 기법에서의 LDP는 상술한 RR 메커니즘일 수 있으며, 블룸 필터(bloom filter) 레버리징 최적화와 연관될 수 있다. 다양한 실시예는, 예를 들어 사용자의 OS(operating system)와 같은 단순한 카테고리 값에 대한 통계 데이터를 수집하는 것에 관할 수 있다. RAPPOR에 의하여 URL 및 이미지 태그와 같은 임의의 스트링(string)의 빈도가 예측될 수 있으며, 이 경우 스트링의 도메인은 미리 알려지지 않을 수 있다. 특히, 이러한 설정에서, 각각의 사용자는 M 길이(length)의 스트링을 소지할 수 있으며, M 캐릭터보다 짧은 스트링은 패딩(padded)될 수 있으며, 이는 합산자에게 스트링의 진정한 길이를 노출하는 것을 방지하기 위함이다. 이러한 스트링은 임의일 수 있으며, 합산자에게 알려지지 않을 수 있다. 서버(108)는, LDP를 만족하면서, 사용자들 사이에서 빈도가 높은 스트링을 발견할 수 있다.

[0036] 예를 들어, 서버(108)는 상술한 RR을 이용하여 가능한 스트링 각각의 빈도를 직접적으로 수집할 수 있다. 다만, M-길이의 스트링의 큰 공간이 문제가 될 수도 있다. 더욱 상세하게, $\mathcal{I}$를 알파벳 집합이라고 상정하며, $|\mathcal{I}|$를 카디널리티(cardinality)라고 상정한다. 이 경우, M-길이의 스트링에 대한 도메인 사이즈는, $|\mathcal{I}|^M$ 일 수 있으며, 계산에 많은 리소스가 요구될 수도 있다.

[0037] 다양한 실시예에 따른 서버(108)는, LDP 하에서 잠재적으로 빈도가 높은 스트링의 비교적 작은 후보 집합(C)를, 프라이버시 버짓(ε)의 제 1 일부(ε_1)를 이용하여 계산할 수 있다. 이후, 서버(108)는, 프라이버시 버짓(ε)

의 나머지($\varepsilon-\varepsilon_1$)를 이용하여 각각의 후보에 대한 계산을 수행할 수 있으며, 이는 ε-LDP를 만족할 수 있다. 비교적 작은 후보 집합(C)을 계산하기 위하여, 각각의 사용자(u_i)는 자신의 스트링(t_i)로부터 두 개의 n-그램을 랜덤하게 샘플링하여야 하며, 이 경우 n은 M에 비하여 작은 수여야 한다. 서버(108)는, 두 개의 n-그램을 RAPPOR 기법을 이용하여 각각의 사용자로부터 수집하고, 비교적 작은 후보 집합(C)을 도출하기 위하여 수집된 정보에 기댓값 최대화(expectation maximization)를 포함하는 알고리즘을 적용할 수 있다. 특히, RAPPOR 기법에 의하여 각각의 노드가 n-그램에 대응하는 그래프(G)가 도출될 수 있다. 그래프(G) 내에서 각각의 엣지는 자주 함께 등장하는 두 개의 n-그램을 연결할 수 있다. 이러한 그래프로부터, 서버(108)는 K-클리크(clique)를 계산할 수 있으며, K는 M을 n으로 나눈 값일 수 있다. 서버(108)는, 클리크 각각으로부터 텀(term)을 재구성할 수 있으며, 재구성한 텀을 후보 집합(C)에 포함시킬 수 있다. 다만, 후보 집합(C)이 노이즈가 포함된 상태에서 구성될 수 있으며, 엣지의 부재에 기인한 그래프(G)로부터의 실수 텀에 대응하는 K 클리크의 부재에 의하여 결과값이 만족스럽지 못할 수도 있다. 아울러, 연산량이 많이 요구될 수도 있다.

[0038] 다양한 실시예에 따른 전자 장치(101)는, LDP를 만족하는 트리(trie)에 기반한 BSL 기법을 이용할 수 있다. 모든 사용자들은 동일한 키보드 사전을 공유할 수 있으며, 각 사용자(u_i)는 데이터 레코드(t_i)를 소지할 수 있으며, 데이터 레코드(t_i)는 사전에 등재되지 않을 수 있다. 데이터 레코드(t_i) 각각은 알파벳($\mathcal{I}$)으로부터의 임의의 캐릭터에 의하여 구성될 수 있다. 만약, 사용자(u_i)가 복수의 텀들을 포함하는 경우에는, 하나를 랜덤하게 선택할 수 있다. 합산자는, 사용자들간에서 ε-LDP 하에서 상위-k 빈도가 높은 새로운 텀들을 수집할 수 있다. 형식적으로, 주어진 텀(t)에 대하여, t를 소지하는 사용자의 숫자가 t의 서포트(support)인 supp(t)로 정의될 수 있다. 합산자는, 모든 가능한 텀들로부터 가장 높은 서포트 값을 가지는 k 텀들을 식별할 수 있다.

[0039] BSL 기법은 상술한 RAPPOR 기법과 유사한 프레임워크에 기반할 수 있다. BSL 기법은 전체의 프라이버시 버짓(ε)을 $\varepsilon_1+\varepsilon_2$로 나눌 수 있으며, RR 메커니즘을 이용하여 ε_2-LDP 하에서 상위-k 빈도가 높은 새로운 텀들을 수집할 수 있으며, ε_1-LDP 하에서 신규 텀들의 후보 집합(C)를 수집할 수 있다. 순차적인 결합(sequential composition)에 의하여, 두 단계 전체는 ε-LDP를 만족시킬 수 있다.

[0040] BSL 기법은, 후보 집합(C)를 계산하는 것에 있어서 RAPPOR 기법과 상이할 수 있다. BSL 기법은 상하 방향으로 새로운 텀들의 트리를 반복적으로 구성할 수 있다. 예를 들어, 도 X에서와 같이, 서버(108)는, 루트(예: v1)이 알파벳에 포함되어 있지 않은 제 1 특수 문자(예: $)를 포함하는 트리를 형성할 수 있으며, 제 1 특수 문자는 텀의 시작을 나타낼 수 있다. 서버(108)에 의하여 형성된 트리는, 리프(leaf)가 아닌 노드가 $|\mathcal{I}|$+1의 자식(children) 노드를 가지며, 자식 노드는 알파벳 내의 캐릭터 또는 제 2 특수 문자(예:&)를 포함할 수 있다. 제 2 특수 문자는, 텀의 종료를 나타낼 수 있다. 각각의 노드 v는 프리픽스(prefix)(p(v))에 대응할 수 있으며, 프리픽스(p(v))는 루트(예: v1)로부터 노드(v)로의 경로(path) 상의 각각의 노드를 연결(concatenating)함으로써 획득될 수 있다. 예를 들어, $p(v_9)$는 "$BB"일 수 있다. 각각의 노드(v)에 대하여, 합산자는 서포트(c(v)) 및 프리픽스(p(v))를 예측할 수 있으며, 이는 결국 프리픽스(p(v))로 시작하는 텀을 가지는 사용자의 개수를 예측하는 것일 수 있다. 프라이버시 비보호 셋팅(non-private setting)에서는, 리프는 제 2 특수 문자 또는 "0"의 서포트와 연관될 수 있다. 서버(108)는, 동일한 프리픽스를 가지는 텀들(예: t_2, t_3, t_5, t_7, t_8, t_9)을, 예를 들어 v_9에 루트된 동일한 브랜치 하에 그룹핑함으로써 텀들에 대한 인덱스를 부여할 수 있다.

[0041] BSL 기법에서, 합산자는 ε_1-LDP 하에서 트리를 형성할 수 있으며, 트리 내에서의 완성된 텀들 각각을 후보 집합(C)에 추가할 수 있다. 하기의 알고리즘 2는 트리 형성 과정을 나타낸다.

Algorithm 2: BSL_Build_Trie(ϵ_1, θ, ℓ)

Initialize the trie $\mathcal{T}$ with a single root node v_r containing the term-start symbol \$, and mark v_r as unvisited;
while *there exists an unvisited non-leaf node* v **do**
  Mark v as visited;
  if *length of* $p(v) \leq \ell + 2$ **then**
    Break;
  for *each character* $i \in (\mathcal{I} \cup \{\text{“\&”}\})$ **do**
    Add a child v_c of v containing character i;
    Compute the prefix $p(v_c)$ of v_c;
    Apply RR to collect information from all users to estimate $c'(v_c)$ of $c(v_c) = supp(p(v_c))$, using privacy budget $\frac{\epsilon_1}{2(\ell+1)}$;
    if $c'(v_c) < \theta$ **then**
      Mark v_c as pruned;
    else
      Mark v_c as unvisited;
Initialize candidate set C to empty;
for *each leaf node* v_l *that contains symbol* “&” **do**
  Add $p(v_l)$ to C;
return C.

[0042]

[0043] 합산자는, 알고리즘 2의 제 1 라인에서, 루트 노드와 함께 트리 형성을 개시할 수 있다. 합산자는, 상기 알고리즘 2에서와 같이, 노드를 반복적으로 분기시킬 수 있다. 합산자는, 리프가 아닌 노드 v에 대하여, LDP 하에서 데이터를 수집할 수 있으며, v에 대응하는 프리픽스의 서포트인 c(v)에 대응하는 c'(v)를 예측할 수 있다. 합산자는, c'(v)가 지정된 값인 Θ보다 작은지 여부를 판단할 수 있다. c'(v)가 지정된 임계치인 Θ보다 작은 것으로 판단되면, 합산자는 해당 v를 종료된(pruned) 것으로 처리할 수 있다. c'(v)가 지정된 임계치인 Θ 이상인 경우에는, 합산자는 노드 v에 $|\mathcal{I}|$+1의 자식 노드들을 추가할 수 있으며, 이는 $\mathcal{I}$ 및 제 2 특수 문자(예: &)에 대응할 수 있다. 합산자는, 다음 반복을 수행할 수 있으며, 리프가 아닌 노드를 다시 분기할 것을 시도할 수 있다. 합산자는, 알고리즘 2을 통하여 트리 내의 완성된 텀들로 구성된 후보 집합(C)를 생성할 수 있다.

[0044] BSL은 텀 l의 최대 길이를 설정하여야 하며, l의 설정 과정에서 프라이버시 버짓(ε)의 일부인 ε_3가 이용될 수 있다. 즉, 프라이버시 버짓(ε)의 일부인 ε_1 및 ε_2는 상술한 과정에서 이용될 수 있으며, ε_3는 l의 설정에 이용될 수 있다. 주어진 l에 대하여, 트리의 최대의 뎁스(depth)는 l+2일 수 있으며, 이는 제 1 특수 문자(예: \$) 및 제 2 특수 문자(&)가 텀에 포함되는 것으로부터 기인한다. 루트의 서포트는 사용자들의 개수 n과 동일할 수 있으며, 이는 LDP 셋팅 내에서 합산자에게 알려져 있다. 합산자(예: 서버(108))는, 사용자들(예: 전자 장치들(101,102,103))과 통신할 수 있으므로, 루트의 서포트를 확인할 수 있다. BSL은 RR 메커니즘의 콜(call) 각각에 대하여 프라이버시 버짓을 $\varepsilon_1/2(l+1)$만큼 할당할 수 있다.

[0045] 다양한 실시예에 따라서, 내부 노드를 분기하기 위한 지정된 임계치인 Θ가 설정될 수 있다. 주어진 노드 v 및 대응하는 프리픽스 p(v)에 대하여, 예측된 p(v)의 서포트가 예측된 노이즈 레벨보다 작으면, v는 분기되지 않을 수 있으며, 이는 실제의 서포트(c(v))가 0일 수 있기 때문이다. 이하에서는, 프리픽스 p(v)의 예측된 서포트인 c'(v)의 노이즈 레벨에 대한 분석을 설명하도록 한다.

[0046] 트리 내의 임의의 노드 v에서, c'(v)는 상술한 알고리즘 2에서 얻어진 p(v)의 서포트일 수 있으며, c(v)는 p(v)의 정확한 서포트일 수 있다. 1-β의 확률로 수학식 2가 성립할 수 있다. β는 0 이상 1 이하의 실수로, 확률을 결정할 수 있다.

수학식 2

[0047] $$|c'(v) - c(v)| = O\left(\frac{(\ell+1)\cdot\sqrt{n\ln(1/\beta)}}{\epsilon_1}\right)$$

[0048] 수학식 2에서, n은 사용자들의 전체 개수일 수 있으며, O()는, 교란된 데이터 레코드로부터 획득된 프리픽스의 서포트(c'(v))와, 실제 데이터의 서포트(c(v))가 O의 괄호 내부의 값으로 바운드 된다는 의미이다. 다양한 실시예에 따른 서버(108)는, Θ를 $\frac{\eta(\ell+1)\cdot\sqrt{n}}{\epsilon_1}$ 로 설정할 수 있으며, η는 시스템 파라미터일 수 있으며, 예를 들어 5의 값을 가질 수 있으나 제한은 없다.

[0049] BSL을 적용하는 경우에, RR 메커니즘의 적용에 따라, 예를 들어 $\varepsilon_1/2(l+1)$만큼의 비교적 작은 프라이버시 버짓이 할당될 수 있으며, 프라이버시의 버짓이 감소함에 따라 부정확한 예측이 발생할 가능성도 있다. 예를 들어, l이 상대적으로 큰 경우에, 부정확한 예측 발생 가능성이 있을 수 있다.

[0050] 다양한 실시예에 따라서, 전자 장치(101), 또는 서버(108) 중 적어도 하나는, 프라이버시 버짓을 트리 내의 레벨별로 할당하지 않을 수 있다. 전자 장치(101), 또는 서버(105)는 사용자들을 구분할 수 있다. 해당 과정을 PrivTrie 기법이라 명명할 수 있으며, 전자 장치(101), 또는 서버(108) 중 적어도 하나는 PrivTrie 기법에 기반하여, 사용자들의 집합을 l+1개의 동일한 크기의 그룹으로 랜덤하게 구분할 수 있으며, 트리의 레벨 각각에 대응될 수 있다. 이러한 알고리즘 변경에 따라서, 예측되는 프리픽스 서포트의 정확도가 $O(\sqrt{\ell})$의 팩터만큼 향상될 수 있다. 성능이 향상된 기법은 IBSL이라 명명할 수 있다. 알고리즘 3은 합산자에 의한 트리 생성 과정을 나타낸다. 알고리즘 3은 ε_1-LDP를 만족할 수 있다.

Algorithm 3: Improved_BSL_Build_Trie(ϵ_1, θ, ℓ)

1 Randomly partition all users into ℓ+1 equal-sized groups G_1, $\cdots$, G_ℓ, with $\lfloor \frac{n}{\ell+1} \rfloor$ users each;
2 Same as line 1 in Algorithm 2;
3 **while** *there exists an unvisited non-leaf node* v **do**
4 Let d ($0 \le d \le \ell$) be the depth of node v;
5 Same as Lines 3-5 in Algorithm 2;
6 **for** *each character* $i \in (\mathcal{I} \cup \{\text{“\&”}\})$ **do**
7 Same as Lines 7-8 in Algorithm 2;
8 Apply RR to collect information from users in group G_{d+1} to compute an estimation $c'(v_c)$ of $c(v_c) = supp(p(v_c))$, using privacy budget $\frac{\epsilon_1}{2}$;
9 Same as Lines 10-13 in Algorithm 2;
10 Same as Lines 14-17 in Algorithm 2;

[0051]

[0052] 알고리즘 2와 비교하여, 트리 내부의 노드 v를 분기하고자 하는 경우에, 서버(108)는 노드 v에 대응하는 사용자

그룹, 즉 d_v 번째 그룹으로부터 정보를 수집할 수 있다. 여기에서, d_v는 뎁스가 0인 루트인 경우의 노드 v의 뎁스를 나타낼 수 있다. 서버(108)는 프라이버시 버짓을 $\varepsilon_1/2$만을 소요하여 노드 v에 대응하는 사용자 그룹, 즉 d_v 번째 그룹으로부터 정보를 수집할 수 있다. 모든 n개의 사용자들 사이에서의 노드 v의 서포트를 예측하기 위하여, 합산자는, l+1의 팩터에 의하여 수집된 예측된 서포트를 스케일링할 필요가 있다. IBSL은 BSL과 비교하여 $\sqrt{\ell+1}$ 의 팩터로 감소하는 향상된 정확도를 가질 수 있으며, 이는 후술하는 이유에 의한 것일 수 있다.

[0053] 트리 내의 임의의 노드에서, c'(v)가 알고리즘 3에서 획득된 p(v)의 예측된 서포트이며, c(v)가 p(v)의 정확한 서포트인 경우에, 1-β의 확률로 수학식 3이 성립할 수 있다.

수학식 3

[0054]
$$|c'(v) - c(v)| = O\left(\frac{\sqrt{n(\ell+1)\ln(1/\beta)}}{\epsilon_1}\right)$$

[0055] 예를 들어, Θ는 $\frac{\eta\sqrt{(\ell+1)n}}{\epsilon_1}$ 로 설정될 수 있으며, η는 시스템 파라미터일 수 있으며, 예를 들어 5의 값을 가질 수 있으나 제한은 없다. 비록 IBSL이 BSL에 비하여 향상된 정확도를 가지나, 사용자의 적은 참여를 야기하며, l이 미리 설정되어야 할 수 있다.

[0056] 다양한 실시예에서, 트리 내의 일부 노드들, 예를 들어 상대적으로 높은 레벨을 가지는 노드들은, 대응하는 프리픽스에 대한 높은 서포트를 가질 수 있다. 예를 들어, 도 3에서의 상대적으로 높은 뎁스의 노드(v3)는 "7"의 진정한 서포트를 가질 수 있다. 이러한 노드 v의 데이터 레코드에 상대적으로 크게 교란을 한다 하더라도, 예측된 서포트 c'(v)는 상대적으로 큰 값을 가질 가능성이 크다. 이러한 측면에서, 서버(108)는, 서포트의 다소 낮은 수준의 예측(coarse grained estimate)에 기반하여도 노드(v)를 분기할지 여부를 판단할 수 있다. 즉, 서버(108)는 노드(v)의 서포트의 예측에 더 적은 수의 사용자들을 할당할 수 있다. 아울러, 주어진 노드의 서포트 예측에 얼마만큼의 사용자가 할당되어야 하는지가 불명확할 수 있으며, 이는 해당 노드가 높은 서포트를 가지는지 여부가 미리 서버(108)에 알려지지 않았기 때문일 수 있다. 예를 들어, 트리 내의 높은 뎁스 레벨은 도 3의 v2와 같이 항상 높은 서포트를 가지지는 않을 수도 있다.

[0057] 다양한 실시예에서, 서버(108)는 상술한 바와 같이, 사용자들 각각에 단일 레벨을 할당할 수 있다. 예를 들어, 서버(108)는, 하나의 사용자가 여러 노드에 대하여 참여하도록 할 수 있다. 사용자 u_i는 노드들 Vi의 임의의 세트에 대한 서포트 예측 프로세스에 관여될 수 있으며, 프라이버시 버짓은 $\varepsilon_1/2$일 수 있다. 이 경우, Vi 내의 노드는 다른 노드의 부모 노드가 아닐 수 있다.

[0058] 더욱 상세하게, 임의의 사용자 ui(i는 1 이상 n 이하의 자연수)에 대하여, 합산자가 사용자 ui로부터 정보를 수집하여 노드들(Vi)의 셋트의 서포트 값을 예측하고, 이는 프라이버시 버짓을 $\varepsilon_1/2$ 만큼 소요한 RR을 이용하여 수행될 수 있다. 아울러, 노드들(Vi)의 셋트의 원소인 v, w는 서로가 어느 하나의 부모 노드가 될 수 없다. Vi 전체에 대한 데이터 수집은, 사용자 ui각각에 대하여 ε_1-LDP를 만족할 수 있다.

[0059] 이에 따라, 주어진 노드 v에 대하여, 서버(108)는 임의의 노드 v의 부모 노드에 대한 서포트 값의 예측에 참여하지 않았던 사용자들의 집합을, 이용가능한 사용자 세트(U(v))로 판단할 수 있다. 예를 들어, 서버(108)는, 도 3의 노드(v11)의 이용가능한 사용자 세트(U(v11))로서, 노드(v11)의 부모 노드(예: 노드(v3) 및 노드(v9))와 관련하여 데이터 수집에 참여하지 않은 사용자들을 확인할 수 있다.

[0060] 다양한 실시예에 따라서, 서버(108)는 모든 리프 노드(vl)에 대하여, 서포트(c'(vl))를 예측하기 위하여 전체 이용가능한 사용자 셋트(U(vl))를 이용할 수 있다. 그러한 리프 노드는, 종료를 나타내는 문자(예: &)를 포함하는 노드이거나, 또는 BSL 또는 IBSL에서 설명한 바와 같이, 서포트가 예측된 예측 에러 미만인 노드일 수 있다. 리프 노드(vl)이 자식 노드를 가지지 않으므로, 상하 방향의 트리 형성 방향에 의하여 리프 노드(vl)이 모

든 부모 노드들은 서버(108)에 의하여 확인된 상태일 수 있다. 이 경우, U(v1)의 이용가능한 사용자 셋트가 이용될 수 있으며, 해당 이용가능한 사용자 셋트는 다른 확인되지 않은 노드들의 데이터 수집에 영향을 미치지 않을 수 있다. 예를 들어, 도 3의 노드(v5)를 참고하면, 노드(v5)는 부족한 서포트에 의하여 분기가 종료될 수 있으며, 서버(108)는 노드(v5)에 대응하는 이용가능한 사용자 셋트(U(v5))의 모든 사용자들을 c'(v5)를 예측하는데 이용할 수 있다. 사용자 셋트(U(v5))는, 예를 들어 노드(V5)의 부모 노드(예: 노드(v2))의 서포트 예측에 참여하지 않은 사용자들을 포함할 수 있다. 해당 사용자 셋트(U(v5)는, 상술한 바에 따라서 또 다른 노드에 대한 데이터 수집에도 참여할 수도 있다.

[0061] 상술한 바와 같은 PrivTrie가 하기의 알고리즘 4로 표현될 수 있다.

Algorithm 4: PrivTrie_Build_Trie(ϵ_1)

1 Same as Line 1 in Algorithm 2;
2 Set available user set $U(v_r)$ for root v_r to $\{u_1, u_2, \cdots, u_n\}$;
3 **while** *there exists an unvisited non-leaf node* v **do**
4 Mark v as visited;
5 Let $U(v)$ be the set of available users for node v;
6 **for** *each character* $i \in (\mathcal{I} \cup \{\text{“\&”}\})$ **do**
7 Same as Lines 7-8 in Algorithm 2, which add a new node v_c as a child of v;
8 Initialize estimated support $c'(v_c)$ to 0;
9 Initialize available user set $U(v_c)$ to $U(v)$;
10 **while** $U(v_c)$ *is not empty* **do**
11 Randomly sample a batch of users U' from $U(v_c)$;
12 Apply RR to collect information from U' using privacy budget $\frac{\epsilon_1}{2}$, in order to incrementally refine the estimated support $c'(v_c)$ of v_c;
13 Update θ according to Equation (1);
14 **if** $i \neq$ “&” *and* $c'(v) \geq \theta$ **then**
15 Mark v_c as unvisited;
16 Break;
17 Remove all users in U' from $U(v_c)$;
18 **if** $U(v_c)$ *is empty* **then**
19 Mark v_c as pruned;
20 Same as Lines 14-17 in Algorithm 2;

[0062]

[0063] 다양한 실시예에 따른 서버(108)는, 내부 노드(vc)에 대하여 노드(vc)의 서포트 예측에 연관할 사용자의 개수를 미리 결정하지 않고, U(vc)로부터 이용가능한 사용자들에게 한번에 하나의 배치(batch)를 요청할 수 있으며, 매번 예측된 서포트(c'(vc))를 갱신(refine)할 수 있다. 예를 들어, 하나의 배치의 사이즈는 예를 들어 1000일 수 있으나, 사이즈에는 제한이 없다. 알고리즘 4는, 예를 들어 c'(v)가 지정된 임계치인 Θ를 초과하거나, 또는 합산자가 U(vc)의 모든 이용가능한 사용자들에게 데이터를 요청하였으나, c'(vc)가 지정된 임계치인 Θ 미만인 경우에 중단될 수 있다. c'(v)가 지정된 임계치인 Θ를 초과하는 것은 노드(vc)가 분기되어야 함을 의미할 수 있다. 합산자가 U(vc)의 모든 이용가능한 사용자들에게 데이터를 요청하였으나, c'(vc)가 지정된 임계치인 Θ 미만인 것은, 노드(v)가 분기되어야 하지 말아야 함을 의미할 수 있다.

[0064] 다양한 실시예에 따른 서버(108)는 서포트 임계치 Θ를 적응적으로 변경할 수 있다. 예를 들어, 서버(108)는, 단계적으로 노드의 서포트를 갱신할 수 있으며, 이용가능한 사용자로의 요청 및 수신된 데이터 레코드에 기반하여 서포트를 갱신할 수 있다. 사용자의 신규 배치로부터의 데이터 수집을 완료한 이후에 적절한 임계치 Θ를 설정하도록, 서버(108)는 하기와 같이, 주어진 갯수의 사용자의 예측된 서포트 값 내의 에러를 분석할 수 있다.

[0065] 트리 내의 임의의 노드(v)에 대하여, m개의 사용자들로부터의 데이터를 수집한 이후에 알고리즘 4에서 획득된 p(v)의 예측된 서포트가 c'(v)라고 상정한다. c(v)가 p(v)의 정확한 서포트인 경우에, 1-β의 확률로 수학식 4가 성립할 수 있다.

수학식 4

$$|c'(v) - c(v)| = O\left(\frac{n\ln(1/\beta)}{\epsilon_1\sqrt{m}}\right)$$

[0066]

[0067] 그러므로, 알고리즘 4의 13라인에 따라서, 서버(108)는 Θ를 반복적으로 갱신할 수 있으며, 예를 들어 수학식 5와 같은 갱신 과정이 수행될 수 있다.

수학식 5

$$\theta = \frac{\eta \cdot n}{\epsilon_1\sqrt{m}}$$

[0068]

[0069] m은 상술한 바와 같이, U(vc)로부터 샘플링된 사용자의 개수이며, η는 시스템 파라미터일 수 있으며, 예를 들어 5의 값을 가질 수 있으나 제한은 없다. 알고리즘 4는 ϵ_1-LDP를 만족할 수 있다. 알고리즘 2 및 알고리즘 3과 비교하여, 알고리즘 4는 노드 서포트 값의 적응적 예측을 가능하도록 하고, 텀의 최대 길이(l)을 미리 설정하지 않도록 한다. 이에 따라, 알고리즘 4는 높은 정확도를 가질 수 있다.

[0070] C가 알고리즘 4에 의하여 식별된 후보 텀의 집합이며, $\mathcal{T}$ 가 형성된 트리를 나타내도록 상정한다. V는 $\mathcal{T}$ 내의 노드인 것을 상정하도록 하며, Vl은 $\mathcal{T}$ 내의 리프 노드인 것을 상정한다. C는 후보 텀에 대응하는 노드의 세트를 나타낼 수 있다. 서버(108)는 트리의 형성에 참여하지 않는 사용자들의 집합인 n_2를 확인할 수 있다. 서버(108)는 확인된 사용자들에 대하여 RR을 C회 적용할 수 있으며, C 내에서의 텀의 빈도의 예측이 획득될 수 있다. 이에 따라서, 서버(108)는, $\mathcal{T}$ 내의 노드들에 대한 |V|+|C|의 서포트 예측을 획득할 수 있다. 더욱 상세하게, 알고리즘 4에 의하여, V에 포함되는 v 노드들에 대한 노이즈가 포함된 예측 값인 c''(v)가 획득될 수 있다. 이러한 예측들은 바이어스되지 않고(unbiased), 예측 값은 아래의 제약들을 가질 수 있다. C에 포함되는 노드(v) 각각에 대하여, 두 개의 예측값인 c'(v) 및 c''(v)가 동일한 값을 가지며, 노드 v가 S 세트의 자식 노드를 가지면, 수학식 6이 성립할 수 있다.

수학식 6

$$\mathbb{E}[c'(v)] = \sum_{u \in S} \mathbb{E}[c'(u)]$$

[0071]

[0072] 수학식 6에서의 E는 랜덤화된 데이터들로부터 얻은 노드에서의 빈도값의 기댓값(또는, 평균)을 나타낼 수 있다. 상술한 제약들이 더 높은 정확도를 획득하기 위하여 노이즈가 포함된 예측값을 갱신하는ㄷ 이용될 수 있다. 새로운 예측 값인 $\hat{c}(v)$ 가 노드 v에 대하여 도출될 수 있으며, 모든 $\hat{c}(v)$는 일관성 제약(consistency constraint)을 만족할 수 있으며, 노이즈 예측값까지의 거리인 L2 또한 최소화할 수 있다. 상술한 알고리즘은, 각각의 노드가 하나의 예측값만을 가지며, 각각의 예측은 동일한 변수를 가짐을 전제로 할 수 있다.

[0073] 서버(108)는, 노드(v) 각각에 대하여, c'(v) 및 c''(v)의 변수들이 σ'(v) 및 σ''(v)가 되도록 할 수 있다. 서버(108)는, child(v)를 노드 v의 자식 노드의 셋트로 설정한다. 서버(108)는, 수학식 8을 만족시키면서 수학식 7의 연산 결과가 최소화 되도록 함으로써, 새로운 예측 값인 $\hat{c}(v)$ 를 획득할 수 있다.

수학식 7

$$\sum_{v \in V} \left(\frac{\hat{c}(v) - c'(v)}{\sigma'(v)} \right)^2 + \sum_{v \in C} \left(\frac{\hat{c}(v) - c''(v)}{\sigma''(v)} \right)^2$$

[0074]

수학식 8

$$\forall v \in V,\ \hat{c}(v) = \sum_{u \in child(v)} \hat{c}(u)$$

[0075]

[0076] 서버(108)는, 두 페이즈(phase)로 구성된 새로운 예측 값인 $\hat{c}(v)$ 을 도출할 수 있다. 서버(108)는, 버텀-업 페이즈(bottom-up phase)에서, 트리 내의 노드는 버텀-업 방식으로 확인할 수 있으며, 네 개의 중간값인 σ(v), r(v), s(v), 및 z(v)를 각각의 노드 v에 대하여 하기 수학식 9 내지 12에 의하여 확인할 수 있다.

수학식 9

$$\sigma(v) = \begin{cases} \frac{\sigma'(v) \cdot \sigma''(v)}{\sqrt{(\sigma'(v))^2 + (\sigma''(v))^2}}, & \text{if } v \in C \\ \sigma'(v), & \text{otherwise} \end{cases}$$

[0077]

수학식 10

$$r(v) = \begin{cases} \sigma^2(v), & \text{if } v \text{ is a leaf} \\ \frac{1}{1 + \sum_{u \in child(v)} \frac{s(u)}{\sigma^2(v)}}, & \text{otherwise} \end{cases}$$

[0079]

수학식 11

$$s(v) = \begin{cases} r(v), & \text{if } v \text{ is a leaf} \\ r(v) \cdot \sum_{u \in child(v)} s(u), & \text{otherwise} \end{cases}$$

[0080]

수학식 12

$$z(v) = \begin{cases} r(v) \cdot \left(\frac{c'(v)}{(\sigma'(v))^2} + \frac{c''(v)}{(\sigma''(v))^2} + \sum_{u \in ancestor(v)} \frac{c'(u)}{\sigma^2(u)} \right) & \text{if } v \in C \\ r(v) \cdot \sum_{u=v \vee u \in ancestor(v)} \frac{c'(u)}{\sigma^2(u)}, & \text{if } v \text{ is a leaf} \\ r(v) \cdot \sum_{u \in child(v)} z(u), & \text{otherwise} \end{cases}$$

[0081]

[0082] ancestor(v)는 노드 v의 부모 노드의 세트를 나타낼 수 있다. 서버(108)는, 이후 탑-다운 페이즈에서, 트리의 루트로부터 breath-first search를 수행함으로써, 노드 v 각각에 대한 새로운 예측 값인 $\hat{c}(v)$을 갱신할 수 있으며, 이는 수학식 13과 같이 수행될 수 있다.

수학식 13

$$\hat{c}(v) = \begin{cases} z(v), & \text{if } v \text{ is the root} \\ z(v) - s(v) \sum_{u \in ancestor(v)} \frac{\hat{c}(u)}{\sigma^2(u)}, & \text{otherwise} \end{cases}$$

[0083]

[0084] 도 4는 다양한 실시예에 따른 전자 장치 및 서버의 동작 방법을 설명하기 위한 흐름도를 도시한다. 도 4의 실시예는 도 5를 참조하여 더욱 상세하게 설명하도록 한다. 도 5는 다양한 실시예에 따른 전자 장치를 도시한다.

[0085] 다양한 실시예에 따라서, 전자 장치(101)는, 401 동작에서, 신규 텍스트, 예를 들어 제 1 텍스트의 “convfefe”를 입력받을 수 있다. 해당 텍스트는, 서버(108) 또는 전자 장치(101)에 등재되어 있지 않은 단어일 수 있다. 전자 장치(101)는, 제 1 텍스트에 대한 정보를 교란하여 서버(108)로 송신할 수 있다. 더욱 상세하게, 403 동작에서, 서버(108)는, 보고 조건을 전자 장치(101)로 송신할 수 있다. 예를 들어, 서버(108)는, 상술한 트리의 적어도 하나의 노드에 대응하는 사용자를 선택할 수 있으며, 해당 사용자에게 노드에 대응하는 뎁스의 유니 그램을 송신할 것을 요청할 수 있다. 더욱 상세하게, 서버(108)는, 상술한 바와 같이 트리를 형성하는 과정에서, 노드를 처리하는 데에 있어 요구되는 뎁스에 대응하는 데이터 레코드를 송신하여 줄 것을 전자 장치(101)에 요청할 수 있다.

[0086] 405 동작에서, 전자 장치(101)는, 제 1 텍스트 중 지정된 뎁스(depth)의 유니 그램(uni-gram)을 확인할 수 있다. 예를 들어, 전자 장치(101)는, “convfefe” 중 “n”의 유니 그램을 송신 대상의 유니 그램으로 확인할 수 있다. 407 동작에서, 전자 장치(101)는, 확인된 유니 그램을 교란(perturb)할 수 있다. 전자 장치(101)는, “n”을 벡터화하여 유니 그램에 대응하는 벡터를 생성할 수 있으며, 벡터 내의 성분 각각에 대하여 RR을 적용함으로써 교란된 유니 그램을 생성할 수 있다. 409 동작에서, 전자 장치(101)는, 교란된 유니 그램을 서버

(108)로 송신할 수 있다.

[0087] 411 동작에서, 서버(108)는, 수신된 데이터에 기반하여 신규 단어를 도출할 수 있다. 서버(108)는, 상술한 다양한 과정 중 적어도 일부를 통하여, 신규 단어, 즉 “convfefe”를 도출할 수 있다. 413 동작에서, 서버(108)는, 도출된 신규 단어를 전자 장치(101)와 공유할 수 있다. 415 동작에서, 전자 장치(101)는, 신규 단어를 저장할 수 있다. 이후, 도 5에서와 같이, 전자 장치(101)는 표시 장치(160) 상에 메모 어플리케이션의 실행 화면(520)을 표시할 수 있다. 메모 어플리케이션의 실행 화면(520)은, 입력된 텍스트(521) 및 가상 키보드(501)를 포함할 수 있다. 현재, 전자 장치(101)는, “conv”의 텍스트(521)를 입력받은 상태일 수 있으며, 이에 따라 전자 장치(101)는 텍스트(521)를 표시 장치(160) 상에 표시할 수 있다. 전자 장치(101)는, 서버(108)로부터 “convfefe”의 신규 단어를 수신하여 저장하였으며, 현재 입력된 “conv”의 텍스트(521)의 적어도 일부가 저장된 신규 단어인 “convfefe”의 적어도 일부가 대응됨을 확인할 수 있다. 이에 따라, 전자 장치(101)는 신규 단어 “convfefe”를 제 2 추천 단어(512)로서 표시할 수 있다. 전자 장치(101)는, 아울러 사전에 등재된 단어 중 “conv”와 적어도 일부가 대응되는 “convention”의 단어를 제 1 추천 단어(511)로서 표시할 수도 있다.

[0088] 도 6은 다양한 실시예에 따른 전자 장치 및 서버의 동작 방법을 설명하기 위한 흐름도를 도시한다.

[0089] 다양한 실시예에 따라서, 전자 장치(101)는, 601 동작에서, 신규 텍스트, 예를 들어 제 1 텍스트의 “convfefe”를 입력받을 수 있다. 603 동작에서, 서버(108)는, 보고 조건을 전자 장치(101)로 송신할 수 있다. 예를 들어, 서버(108)는, 상술한 트리의 제 1 노드에 대응하는 사용자를 선택할 수 있으며, 해당 사용자에게 제 1 노드에 대응하는 뎁스의 유니 그램을 송신할 것을 요청할 수 있다. 더욱 상세하게, 서버(108)는, 상술한 바와 같이 트리를 형성하는 과정에서, 제 1 노드를 처리하는 데에 있어 요구되는 뎁스에 대응하는 데이터 레코드를 송신하여 줄 것을 전자 장치(101)에 요청할 수 있다.

[0090] 605 동작에서, 전자 장치(101)는, 제 1 텍스트 중 지정된 뎁스(depth)의 유니 그램(uni-gram)을 확인할 수 있다. 예를 들어, 전자 장치(101)는, “convfefe” 중 “n”의 유니 그램을 송신 대상의 유니 그램으로 확인할 수 있다. 607 동작에서, 전자 장치(101)는, 확인된 유니 그램을 교란(perturb)할 수 있다. 609 동작에서, 전자 장치(101)는, 교란된 유니 그램을 서버(108)로 송신할 수 있다.

[0091] 611 동작에서, 서버(108)는, 추가 보고 조건을 송신할 수 있다. 예를 들어, 서버(108)는, 트리를 구성하는 제 2 노드를 처리하는 데에 있어 요구되는 뎁스에 대응하는 데이터 레코드를 송신하여 줄 것을 전자 장치(101)에 요청할 수 있다. 추가 보고 조건은, 알고리즘 4와 관련하여 상술한 바와 같이, 제 2 노드의 부모 노드에 대하여 데이터를 제공하지 않은 전자 장치가 데이터를 제공하는 것일 수 있다. 예를 들어, 전자 장치(101)가 제 2 노드의 부모 노드에 대하여 데이터를 제공하지 않은 것을 상정하도록 한다.

[0092] 613 동작에서, 전자 장치(101)는, 추가 보고 조건의 만족 여부를 판단할 수 있다. 전자 장치(101)가 제 2 노드의 부모 노드에 대하여 데이터를 제공하지 않은 경우에는, 전자 장치(101)는 추가 보고 조건에 대응하는 뎁스의 유니 그램(예: 첫번째 f)를 확인할 수 있다. 615 동작에서, 전자 장치(101)는, 추가 보고 조건에 대응하는 뎁스의 제 2 유니 그램을 확인하여 교란할 수 있다. 617 동작에서, 전자 장치(101)는, 제 2 유니 그램을 송신할 수 있다. 619 동작에서, 서버(108)는, 수신된 데이터에 기반하여 신규 단어를 도출할 수 있다. 621 동작에서, 서버(108)는, 도출된 신규 단어를 전자 장치(101)와 공유할 수 있다. 623 동작에서, 전자 장치(101)는, 공유된 신규 단어를 저장할 수 있다. 한편, 상기 제 1 교란된 유니 그램을 획득하는 과정에서, 제 1 크기의 제 1 프라이버시 버짓(privacy budget)이 소요될 수 있으며, 상기 제 2 교란된 유니 그램을 획득하는 과정에서, 제 1 크기의 제 2 프라이버시 버짓(privacy budget)을 소요될 수 있다. 하지만, 전자 장치(101)는, 제 1 노드 및 제 2 노드에 대하여 독립적으로 데이터 레코드를 송신하기 때문에, 제 1 프라이버시 버짓 및 제 2 프라이버시 버짓 또한 서로에 대하여 독립적일 수 있으며, 각각의 교란 과정에서 상대적으로 큰 크기의 프라이버시 버짓이 소요될 수 있어, 정확도가 향상될 수 있다.

[0093] 본 문서에 개시된 다양한 실시예들에 따른 전자 장치는 다양한 형태의 장치가 될 수 있다. 전자 장치는, 예를 들면, 휴대용 통신 장치 (예: 스마트폰), 컴퓨터 장치, 휴대용 멀티미디어 장치, 휴대용 의료 기기, 카메라, 웨어러블 장치, 또는 가전 장치 중 적어도 하나를 포함할 수 있다. 본 문서의 실시예에 따른 전자 장치는 전술한 기기들에 한정되지 않는다.

[0094] 본 문서의 다양한 실시예들 및 이에 사용된 용어들은 본 문서에 기재된 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 해당 실시예의 다양한 변경, 균등물, 및/또는 대체물을 포함하는 것으로 이해되어야 한다. 도면의 설명과 관련하여, 유사한 구성요소에 대해서는 유사한 참조 부호가 사용될 수 있다. 단수의 표현은 문맥

상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함할 수 있다. 본 문서에서, "A 또는 B", "A 및/또는 B 중 적어도 하나", "A, B 또는 C" 또는 "A, B 및/또는 C 중 적어도 하나" 등의 표현은 함께 나열된 항목들의 모든 가능한 조합을 포함할 수 있다. "제 1", "제 2", "첫째" 또는 "둘째" 등의 표현들은 해당 구성요소들을, 순서 또는 중요도에 상관없이 수식할 수 있고, 한 구성요소를 다른 구성요소와 구분하기 위해 사용될 뿐 해당 구성요소들을 한정하지 않는다. 어떤(예: 제 1) 구성요소가 다른(예: 제 2) 구성요소에 "(기능적으로 또는 통신적으로) 연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 상기 어떤 구성요소가 상기 다른 구성요소에 직접적으로 연결되거나, 다른 구성요소(예: 제 3 구성요소)를 통하여 연결될 수 있다.

[0095] 본 문서에서 사용된 용어 "모듈"은 하드웨어, 소프트웨어 또는 펌웨어로 구성된 유닛을 포함하며, 예를 들면, 로직, 논리 블록, 부품, 또는 회로 등의 용어와 상호 호환적으로 사용될 수 있다. 모듈은, 일체로 구성된 부품 또는 하나 또는 그 이상의 기능을 수행하는 최소 단위 또는 그 일부가 될 수 있다. 예를 들면, 모듈은 ASIC(application-specific integrated circuit)으로 구성될 수 있다.

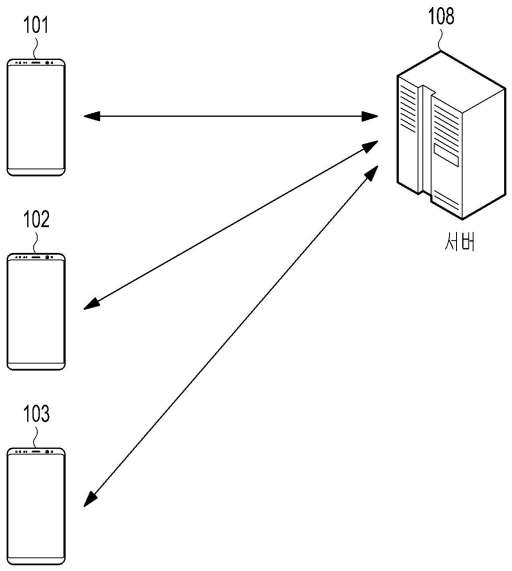
[0096] 본 문서의 다양한 실시예들은 기기(machine)(예: 컴퓨터)로 읽을 수 있는 저장 매체(machine-readable storage media)(예: 내장 메모리(136) 또는 외장 메모리(138))에 저장된 명령어를 포함하는 소프트웨어(예: 프로그램(140))로 구현될 수 있다. 기기는, 저장 매체로부터 저장된 명령어를 호출하고, 호출된 명령어에 따라 동작이 가능한 장치로서, 개시된 실시예들에 따른 전자 장치(예: 전자 장치(101))를 포함할 수 있다. 상기 명령이 프로세서(예: 프로세서(120))에 의해 실행될 경우, 프로세서가 직접, 또는 상기 프로세서의 제어하에 다른 구성요소들을 이용하여 상기 명령에 해당하는 기능을 수행할 수 있다. 명령은 컴파일러 또는 인터프리터에 의해 생성 또는 실행되는 코드를 포함할 수 있다. 기기로 읽을 수 있는 저장매체는, 비일시적(non-transitory) 저장매체의 형태로 제공될 수 있다. 여기서, '비일시적' 은 저장매체가 신호(signal)를 포함하지 않으며 실재(tangible)한다는 것을 의미할 뿐 데이터가 저장매체에 반영구적 또는 임시적으로 저장됨을 구분하지 않는다.

[0097] 일시예에 따르면, 본 문서에 개시된 다양한 실시예들에 따른 방법은 컴퓨터 프로그램 제품(computer program product)에 포함되어 제공될 수 있다. 컴퓨터 프로그램 제품은 상품으로서 판매자 및 구매자 간에 거래될 수 있다. 컴퓨터 프로그램 제품은 기기로 읽을 수 있는 저장 매체(예: compact disc read only memory (CD-ROM))의 형태로, 또는 어플리케이션 스토어(예: 플레이 스토어™)를 통해 온라인으로 배포될 수 있다. 온라인 배포의 경우에, 컴퓨터 프로그램 제품의 적어도 일부는 제조사의 서버, 어플리케이션 스토어의 서버, 또는 중계 서버의 메모리와 같은 저장 매체에 적어도 일시 저장되거나, 임시적으로 생성될 수 있다.

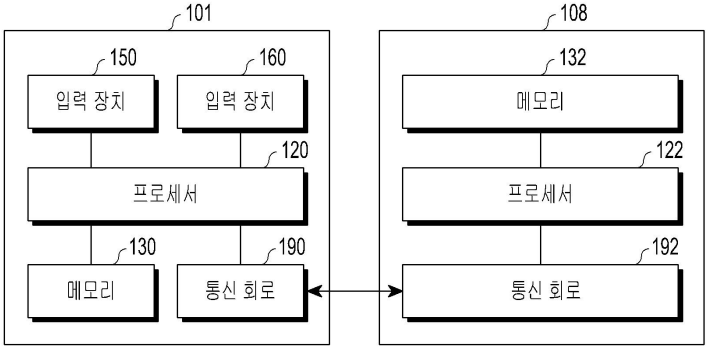
[0098] 다양한 실시예들에 따른 구성 요소(예: 모듈 또는 프로그램) 각각은 단수 또는 복수의 개체로 구성될 수 있으며, 전술한 해당 서브 구성 요소들 중 일부 서브 구성 요소가 생략되거나, 또는 다른 서브 구성 요소가 다양한 실시예에 더 포함될 수 있다. 대체적으로 또는 추가적으로, 일부 구성 요소들(예: 모듈 또는 프로그램)은 하나의 개체로 통합되어, 통합되기 이전의 각각의 해당 구성 요소에 의해 수행되는 기능을 동일 또는 유사하게 수행할 수 있다. 다양한 실시예들에 따른, 모듈, 프로그램 또는 다른 구성 요소에 의해 수행되는 동작들은 순차적, 병렬적, 반복적 또는 휴리스틱하게 실행되거나, 적어도 일부 동작이 다른 순서로 실행되거나, 생략되거나, 또는 다른 동작이 추가될 수 있다.

도면

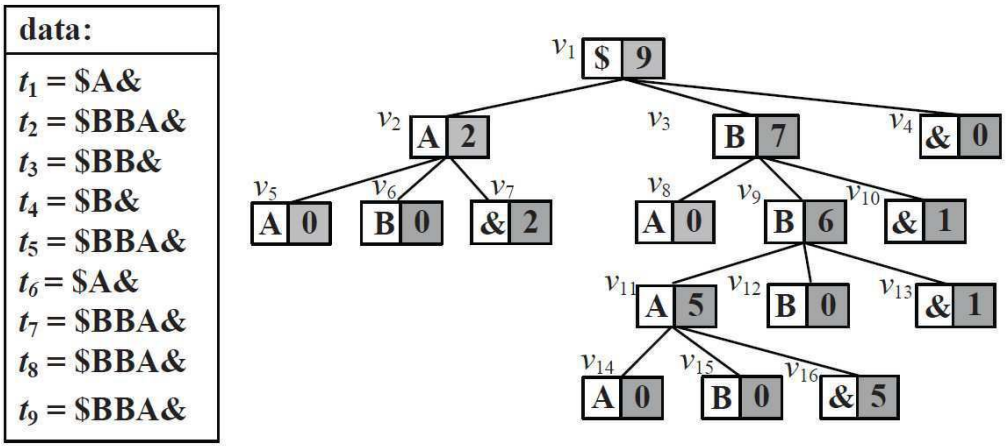
도면1


101
102
103
108
서버

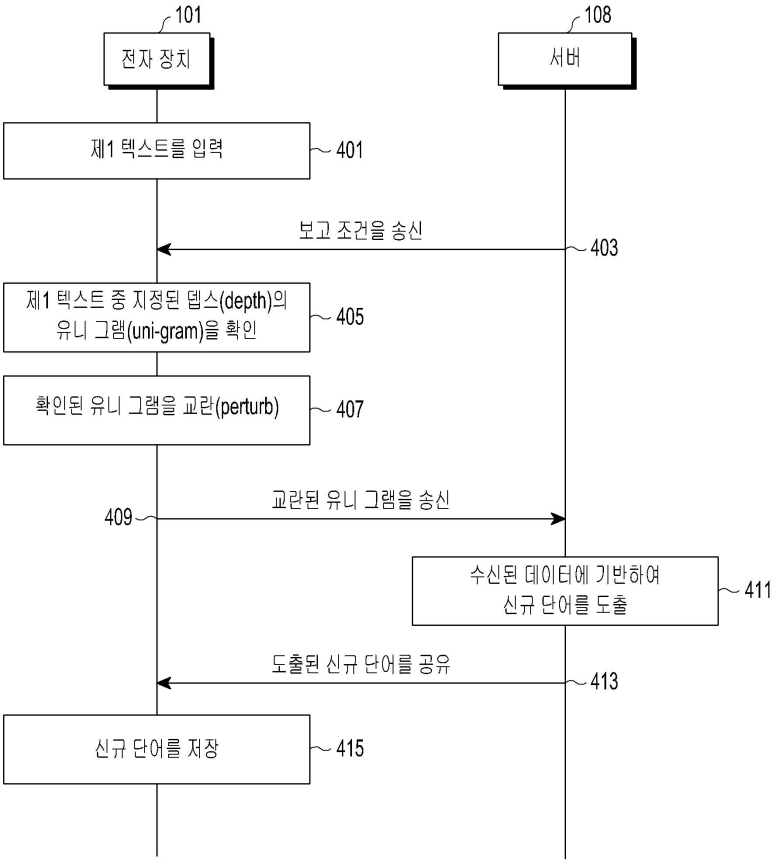
도면2


101
150
입력 장치
160
입력 장치
120
프로세서
130
메모리
190
통신 회로
108
132
메모리
122
프로세서
192
통신 회로

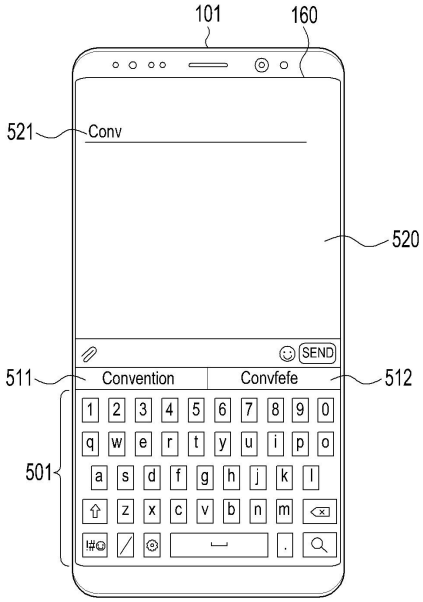
도면3


data:
t_1 = $A&
t_2 = $BBA&
t_3 = $BB&
t_4 = $B&
t_5 = $BBA&
t_6 = $A&
t_7 = $BBA&
t_8 = $BBA&
t_9 = $BBA&
v_1 $ 9
v_2 A 2
v_3 B 7
v_4 & 0
v_5 A 0
v_6 B 0
v_7 & 2
v_8 A 0
v_9 B 6
v_{10} & 1
v_{11} A 5
v_{12} B 0
v_{13} & 1
v_{14} A 0
v_{15} B 0
v_{16} & 5

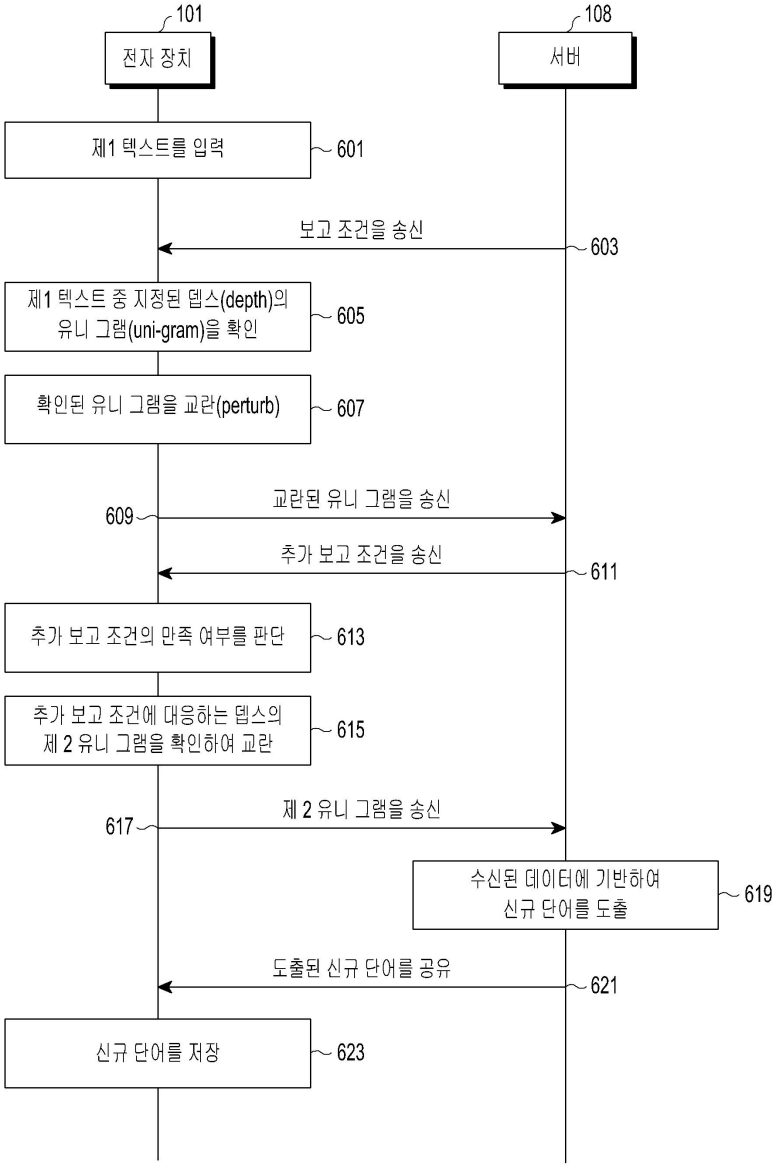
도면4


101
전자 장치
108
서버
제1 텍스트를 입력
401
보고 조건을 송신
403
제1 텍스트 중 지정된 뎁스(depth)의 유니 그램(uni-gram)을 확인
405
확인된 유니 그램을 교란(perturb)
407
409
교란된 유니 그램을 송신
수신된 데이터에 기반하여 신규 단어를 도출
411
도출된 신규 단어를 공유
413
신규 단어를 저장
415

도면5


101
160
521
Conv
520
SEND
511
Convention
Convfefe
512
501
1 2 3 4 5 6 7 8 9 0
q w e r t y u i p o
a s d f g h j k l
z x c v b n m

도면6


101
전자 장치
108
서버
제1 텍스트를 입력
601
보고 조건을 송신
603
제1 텍스트 중 지정된 뎁스(depth)의 유니 그램(uni-gram)을 확인
605
확인된 유니 그램을 교란(perturb)
607
609
교란된 유니 그램을 송신
추가 보고 조건을 송신
611
추가 보고 조건의 만족 여부를 판단
613
추가 보고 조건에 대응하는 뎁스의 제 2 유니 그램을 확인하여 교란
615
617
제 2 유니 그램을 송신
수신된 데이터에 기반하여 신규 단어를 도출
619
도출된 신규 단어를 공유
621
신규 단어를 저장
623