(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0146061
(43) 공개일자 2022년11월01일

(51) 국제특허분류(Int. Cl.)
G06F 11/36 (2006.01) *G06N 3/04* (2006.01)
G06N 3/08 (2006.01)
(52) CPC특허분류
G06F 11/362 (2013.01)
G06F 11/3664 (2013.01)
(21) 출원번호 10-2021-0052882
(22) 출원일자 2021년04월23일
심사청구일자 2021년04월23일

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
한요섭
서울특별시 은평구 진관1로 77-8, 403동 204호 (진관동,은평뉴타운폭포동아파트)
서현태
경기도 안양시 만안구 박달로 403, 101동 1502호 (박달동, 한일유앤아이아파트)
한중혁
서울특별시 마포구 토정로18길 11, 102동 1604호 (현석동, 래미안웰스트림)
(74) 대리인
민영준

전체 청구항 수 : 총 15 항

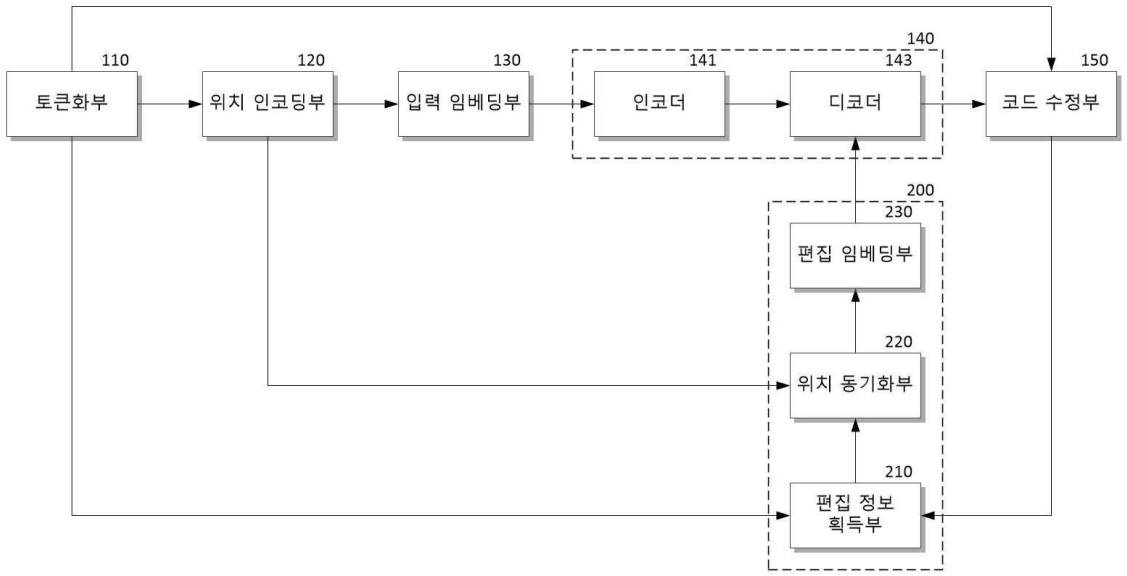
(54) 발명의 명칭 **다중 오류 프로그램 소스 코드 자동 수정 장치 및 방법**

(57) 요 약

본 발명은 입력 프로그램을 인가받아 최소 구성 요소 단위로 구분하고, 구분된 각 구성 요소를 기지정된 방식으로 토큰화하여 다수의 토큰을 포함하는 토큰 테이블을 획득하는 토큰화부, 토큰 테이블의 다수의 토큰 각각의 배치 순서에 따른 위치 정보를 생성하여 맵핑하는 위치 인코딩부, 다수의 토큰 각각과 대응하는 위치 정보 각각을

(뒷면에 계속)

대 표 도



벡터화하고 결합하여 다수의 입력 벡터를 획득하는 입력 임베딩부 및 미리 학습된 인공 신경망으로 구현되어, 다수의 입력 벡터를 인가받아 학습된 방식에 따라 인코딩 및 디코딩하여 입력 프로그램의 다수의 토큰 중 오류 토큰을 수정하기 위한 다수의 수정 명령이 각 토큰의 위치에 대응하도록 출력하는 신경망을 포함하되, 신경망은 학습 시에 입력 프로그램으로서 인가되는 학습 프로그램과 학습 프로그램의 오류가 수정된 진리 프로그램으로 편집하기 위한 편집 정보 및 학습 프로그램과 진리 프로그램에서 서로 대응하는 토큰들 사이의 위치 정보를 동기화하는 동기 위치 정보 각각을 벡터화하고 결합된 업데이트 벡터에 의해 학습되어 컴파일러나 사용자의 개입 없이 다수의 오류가 포함된 프로그램 소스 코드에서 각 오류가 수정될 위치 정보를 고려하여 수정 명령을 생성할 수 있어, 한번의 신경망 연산으로 다수의 오류를 정확하게 수정할 수 있는 프로그램 소스 코드 자동 수정 장치 및 방법을 제공한다.

(52) CPC특허분류

G06N 3/04 (2013.01)
G06N 3/08 (2013.01)
Z01A 10/00 (2019.06)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711119683
과제번호	2020R1A4A3079947
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	기초연구실육성사업
연구과제명	휴먼-AI 협업 프로그래밍 플랫폼 기술 연구실
기 여 율	1/2
과제수행기관명	연세대학교 산학협력단
연구기간	2020.07.01 ~ 2021.05.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126082
과제번호	2020-0-01361-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원(한국연구재단부설)
연구사업명	정보통신방송연구개발사업
연구과제명	인공지능대학원지원사업[1단계]
기 여 율	1/2
과제수행기관명	연세대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

입력 프로그램을 인가받아 최소 구성 요소 단위로 구분하고, 구분된 각 구성 요소를 기지정된 방식으로 토큰화하여 다수의 토큰을 포함하는 토큰 테이블을 획득하는 토큰화부;

상기 토큰 테이블의 다수의 토큰 각각의 배치 순서에 따른 위치 정보를 생성하여 맵핑하는 위치 인코딩부;

상기 다수의 토큰 각각과 대응하는 위치 정보 각각을 벡터화하고 결합하여 다수의 입력 벡터를 획득하는 입력 임베딩부; 및

미리 학습된 인공 신경망으로 구현되어, 상기 다수의 입력 벡터를 인가받아 학습된 방식에 따라 인코딩 및 디코딩하여 상기 입력 프로그램의 다수의 토큰 중 오류 토큰을 수정하기 위한 다수의 수정 명령이 각 토큰의 위치에 대응하도록 출력하는 신경망을 포함하되,

상기 신경망은 학습 시에 입력 프로그램으로서 인가되는 학습 프로그램과 상기 학습 프로그램의 오류가 수정된 진리 프로그램으로 편집하기 위한 편집 정보 및 상기 학습 프로그램과 상기 진리 프로그램에서 서로 대응하는 토큰들 사이의 위치 정보를 동기화하는 동기 위치 정보 각각을 벡터화하고 결합된 업데이트 벡터에 의해 학습되는 프로그램 소스 코드 자동 수정 장치.

청구항 2

제1항에 있어서, 상기 신경망은

순차 연결된 다수의 인코딩 셀을 포함하여, 상기 다수의 입력 벡터를 인가받아 미리 학습된 방식에 따라 인코딩하여 다수의 토큰 각각에 대응하는 다수의 히든 벡터를 추출하는 인코더; 및

순차 연결된 다수의 디코딩 셀을 포함하여, 상기 다수의 인코딩 셀 각각에서 추출된 히든 벡터를 인가받아 미리 학습된 방식에 따라 디코딩하여 상기 다수의 토큰 각각의 오류를 수정하기 위한 다수의 수정 명령을 각 토큰의 위치에 대응하여 출력하는 디코더를 포함하는 프로그램 소스 코드 자동 수정 장치.

청구항 3

제2항에 있어서, 상기 다수의 인코딩 셀 각각은

다수의 입력 벡터 중 대응하는 입력 벡터와 이전 배치된 인코딩 셀에서 출력되는 히든 벡터를 인가받아 학습된 방식에 따라 인코딩하여 특징 벡터와 히든 벡터를 출력하는 프로그램 소스 코드 자동 수정 장치.

청구항 4

제2항에 있어서, 상기 다수의 디코딩 셀 각각은

이전 배치된 디코딩 셀에서 추출된 히든 벡터를 인가받고, 상기 다수의 인코딩 셀 각각에서 추출된 히든 벡터와 함께 디코딩하여 상기 다수의 수정 명령을 획득하는 프로그램 소스 코드 자동 수정 장치.

청구항 5

제4항에 있어서, 상기 신경망은

상기 다수의 인코딩 셀에서 추출된 다수의 히든 벡터와 각 디코딩 셀에서 추출된 히든 벡터를 각각 내적하고 정규화하여 다수의 디코딩 셀 각각에 대응하는 다수의 어텐션 스코어를 획득하고, 획득된 다수의 어텐션 스코어를 상기 다수의 인코딩 셀에서 추출된 다수의 히든 벡터에 가중하여 대응하는 디코딩 셀로 인가하는 어텐션부를 더 포함하는 프로그램 소스 코드 자동 수정 장치.

청구항 6

제5항에 있어서, 상기 신경망은

상기 학습 프로그램과 상기 진리 프로그램의 각 토큰들 사이의 차이를 편집 거리 알고리즘에 따라 비교하여, 최소의 편집으로 상기 학습 프로그램이 상기 진리 프로그램으로 변환되도록 하기 위한 수정 명령으로 구성되는 다수의 편집 정보와 상기 학습 프로그램과 상기 진리 프로그램에서 서로 대응하는 토큰들 사이의 위치가 동일해지도록 동기화하는 다수의 동기 위치 정보가 획득되어 벡터화되고, 벡터화된 다수의 편집 정보와 대응하는 동기 위치 정보가 결합된 다수의 업데이트 벡터가 상기 다수의 디코딩 셀 중 대응하는 디코딩 셀에 인가되어 학습되는 프로그램 소스 코드 자동 수정 장치.

청구항 7

제1항에 있어서, 상기 프로그램 소스 코드 자동 수정 장치는

상기 신경망에서 출력되는 다수의 수정 명령에 따라 상기 입력 프로그램의 오류를 수정하고, 역토큰화하여 오류가 제거된 수정 프로그램을 출력하는 코드 수정부를 더 포함하는 프로그램 소스 코드 자동 수정 장치.

청구항 8

입력 프로그램을 인가받아 최소 구성 요소 단위로 구분하고, 구분된 각 구성 요소를 기지정된 방식으로 토큰화하여 다수의 토큰을 포함하는 토큰 테이블을 획득하는 단계;

상기 토큰 테이블의 다수의 토큰 각각의 배치 순서에 따른 위치 정보를 생성하여 맵핑하는 단계;

상기 다수의 토큰 각각과 대응하는 위치 정보 각각을 벡터화하고 결합하여 다수의 입력 벡터를 획득하는 단계; 및

미리 학습된 인공 신경망에 상기 다수의 입력 벡터를 입력하여, 학습된 방식에 따라 상기 입력 벡터를 인코딩 및 디코딩하여 상기 입력 프로그램의 다수의 토큰 중 오류 토큰을 수정하기 위한 다수의 수정 명령이 각 토큰의 위치에 대응하도록 출력하는 단계를 포함하되,

상기 신경망은 학습 시에 입력 프로그램으로서 인가되는 학습 프로그램과 상기 학습 프로그램의 오류가 수정된 진리 프로그램으로 편집하기 위한 편집 정보 및 상기 학습 프로그램과 상기 진리 프로그램에서 서로 대응하는 토큰들 사이의 위치 정보를 동기화하는 동기 위치 정보 각각을 벡터화하고 결합된 업데이트 벡터에 의해 학습되는 프로그램 소스 코드 자동 수정 방법.

청구항 9

제8항에 있어서, 상기 각 토큰의 위치에 대응하도록 출력하는 단계는

순차 연결된 다수의 인코딩 셀이 포함된 상기 신경망의 인코더가 상기 다수의 입력 벡터를 인가받아 미리 학습된 방식에 따라 인코딩하여 다수의 토큰 각각에 대응하는 다수의 히든 벡터를 추출하는 단계; 및

순차 연결된 다수의 디코딩 셀을 포함하는 상기 신경망의 디코더가 상기 다수의 인코딩 셀 각각에서 추출된 히든 벡터를 인가받아 미리 학습된 방식에 따라 디코딩하여 상기 다수의 토큰 각각의 오류를 수정하기 위한 다수의 수정 명령을 각 토큰의 위치에 대응하여 출력하는 단계를 포함하는 프로그램 소스 코드 자동 수정 방법.

청구항 10

제9항에 있어서, 상기 다수의 인코딩 셀 각각은

다수의 입력 벡터 중 대응하는 입력 벡터와 이전 배치된 인코딩 셀에서 출력되는 히든 벡터를 인가받아 학습된 방식에 따라 인코딩하여 특징 벡터와 히든 벡터를 출력하는 프로그램 소스 코드 자동 수정 방법.

청구항 11

제10항에 있어서, 상기 다수의 디코딩 셀 각각은

이전 배치된 디코딩 셀에서 추출된 히든 벡터를 인가받고, 상기 다수의 인코딩 셀 각각에서 추출된 히든 벡터와 함께 디코딩하여 상기 다수의 수정 명령을 획득하는 프로그램 소스 코드 자동 수정 방법.

청구항 12

제11항에 있어서, 상기 각 토큰의 위치에 대응하도록 출력하는 단계는

상기 다수의 히든 벡터를 추출하는 단계 이후, 상기 다수의 인코딩 셀에서 추출된 다수의 히든 벡터와 각 디코딩 셀에서 추출된 히든 벡터를 각각 내적하고 정규화하여 다수의 디코딩 셀 각각에 대응하는 다수의 어텐션 스코어를 획득하는 단계; 및

획득된 다수의 어텐션 스코어를 상기 다수의 인코딩 셀에서 추출된 다수의 히든 벡터에 가중하여 대응하는 디코딩 셀로 인가하는 단계를 더 포함하는 프로그램 소스 코드 자동 수정 방법.

청구항 13

제12항에 있어서, 상기 신경망은

상기 학습 프로그램과 상기 진리 프로그램의 각 토큰들 사이의 차이를 편집 거리 알고리즘에 따라 비교하여, 최소의 편집으로 상기 학습 프로그램이 상기 진리 프로그램으로 변환되도록 하기 위한 수정 명령으로 구성되는 다수의 편집 정보와 상기 학습 프로그램과 상기 진리 프로그램에서 서로 대응하는 토큰들 사이의 위치가 동일해지도록 동기화하는 다수의 동기 위치 정보가 획득되어 벡터화되고, 벡터화된 다수의 편집 정보와 대응하는 동기 위치 정보가 결합된 다수의 업데이트 벡터가 상기 다수의 디코딩 셀 중 대응하는 디코딩 셀에 인가되어 학습되는 프로그램 소스 코드 자동 수정 방법.

청구항 14

제8항에 있어서, 상기 프로그램 소스 코드 자동 수정 방법은

상기 신경망에서 출력되는 다수의 수정 명령에 따라 상기 입력 프로그램의 오류를 수정하고, 역토큰화하여 오류가 제거된 수정 프로그램을 출력하는 단계를 더 포함하는 프로그램 소스 코드 자동 수정 방법.

청구항 15

제8항 내지 제14항 중 어느 한에 있어서, 상기 프로그램 소스 코드 자동 수정 방법을 수행하기 위한 컴퓨팅 장치에서 판독 가능한 프로그램 명령어가 기록된 기록 매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 프로그램 소스 코드 자동 수정 장치 및 방법에 관한 것으로, 다중 오류 프로그램 소스 코드 자동 수정 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 종래에 프로그램의 소스 코드에서 문법적인 오류는 컴파일러가 프로그램 소스 코드를 컴파일링하는 과정에서 검출되었다. 그러나 컴파일러는 변수 선언 누락이나 오타와 같은 오류를 찾아서 알려줄 뿐, 오류를 자동으로 수정하지 못하는 한계가 있었다. 따라서 프로그래머, 즉 사용자의 직접적인 개입이 요구되었다.

[0003] 이에 프로그램 소스 코드를 자동으로 수정할 수 있도록 하는 다양한 연구가 진행되고 있다. 초기에 오류를 자동으로 수정하는 기법으로는 규칙 기반 알고리즘이 제안되었으나, 규칙 기반 알고리즘에서는 고려해야하는 사례가 너무 많아 유용한 수정 프로그램을 개발하기 어려울 뿐만 아니라 그 성능에 제약이 많았다. 이러한 한계를 극복하고자 최근에는 인공 신경망을 이용하여 프로그램 소스 코드의 오류를 자동으로 수정하고자 하는 연구가 수행되고 있다. 다만 기존 연구에서는 소스 코드의 오류를 자동으로 수정하기 위해 인공 신경망을 이용하더라도, 한번의 신경망 연산에서 수정 가능한 오류는 프로그램 소스 코드의 한 라인 이내의 오류 또는 가장 우선 검출되는 하나의 오류만을 수정할 수 있다는 한계가 있다. 따라서 다수의 오류가 포함된 프로그램 소스 코드 전체를 자동 수정하기 위해서는 프로그램 소스 코드의 길이와 오류의 개수에 대응하는 횟수로 반복적으로 신경망 연산이 수행되어야 한다는 문제가 있다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 한국 등록 특허 제10-1850303호 (2018.04.13 등록)

발명의 내용

해결하려는 과제

[0005] 본 발명의 목적은 다수의 오류가 포함된 프로그램 소스 코드를 자동으로 수정할 수 있는 프로그램 소스 코드 자동 수정 장치 및 방법을 제공하는데 있다.

[0006] 본 발명의 다른 목적은 프로그램 소스 코드에 다수의 오류가 포함된 경우에도 한 번에 정확하게 수정할 수 있는 프로그램 소스 코드 자동 수정 장치 및 방법을 제공하는데 있다.

과제의 해결 수단

[0007] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 프로그램 소스 코드 자동 수정 장치는 입력 프로그램을 인가받아 최소 구성 요소 단위로 구분하고, 구분된 각 구성 요소를 기지정된 방식으로 토큰화하여 다수의 토큰을 포함하는 토큰 테이블을 획득하는 토큰화부; 상기 토큰 테이블의 다수의 토큰 각각의 배치 순서에 따른 위치 정보를 생성하여 맵핑하는 위치 인코딩부; 상기 다수의 토큰 각각과 대응하는 위치 정보 각각을 벡터화하고 결합하여 다수의 입력 벡터를 획득하는 입력 임베딩부; 및 미리 학습된 인공 신경망으로 구현되어, 상기 다수의 입력 벡터를 인가받아 학습된 방식에 따라 인코딩 및 디코딩하여 상기 입력 프로그램의 다수의 토큰 중 오류 토큰을 수정하기 위한 다수의 수정 명령이 각 토큰의 위치에 대응하도록 출력하는 신경망을 포함하되, 상기 신경망은 학습 시에 입력 프로그램으로서 인가되는 학습 프로그램과 상기 학습 프로그램의 오류가 수정된 진리 프로그램으로 편집하기 위한 편집 정보 및 상기 학습 프로그램과 상기 진리 프로그램에서 서로 대응하는 토큰들 사이의 위치 정보를 동기화하는 동기 위치 정보 각각을 벡터화하고 결합된 업데이트 벡터에 의해 학습될 수 있다.

[0008] 상기 신경망은 순차 연결된 다수의 인코딩 셀을 포함하여, 상기 다수의 입력 벡터를 인가받아 미리 학습된 방식에 따라 인코딩하여 다수의 토큰 각각에 대응하는 다수의 히든 벡터를 추출하는 인코더; 및 순차 연결된 다수의 디코딩 셀을 포함하여, 상기 다수의 인코딩 셀 각각에서 추출된 히든 벡터를 인가받아 미리 학습된 방식에 따라 디코딩하여 상기 다수의 토큰 각각의 오류를 수정하기 위한 다수의 수정 명령을 각 토큰의 위치에 대응하여 출력하는 디코더를 포함할 수 있다.

[0009] 상기 다수의 인코딩 셀 각각은 다수의 입력 벡터 중 대응하는 입력 벡터와 이전 배치된 인코딩 셀에서 출력되는 히든 벡터를 인가받아 학습된 방식에 따라 인코딩하여 특징 벡터와 히든 벡터를 출력할 수 있다.

[0010] 상기 다수의 디코딩 셀 각각은 이전 배치된 디코딩 셀에서 추출된 히든 벡터를 인가받고, 상기 다수의 인코딩 셀 각각에서 추출된 히든 벡터와 함께 디코딩하여 상기 다수의 수정 명령을 획득할 수 있다.

[0011] 상기 신경망은 상기 다수의 인코딩 셀에서 추출된 다수의 히든 벡터와 각 디코딩 셀에서 추출된 히든 벡터를 각각 내적하고 정규화하여 다수의 디코딩 셀 각각에 대응하는 다수의 어텐션 스코어를 획득하고, 획득된 다수의 어텐션 스코어를 상기 다수의 인코딩 셀에서 추출된 다수의 히든 벡터에 가중하여 대응하는 디코딩 셀로 인가하는 어텐션부를 더 포함할 수 있다.

[0012] 상기 신경망은 상기 학습 프로그램과 상기 진리 프로그램의 각 토큰들 사이의 차이를 편집 거리 알고리즘에 따라 비교하여, 최소의 편집으로 상기 학습 프로그램이 상기 진리 프로그램으로 변환되도록 하기 위한 수정 명령으로 구성되는 다수의 편집 정보와 상기 학습 프로그램과 상기 진리 프로그램에서 서로 대응하는 토큰들 사이의 위치가 동일해지도록 동기화하는 다수의 동기 위치 정보가 획득되어 벡터화되고, 벡터화된 다수의 편집 정보와 대응하는 동기 위치 정보가 결합된 다수의 업데이트 벡터가 상기 다수의 디코딩 셀 중 대응하는 디코딩 셀에 인가되어 학습될 수 있다.

[0013] 상기 프로그램 소스 코드 자동 수정 장치는 상기 신경망에서 출력되는 다수의 수정 명령에 따라 상기 입력 프로그램의 오류를 수정하고, 역토큰화하여 오류가 제거된 수정 프로그램을 출력하는 코드 수정부를 더 포함할 수 있다.

[0014] 상기 목적을 달성하기 위한 본 발명의 다른 실시예에 따른 프로그램 소스 코드 자동 수정 방법은 입력 프로그램을 인가받아 최소 구성 요소 단위로 구분하고, 구분된 각 구성 요소를 기지정된 방식으로 토큰화하여 다수의 토큰을 포함하는 토큰 테이블을 획득하는 단계; 상기 토큰 테이블의 다수의 토큰 각각의 배치 순서에 따른 위치

정보를 생성하여 맵핑하는 단계; 상기 다수의 토큰 각각과 대응하는 위치 정보 각각을 벡터화하고 결합하여 다수의 입력 벡터를 획득하는 단계; 및 미리 학습된 인공 신경망에 상기 다수의 입력 벡터를 입력하여, 학습된 방식에 따라 상기 입력 벡터를 인코딩 및 디코딩하여 상기 입력 프로그램의 다수의 토큰 중 오류 토큰을 수정하기 위한 다수의 수정 명령이 각 토큰의 위치에 대응하도록 출력하는 단계를 포함하되, 상기 신경망은 학습 시에 입력 프로그램으로서 인가되는 학습 프로그램과 상기 학습 프로그램의 오류가 수정된 진리 프로그램으로 편집하기 위한 편집 정보 및 상기 학습 프로그램과 상기 진리 프로그램에서 서로 대응하는 토큰들 사이의 위치 정보를 동기화하는 동기 위치 정보 각각을 벡터화하고 결합된 업데이트 벡터에 의해 학습될 수 있다.

발명의 효과

[0015] 따라서, 본 발명의 실시예에 따른 프로그램 소스 코드 자동 수정 장치 및 방법은 컴파일러나 사용자의 개입없이 다수의 오류가 포함된 프로그램 소스 코드에서 각 오류가 수정될 위치 정보를 고려하여 수정 명령을 생성할 수 있어, 한번의 신경망 연산으로 다수의 오류를 정확하게 수정할 수 있다. 그러므로 빠르게 프로그램 소스 코드의 오류를 수정할 수 있어 작업 효율성을 개선할 수 있다.

도면의 간단한 설명

[0016] 도 1은 본 발명의 일 실시예에 따른 프로그램 소스 코드 자동 수정 장치의 개략적 구성을 나타낸다.

도 2는 프로그램 소스 코드 자동 수정 장치의 동작을 설명하기 위한 도면이다.

도 3은 도 1의 토큰화부에 의해 토큰화된 입력 프로그램의 일 예를 나타낸다.

도 4는 도 1의 위치 인코딩부에 의해 위치 정보가 설정된 토큰의 일 예를 나타낸다.

도 5는 학습 시에 도 1의 편집 정보 획득부에 의해 획득되는 수정 명령의 일 예를 나타낸다.

도 6은 도 1의 위치 동기화부가 도 5의 수정 명령에 따라 수정되는 입력 프로그램의 위치 정보를 동기화하기 위해 획득하는 동기 위치의 일 예를 나타낸다.

도 7은 본 발명의 일 실시예에 따른 프로그램 소스 코드 자동 수정 방법을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0017] 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.

[0018] 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 그러나, 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 설명하는 실시예에 한정되는 것이 아니다. 그리고, 본 발명을 명확하게 설명하기 위하여 설명과 관계없는 부분은 생략되며, 도면의 동일한 참조부호는 동일한 부재임을 나타낸다.

[0019] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다. 또한, 명세서에 기재된 "...부", "...기", "모듈", "블록" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0020] 도 1은 본 발명의 일 실시예에 따른 프로그램 소스 코드 자동 수정 장치의 개략적 구성을 나타내고, 도 2는 프로그램 소스 코드 자동 수정 장치의 동작을 설명하기 위한 도면이며, 도 3은 도 1의 토큰화부에 의해 토큰화된 입력 프로그램의 일 예를 나타낸다. 그리고 도 4는 도 1의 위치 인코딩부에 의해 위치 정보가 설정된 토큰의 일 예를 나타내고, 도 5는 학습 시에 도 1의 편집 정보 획득부에 의해 획득되는 수정 명령의 일 예를 나타내며, 도 6은 도 1의 위치 동기화부가 도 5의 수정 명령에 따라 수정되는 입력 프로그램의 위치 정보를 동기화하기 위해 획득하는 동기 위치의 일 예를 나타낸다.

[0021] 본 실시예에서 프로그램 소스 코드 자동 수정 장치는 입력 프로그램을 인가받아, 입력 프로그램에 오류가 존재하는지 여부를 검출하여 검출된 오류를 자동으로 수정함으로써 수정 프로그램을 출력할 수 있다. 여기서는 설명의 편의를 위하여 컴파일되기 이전의 프로그램 소스 코드를 간략하게 프로그램이라 할 수 있으며, 따라서 본 실시예에서 입력 프로그램과 입력 프로그램 소스 코드는 동일한 의미를 가지며, 수정 프로그램과 수정 프로그램 소스 코드는 동일한 의미를 갖는다.

[0022] 도 1을 참조하면, 본 실시예에 따른 프로그램 소스 코드 자동 수정 장치는 토큰화부(110), 위치 인코딩부(120), 입력 임베딩부(130), 신경망(140) 및 코드 수정부(150)를 포함한다.

[0023] 우선 토큰화부(110)는 입력 프로그램을 인가받고, 인가된 입력 프로그램을 기지정된 방식에 따라 토큰화한다.

[0024] 도 2의 오른쪽 하단에 도시된 바와 같이, 프로그램 소스 코드는 프로그램 언어에 따른 작성 기법에 의해 지정된 규칙을 따라 라이브러리, 변수, 변수 유형, 키워드, 특수 문자(예를 들면, 괄호, 세미 콜론), 함수 및 리터럴(literal)과 같은 기지정된 양식의 구성 요소로 작성된다. 이에 토큰화부(110)는 입력 프로그램의 텍스트를 프로그램에서 지정된 최소 구성 요소 단위로 구분하여 지정된 토큰에 맵핑시킴으로써 입력 프로그램을 토큰화할 수 있다.

[0025] 일 예로 도 3에 도시된 바와 같이, 입력 프로그램의 텍스트는 대응하는 프로그램 언어의 작정 기법의 양식에 따라 최소 구성 요소로 지정된 다양한 종류의 토큰에 맵핑되어 구분될 수 있다. 도 3을 참조하면, 입력 프로그램에서 "#include" 텍스트는 "_<directive>_#include" 토큰과 토큰값으로 맵핑되고, "char" 텍스트는 "_<type>_char" 토큰과 토큰값으로 맵핑될 수 있다. 다만, "Hello World"나 "0"과 같은 리터럴에 해당하는 텍스트, 즉 토큰값이 컴파일 오류를 유발하는 요인이 아니다. 따라서 리터럴에 해당하는 텍스트는 미리 지정된 특수 토큰에 맵핑될 수 있다. 일 예로 도 3에서 문자열 리터럴인 "Hello World"는 토큰값이 생략된 "_<STR>_" 토큰에 맵핑되고, 숫자 리터럴인 "0" 또한 토큰값이 생략된 "_<NUM>_" 토큰에 맵핑될 수 있다.

[0026] 도 3과 같이 토큰화부(110)에 의해 입력 프로그램의 텍스트와 토큰이 맵핑된 테이블을 토큰 테이블(Token Table)이라 하며, 어휘 사전이라고도 할 수 있다.

[0027] 위치 인코딩부(120)는 토큰화부(110)가 획득한 토큰 테이블에서 각 토큰의 배치 순서에 따른 위치 정보를 생성하여 맵핑할 수 있다. 여기서 위치 정보는 추후 입력 프로그램에서 수정되어야할 위치를 정확하게 지적할 수 있도록 하기 위함이다. 실제 프로그램의 소스 코드에서는 지정한 양식에 따라 동일한 토큰 및 토큰값이 반복적으로 사용되는 경우가 빈번하게 발생한다. 따라서 입력 프로그램에서 오류가 다수의 동일한 토큰에 맵핑된 동일한 토큰값의 텍스트에서 발생된 경우, 입력 프로그램의 어느 텍스트를 수정해야 하는지 판단하기 어렵게 되는 문제가 있다. 이에 본 실시예에서는 위치 인코딩부(120)가 토큰 테이블의 각 토큰에 대한 위치 정보를 생성하여 맵핑함으로써, 입력 프로그램에서 수정되어야 할 토큰과 수정되지 않아야 할 토큰을 정확하게 지정할 수 있도록 한다. 특히 위치 인코딩부(120)가 각 토큰의 위치 정보를 맵핑함에 따라 입력 프로그램에 다수의 오류가 포함된 경우에도 각 오류가 발생된 위치를 정확하게 지적함에 따라 다수의 오류를 일괄하여 수정할 수 있도록 한다.

[0028] 위치 인코딩부(120)는 토큰화부(110)에 의해 토큰화된 입력 프로그램의 각 토큰의 배치 순서에 따라 오름차순으로 넘버링함으로써, 각 토큰의 위치 정보를 생성할 수 있다. 그러나 경우에 따라서 위치 인코딩부(120)는 토큰의 배치 순서에 따라 내림 차순으로 넘버링할 수도 있다. 일 예로 신경망(140)에 입력되는 최대 입력 시퀀스 길이가 L인 경우에 위치 인코딩부(120)가 다수의 토큰(T1 ~ TL)의 배치 순서에 대해 L부터 내림 차순으로 위치 정보를 생성하여 맵핑한 경우를 도시하였다. 즉 위치 인코딩부(120)는 토큰의 위치를 식별할 수 있는 다양한 방식으로 각 토큰에 대한 위치 정보를 생성하여 맵핑할 수 있다. 이는 신경망(140)의 구조에 따라 오름 차순 또는 내림 차순 위치 정보가 서로 상이한 성능을 나타내는 경우가 있기 때문이며, 이러한 위치 정보 맵핑 방식은 실험을 통해 결정될 수 있다. 그리고 위치 인코딩부(120)는 각 토큰에 대해 생성된 위치 정보를 맵핑하여 토큰 테이블에 추가할 수 있다.

[0029] 여기서는 위치 인코딩부(120)가 토큰화부(110)로부터 토큰 테이블을 인가받아 위치 정보를 생성 및 맵핑하는 것으로 설명하였으나, 위치 인코딩부(120)는 토큰화부(110)와 별개로 입력 프로그램을 직접 인가받아, 입력 프로그램의 각 구성 요소별 위치 정보를 생성 및 맵핑하여 도 4에 도시된 바와 같이, 별도의 위치 정보 테이블을 생성할 수도 있다.

[0030] 입력 임베딩부(130)는 위치 정보가 추가된 토큰 테이블의 토큰들 또는 토큰 테이블의 토큰들과 위치 정보 테이블의 위치 정보를 인가받아 신경망(140)에 임베딩하기 용이한 형태로 변환한다. 입력 임베딩부(130)는 토큰 테이블의 각 토큰과 위치 정보를 개별적으로 벡터화하여 토큰 벡터와 위치 벡터를 각각 획득하고, 획득된 토큰 벡터와 위치 벡터를 결합하여 입력 벡터를 획득할 수 있다. 입력 임베딩부(130)는 기지정된 방식에 따라 토큰과 위치 정보 각각을 벡터화할 수 있다. 일 예로 토큰을 벡터화하는 경우, 미리 획득된 학습용 데이터 셋에 포함된 토큰들을 기반으로, 잘 알려진 Word2Vec와 같은 워드 임베딩 기법으로 벡터화할 수 있다. 그리고 위치 정보는 이미 숫자화된 정보이므로 용이하게 벡터화가 가능하다.

[0031] 한편 입력 임베딩부(130)는 토큰 벡터와 위치 벡터의 결합 시, 다수의 토큰과 각 토큰에 대응하는 위치 정보에서 획득된 서로 대응하는 토큰 벡터와 위치 벡터를 서로 가산하거나 또는 서로 내적하여 다수의 입력 벡터를 획득할 수 있다. 여기서는 일 예로 토큰 벡터와 위치 벡터를 서로 가산하여 입력 벡터를 획득하는 것으로 가정한다.

[0032] 그리고 신경망(140)은 입력 벡터를 인가받아 미리 학습된 방식에 따라 인코딩하여 입력 프로그램의 의미적 특징이 함축된 컨텍스트 벡터(Context Vector)를 획득하고, 획득된 컨텍스트 벡터를 다시 학습된 방식에 따라 디코딩하여 입력 프로그램의 오류를 수정하기 수정 명령을 출력할 수 있다.

[0033] 본 실시예에서 신경망(140)은 신경망 모델의 일종인 시퀀스 투 시퀀스(sequence-to-sequence: 이하 seq2seq) 모델로 구현될 수 있다. seq2seq 모델은 번역 어플리케이션을 위해 주로 이용되는 신경망으로서, 현재 특정 언어로 구현된 시퀀스를 타겟 언어의 시퀀스로 번역하기 위한 표준 신경망으로 이용되고 있다. 뿐만 아니라, seq2seq 모델은 대화 시스템, 텍스트 요약, 프로그램 합성, 비디오 캡션, 문법 오류 감지 및 수정 등의 다양한 어플리케이션에도 이용되고 있다.

[0034] 도 1과 같이 seq2seq 모델에 따라 구현되는 신경망(140)은 인코더(141)와 디코더(143)로 구성될 수 있다. 그리고 인코더(141)와 디코더(143)는 각각 순차적으로 연결된 다수의 신경망 셀을 포함하여 구성될 수 있으며, 다수의 신경망 셀 각각은 일 예로 도 2에 도시된 바와 같이, LSTM(Long Short-Term Memory) 셀(또는 다수의 GRU(Gated Recurrent Unit) 셀)로 구현될 수 있다. 즉 인코더와 디코더는 각각 LSTM(또는 GRU)로 구현될 수 있으며, 경우에 따라서는 다른 RNN 계열의 신경망으로 구현될 수도 있다.

[0035] 인코더(141)의 다수의 신경망 셀인 인코딩 셀들 각각은 토큰에 대응하는 위치 정보가 가중된 입력 벡터 및 이전 배치된 인코딩 셀에서 출력되는 히든 벡터를 인가받고, 학습된 방식에 따라 인가된 입력 벡터와 히든 벡터로부터 특징 벡터와 히든 벡터를 추출한다. 그리고 다수의 인코딩 셀 중 최종 배치된 인코딩 셀에서 추출된 히든 벡터는 컨텍스트 벡터로서 디코더(143)로 전달된다.

[0036] 디코더(143)의 다수의 신경망 셀인 디코딩 셀들 중 초기에 배치된 디코딩 셀은 인코더(141)에서 획득된 히든 벡터인 컨텍스트 벡터를 인가받고, 나머지 신경망 셀은 이전 배치된 디코딩 셀에서 추출된 히든 벡터를 인가받아, 미리 학습된 방식에 따라 디코딩하여 다수의 입력 벡터 각각에 대응하는 수정 명령을 출력한다. 즉 각 토큰에 대응하는 수정 명령을 출력한다. 도 2에서는 설명의 편의를 위하여, 디코더(143)의 다수의 디코딩 셀이 LSTM 셀로 구성되는 것으로 도시하였으나, 다수의 디코딩 셀 각각은 LSTM 셀에서 추출되는 디코딩 값을 수정 명령으로 변환하기 위한 추가 구성을 포함할 수 있다.

[0037] 디코더(143)는 토큰 테이블의 다수의 토큰 각각에 대한 수정 명령을 획득할 수 있다. 예로서 디코더(143)는 수정하지 않을 토큰에 대해서는 "Nothing" 명령을 출력하고, 수정할 토큰에 대해서는 "Replace", "Insert" 및 "Delete"의 세가지 수정 명령을 출력할 수 있다.

[0038] 다만 디코더(143)가 인코더(141)의 각 인코딩 셀에서 추출된 히든 벡터 중 최종 인코딩 셀에서 추출된 히든 벡터인 컨텍스트 벡터만을 인가받아 디코딩하여 각 토큰에 대응하는 수정 명령을 생성하는 경우, 획득되는 수정 명령의 정확도가 낮아질 수 있다. 이에 디코더(143)의 다수의 디코딩 셀 각각은 인코더(141)의 다수의 인코딩 셀 각각에서 획득된 다수의 히든 벡터를 추가로 인가받고, 추가로 인가된 히든 벡터를 참조하여, 디코딩함으로써 디코딩 성능을 향상시킬 수 있다. 이때 다수의 디코딩 셀 각각이 다수의 인코딩 셀 각각에서 획득된 다수의 히든 벡터를 참조함에 있어 서로 다르게 주의를 기울여 참조할 수 있도록 신경망(140)은 어텐션부(미도시)를 추가로 적용할 수 있다.

[0039] 도 2에 도시된 바와 같이, 어텐션부는 인코더(141)와 디코더(143) 사이에 적용될 수 있으며, 디코더(143)의 다수의 디코딩 셀 각각이 디코딩 동작을 수행할 때, 인코더(141)의 다수의 인코딩 셀 각각에서 추출된 다수의 히든 벡터 각각에 대해 어텐션 스코어를 기지정된 방식으로 획득하여 가중함으로써, 각 히든 벡터에 주의를 기울이는 수준을 상이하게 지정할 수 있다. 즉 어텐션부는 각 디코딩 셀이 다수의 토큰 각각에 대해 관심을 기울이는 수준을 조절한다. 어텐션부는 알려진 어텐션 메커니즘에 따라 다양한 방식으로 구현될 수 있으며, 일 예로, 각 디코딩 셀에서 추출된 히든 벡터와 다수의 인코딩 셀에서 추출된 히든 벡터들을 내적하고 정규화하여 획득될 수 있다. 여기서 어텐션 스코어는 미리 지정된 방식에 따라 수학적으로 계산될 수도 있으나, 신경망(140)의 학습 시에 학습에 의해 획득될 수도 있다.

[0040] 코드 수정부(150)는 신경망(140)에서 출력되는 수정 명령에 따라 입력 프로그램을 수정하여 오류가 수정된 수정 프로그램을 출력한다. 즉 소스 코드 수정부(150)는 신경망(140)에서 "Nothing" 명령이 출력된 토큰은 변경하지

않고 유지하며, "Replace" 명령이 출력된 토큰은 명령과 함께 출력된 구성 요소로 토큰을 교체하며, "Insert" 명령이 출력된 토큰은 해당 위치에 명령과 함께 출력된 구성 요소를 추가하며, "Delete" 명령이 출력된 토큰은 삭제하여 입력 프로그램을 수정한다. 코드 수정부(150)는 "Nothing", "Replace", "Insert" 및 "Delete"의 명령에 따라 다수의 토큰 각각에 대해 순차적 수정을 수행하므로, 신경망(140)에서 출력되는 수정 명령에는 이미 위치 정보가 반영된 것으로 볼 수 있다. 따라서 신경망(140)은 별도로 위치 정보를 출력하지 않고서, 다수의 오류가 포함된 입력 프로그램을 한 번에 수정할 수 있다.

[0041] 여기서 코드 수정부(150)는 토큰 단위로 입력 프로그램을 수정하므로, 수정이 완료된 이후에는 역토큰화를 통해 오류가 수정된 수정 프로그램을 복원할 수 있다.

[0042] 상기한 프로그램 소스 코드 자동 수정 장치는 신경망(140)을 포함하여 구성되므로, 미리 학습이 수행되어야 한다. 이에 프로그램 소스 코드 자동 수정 장치는 신경망(140)을 학습시키기 위한 편집 동기화부(200)를 더 포함할 수 있다.

[0043] 도 1에 도시된 바와 같이, 편집 동기화부(200)는 편집 정보 획득부(210), 위치 동기화부(220) 및 편집 임베딩부(230)를 포함할 수 있다. 편집 정보 획득부(210)는 학습 시에 입력되는 입력 프로그램인 학습 프로그램과 학습 프로그램의 오류가 수정된 진리 프로그램을 인가받아 비교하여, 도 5에 도시된 바와 같이, 학습 프로그램이 진리 프로그램으로 수정되도록 하기 위한 편집 정보를 획득한다.

[0044] 학습 프로그램과 진리 프로그램은 학습 데이터 셋으로 학습 프로그램은 상기한 바와 같이 입력 프로그램에 대응하는 프로그램으로 오류가 포함될 수 있는 반면, 진리 프로그램은 사전에 오류가 수정되어 컴파일 가능한 프로그램을 의미한다.

[0045] 편집 정보 획득부(210)는 편집 거리(Edit distance) 알고리즘 기법을 이용하여 두 프로그램 사이의 유사도를 판단하고, 판단된 유사도를 기반으로 최소 편집으로 학습 프로그램이 진리 프로그램으로 전환되도록 하기 위한 편집 정보를 획득한다. 여기서 편집 정보 또한 디코더에서 출력되는 수정 명령과 마찬가지로 "Nothing", "Replace", "Insert" 및 "Delete" 명령의 조합으로 획득된다.

[0046] 도 5를 참조하면, 진리 프로그램과 학습 프로그램에서 서로 대응하는 토큰들이 동일한 경우, 편집 정보 획득부(210)는 "Nothing" 명령을 획득한다. 그리고 진리 프로그램의 3번째 토큰 "int"가 학습 프로그램에서는 "char"로 표기된 텍스트 표기 오류가 있다. 이와 같은 텍스트 표기 오류가 존재하는 경우, 편집 정보 획득부(210)는 학습 프로그램의 "char"가 "int"로 대체되도록 "Replace_<type>_int" 명령을 획득할 수 있다. 또한 진리 프로그램의 7번째 토큰에는 "{"가 표기되어 있으나, 학습 프로그램에는 누락되었다. 이렇게 누락이 검출되는 경우, 편집 정보 획득부(210)는 "Insert_<op>_{"를 획득하여, 누락된 "{"가 학습 프로그램에 추가되어야 함을 지정할 수 있으며, 학습 프로그램의 9번째 토큰에 "("가 더 포함된 경우, 편집 정보 획득부(210)는 "Delete_<op>_("를 획득하여, 학습 프로그램에 잘못 삽입된 "("가 삭제되도록 지정할 수 있다.

[0047] 여기서 편집 정보 획득부(210)는 토큰화된 학습 프로그램과 진리 프로그램을 인가받아 편집 거리 알고리즘을 적용하여 편집 정보를 획득할 수 있다.

[0048] 그리고 편집 정보 획득부(210)는 획득된 편집 정보를 벡터화하여 편집 벡터를 획득할 수 있다.

[0049] 한편, 위치 동기화부(220)는 학습 프로그램의 토큰과 진리 프로그램의 토큰 사이의 위치 정보 변화를 확인하여 동기화를 수행한다.

[0050] 프로그램 소스 코드 자동 수정 장치의 코드 수정부(150)가 입력 프로그램의 오류를 수정하여 수정 프로그램을 획득하는 경우, 수정 프로그램의 각 토큰의 위치 정보에 변화가 발생할 수 있다. 입력 프로그램의 토큰을 수정하지 않는 "Nothing" 명령의 경우, 수정 사항이 없으므로 토큰의 위치 정보에는 변화가 발생하지 않는다. 그리고 특정 토큰을 다른 구성 요소로 대체하는 "Replace" 명령의 경우에도 대체되는 구성 요소가 해당 토큰의 위치에 대치될 뿐이므로 대체된 토큰에 의한 위치 정보 변화는 발생하지 않는다. 또한 "Delete" 명령의 경우, 해당 토큰이 삭제되지만, 삭제된 위치 정보를 제외한 나머지 토큰에 대한 위치 정보를 그대로 유지할 수 있으므로, 나머지 토큰들의 위치 정보를 변경할 필요는 없다.

[0051] 그러나 "Insert" 명령의 경우, 서로 인접한 위치 정보를 갖는 토큰 사이에 새로운 토큰, 즉 구성 요소가 삽입됨에 따라 위치 정보에 변화가 발생할 수 있다. 즉 "Insert" 명령 이후에 위치하는 토큰들의 위치 정보가 증가될 수 있으며, 이는 코드 수정부(150)가 수정하지 않아야 할 토큰을 수정하게 되는 문제를 야기할 수 있다.

[0052] 따라서 신경망(140)의 디코더는 수정 명령을 출력할 때, 원래 위치에 해당하는 토큰 위치로 업데이트된 수정 명

령이 출력되도록 학습될 필요가 있으며, 이에 위치 동기화부(220)는 도 6에 도시된 바와 같이, 학습 프로그램의 토큰과 진리 프로그램의 토큰 사이의 위치 정보 변화를 확인하여, 위치 정보가 동기화될 수 있는 동기 위치 정보를 획득할 수 있다. 그리고 동기 위치 정보를 벡터화하여 동기 위치 벡터를 획득할 수 있다.

[0053] 편집 임베딩부(230)는 편집 벡터와 동기 위치 벡터를 인가받아 결합하여 업데이트 벡터를 획득하고, 획득된 업데이트 벡터를 디코더(143)의 각 디코딩 셀에 입력하여 각 디코딩 셀이 업데이트되도록 한다. 즉 업데이트 벡터를 손실로 역전파하여 신경망(140)을 학습시킬 수 있다.

[0054] 프로그램 소스 코드 자동 수정 장치의 학습 시에 편집 동기화부(200)는 코드 수정부(150)에서 수정되어 출력되는 수정 프로그램을 다시 학습 프로그램으로서 토큰화부(110)에 입력하여 반복적으로 업데이트 벡터를 신경망(140)의 디코더(143)로 임베딩함으로써, 반복 학습을 수행할 수 있다.

[0055] 따라서 디코더(143)의 다수의 디코딩 셀 각각은 학습 시에 도 2에 도시된 바와 같이, 동기 위치 정보가 포함된 업데이트 벡터를 인가받아 학습되며, 이로 인해 이후 실제 수정 명령 획득 시에 동기 위치 벡터가 인가되지 않아도 수정 위치 오류가 발생되지 않도록 수정 명령을 생성할 수 있다.

[0056] 즉 본 실시예에서 신경망(140)은 입력 프로그램과 진리 프로그램 사이의 차이를 역전파하여 지도 학습이 수행되는 것으로 볼 수 있다.

[0057] 도 7은 본 발명의 일 실시예에 따른 프로그램 소스 코드 자동 수정 방법을 나타낸다.

[0058] 도 1 내지 도 6을 참조하면, 본 실시예에 따른 프로그램 소스 코드 자동 수정 방법은 크게 코드 수정 단계(S10) 및 학습 단계(S20)를 포함할 수 있다.

[0059] 코드 수정 단계(S10)에서는 우선 입력 프로그램이 인가되면, 입력 프로그램을 최소 구성 요소 단위로 구분하여 기지정된 방식으로 토큰화하여 토큰 테이블을 획득한다(S11). 그리고 토큰 테이블의 다수의 토큰 각각의 배치 순서에 따른 위치 정보를 생성하여 맵핑한다(S12). 그리고 다수의 토큰 각각과 대응하는 위치 정보 각각을 벡터화하고 결합하여 다수의 입력 벡터를 획득한다(S13).

[0060] 입력 벡터가 획득되면 미리 학습된 신경망(140)의 인코더(141)에 다수의 입력 벡터를 인가받아 인코딩한다(S14). 여기서 신경망(140)의 인코더(141)는 순차 연결된 다수의 인코딩 셀로 구성되고, 다수의 인코딩 셀 각각은 대응하는 입력 벡터와 이전 배치된 인코딩 셀에서 출력되는 히든 벡터를 인가받아 미리 학습된 방식에 따라 인코딩하여 특징 벡터와 히든 벡터를 추출한다.

[0061] 그리고 신경망의 디코더(143)는 인코더(141)의 다수의 인코딩 셀 중 마지막에 배치된 인코딩 셀에서 출력되는 히든 벡터인 컨텍스트 벡터를 인가받아, 미리 학습된 방식으로 디코딩하여 입력 프로그램의 오류를 수정하기 위한 수정 명령을 획득한다(S15). 이때 디코더(143)에서 순차적으로 연결된 다수의 디코딩 셀은 이전 디코딩 셀에서 추출된 히든 벡터를 디코딩하여, 각 토큰에 대응하는 수정 명령을 출력한다.

[0062] 그리고 디코더(143)는 어텐션 메커니즘에 따라 기지정된 방식으로 획득되는 어텐션 스코어를 다수의 인코딩 셀 각각에서 추출된 히든 벡터에 가중하여 서로 상이한 관심도로 각 히든 벡터를 참조하여, 이전 디코딩 셀에서 추출된 히든 벡터를 디코딩하여 수정 명령을 획득할 수 있다.

[0063] 각 토큰에 대한 수정 명령이 획득되면, 토큰 테이블의 다수의 토큰 각각을 대응하는 수정 명령에 따라 수정하고, 역토큰화하여 입력 프로그램의 오류가 수정된 수정 프로그램을 획득한다(S16).

[0064] 그리고 학습 단계(S20)에서는 학습 프로그램과 학습 프로그램의 오류가 미리 수정된 진리 프로그램을 인가받고, 학습 프로그램과 진리 프로그램의 각 토큰들의 변화를 편집 거리 알고리즘에 따라 비교하여 학습 프로그램이 진리 프로그램으로 전환되도록 하기 위한 다수의 수정 명령을 포함하는 편집 정보를 획득한다(S21). 여기서 학습 프로그램은 학습 시에 이용되는 입력 프로그램에 대응하는 프로그램으로서 진리 프로그램과 함께 데이터 셋으로 미리 획득된 프로그램을 나타낸다.

[0065] 그리고 편집 정보를 벡터로 변환하여 편집 벡터를 획득한다(S22). 한편, 학습 프로그램과 진리 프로그램의 대응하는 토큰들 사이의 위치를 동기화하기 위한 동기 위치 정보를 획득하고 벡터화하여 동기 위치 벡터를 획득한다(S23).

[0066] 편집 벡터와 동기 위치 벡터가 획득되면, 편집 벡터와 동기 위치 벡터를 기지정된 방식에 따라 결합하여 업데이트 벡터를 획득하고, 획득된 업데이트 벡터를 디코더(143)의 각 디코딩 셀에 입력하여 각 디코딩 셀이 업데이트되도록 한다(S24). 즉 업데이트 벡터를 손실로 역전파하여 신경망(140)을 학습시킬 수 있다.

[0067] 학습 단계(S20)는 프로그램 소스 코드 자동 수정 장치에 입력된 학습 프로그램이 진리 프로그램과 동일해질 때까지 반복 수행될 수 있다.

[0068] 본 발명에 따른 방법은 컴퓨터에서 실행시키기 위한 매체에 저장된 컴퓨터 프로그램으로 구현될 수 있다. 여기서 컴퓨터 판독가능 매체는 컴퓨터에 의해 액세스 될 수 있는 임의의 가용 매체일 수 있고, 또한 컴퓨터 저장 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함하며, ROM(판독 전용 메모리), RAM(랜덤 액세스 메모리), CD(컴팩트 디스크)-ROM, DVD(디지털 비디오 디스크)-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등을 포함할 수 있다.

[0069] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다.

[0070] 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

[0071] 110: 토큰화부 120: 위치 인코딩부

130: 입력 임베딩부 140: 신경망

141: 인코더 143: 디코더

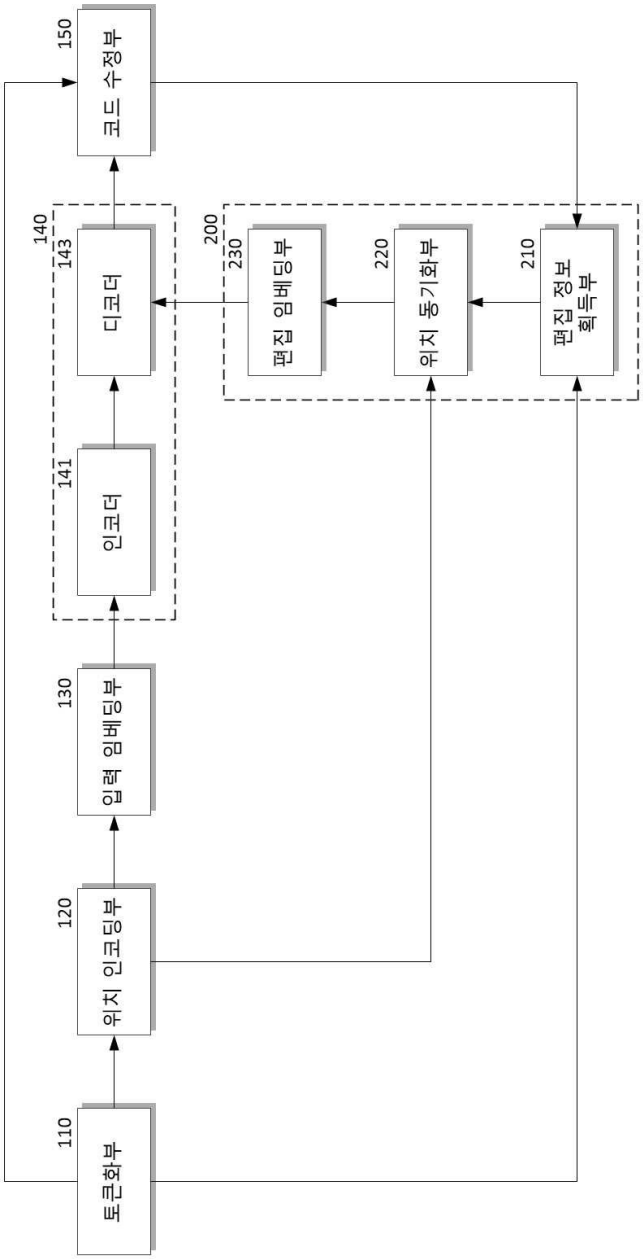
150: 코드 수정부 200: 편집 동기화부

210: 편집 정보 획득부 220: 위치 동기화부

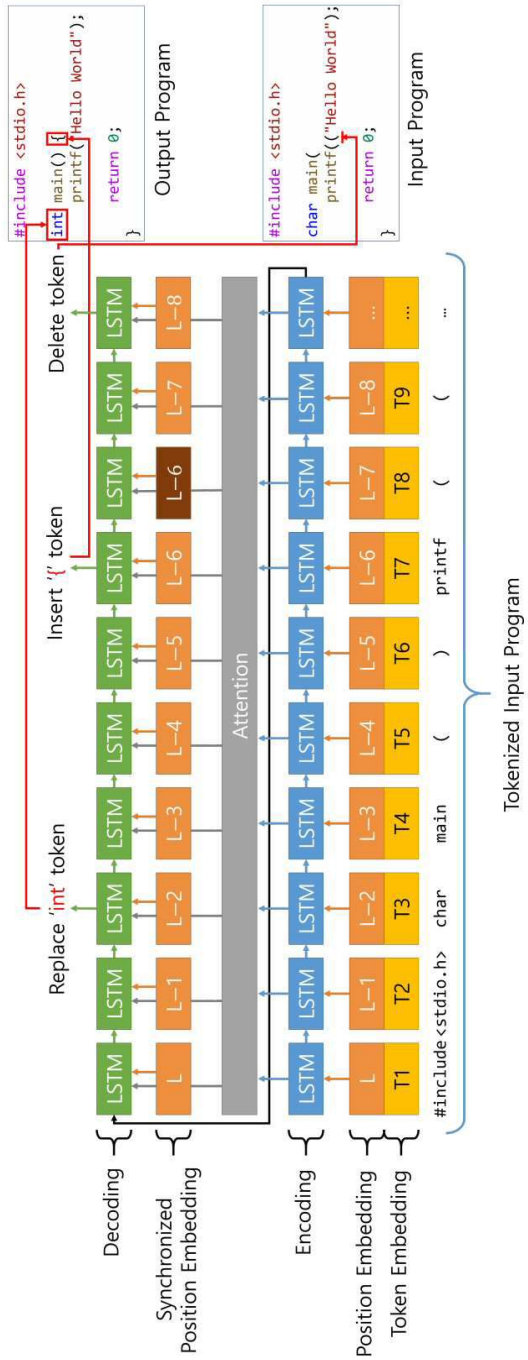
230: 편집 임베딩부

도면

도면1


150
코드 수정부
140
143
디코더
141
인코더
130
입력 임베딩부
120
위치 인코딩부
110
토큰화부
200
230
편집 임베딩부
220
위치 동기화부
210
편집 정보 획득부

도면2


#include <stdio.h>
int main() {
printf("Hello World");
return 0;
}
Output Program
#include <stdio.h>
char main(
printf("Hello World");
return 0;
}
Input Program
Delete token
Insert '(' token
Replace 'int' token
Attention
LSTM
L
L-1
L-2
L-3
L-4
L-5
L-6
L-7
L-8
T1
T2
T3
T4
T5
T6
T7
T8
T9
#include <stdio.h> char main ( ) printf (...
Tokenized Input Program
Decoding
Synchronized Position Embedding
Encoding
Position Embedding
Token Embedding

도면3

	어휘 사전
Text	Token
#include	_<directive>_#include
<stdio.h>	_<include>_<stdio.h>
char	_<type>_char
main	_<APIcall>_main
(	_<op>_(
)	_<op>_)
printf	_<APIcall>_printf
“Hello World”	_<STR>_
;	_<op>_;
return	_<keyword>_return
0	_<NUM>_
}	_<op>_}
…	…

도면4

Text	#include	<stdio.h>	char	main	(	)	printf	(	(	…
Position	1	2	3	4	5	6	7	8	9	…

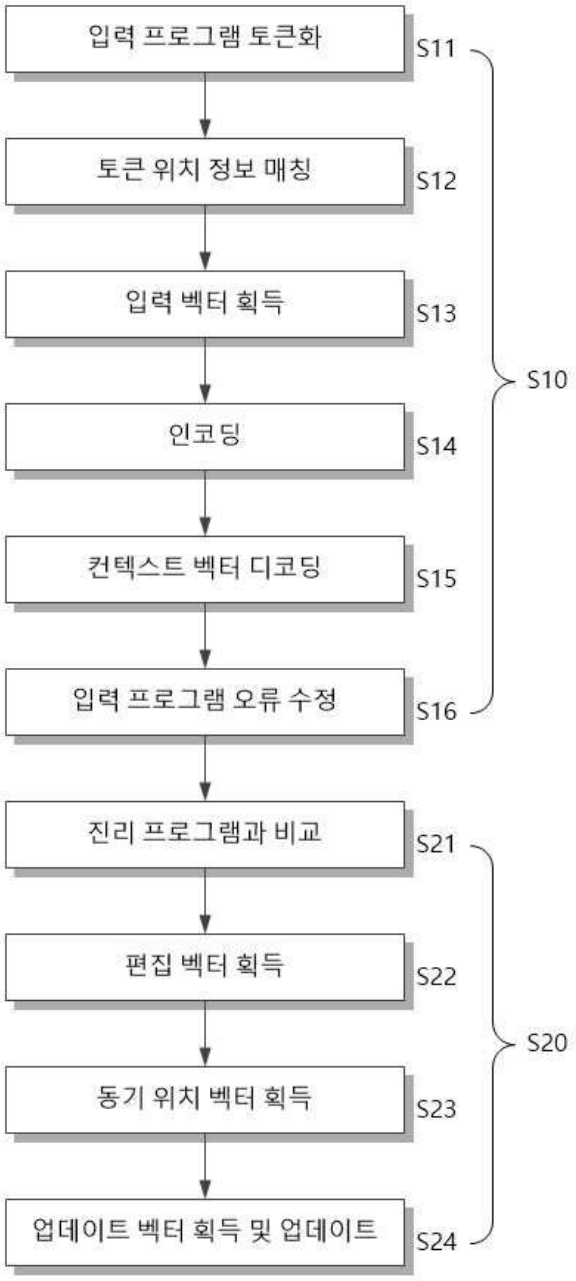
도면5

Ground-truth	Input	Target
#include	#include	Nothing
<stdio.h>	<stdio.h>	Nothing
int	char	Replace_<type>_int
main	main	Nothing
(	(	Nothing
)	)	Nothing
{	printf	Insert_<op>_{
printf	(	Nothing
(	(	Nothing
“Hello World”	“Hello World”	Delete
)	)	Nothing
…	…	…

도면6

Ground-truth	Input	Position	Target	Position	Synchronized Position
#include	#include	1	Nothing	1	1
<stdio.h>	<stdio.h>	2	Nothing	2	2
int	char	3	Replace_<type>_int	3	3
main	main	4	Nothing	4	4
(	(	5	Nothing	5	5
)	)	6	Nothing	6	6
{	printf	7	Insert_<op>_{	7	7
printf	(	8	Nothing	8	7
(	(	9	Nothing	9	8
“Hello World”	“Hello World”	10	Delete	10	9
)	)	11	Nothing	11	10
…	…	…	…	…	…

도면7


입력 프로그램 토큰화
S11
토큰 위치 정보 매칭
S12
입력 벡터 획득
S13
인코딩
S14
컨텍스트 벡터 디코딩
S15
입력 프로그램 오류 수정
S16
S10
진리 프로그램과 비교
S21
편집 벡터 획득
S22
동기 위치 벡터 획득
S23
업데이트 벡터 획득 및 업데이트
S24
S20