(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년12월06일
(11) 등록번호 10-2610431
(24) 등록일자 2023년12월01일

(51) 국제특허분류(Int. Cl.)
G06F 8/73 (2018.01) *G06F 16/34* (2019.01)
G06F 16/36 (2019.01) *G06F 8/75* (2018.01)
(52) CPC특허분류
G06F 8/73 (2013.01)
G06F 16/345 (2019.01)
(21) 출원번호 10-2021-0150771
(22) 출원일자 2021년11월04일
심사청구일자 2021년11월04일
(65) 공개번호 10-2023-0065017
(43) 공개일자 2023년05월11일
(56) 선행기술조사문헌
KR1020210058059 A
KR1020200097218 A
JP2021121952 A
JP2020087127 A

(73) 특허권자
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
한요섭
서울특별시 서대문구 연세로 50 연세대학교 컴퓨터과학과
손지경
서울특별시 관악구 승방6길 6, 401호
(뒷면에 계속)
(74) 대리인
정부연

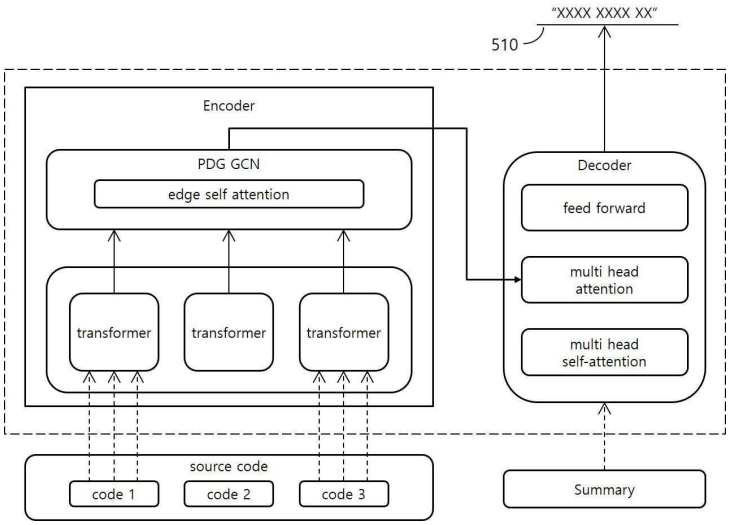
전체 청구항 수 : 총 14 항 심사관 : 지정훈

(54) 발명의 명칭 **인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치 및 방법**

(57) 요 약

본 발명은 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치 및 방법에 관한 것으로, 상기 장치는 특정 프로그램에 관한 소스코드 및 요약문을 입력받는 프로그램 입력부; 상기 소스코드를 분석하여 간선과 노드로 구성된 프로그램 의존 그래프를 생성하는 그래프 생성부; 상기 소스코드 및 상기 요약문을 기초로 적어도 하나의 어휘사전을 생성하는 어휘사전 생성부; 및 동작 과정에서 상기 적어도 하나의 어휘사전이 적용되는 인코더(Encoder)와 디코더(Decoder)를 포함하고, 상기 소스코드에 대응되는 소스코드 임베딩을 기초로 상기 노드의 어텐션을 산출하여 소스코드 요약문을 생성하는 소스코드 요약문 생성부;를 포함한다.

대 표 도 - 도5



(52) CPC특허분류
G06F 16/36 (2019.01)
G06F 8/75 (2013.01)

(72) 발명자
한중혁
서울특별시 마포구 토정로18길 11, 래미안웰스트림 아파트 102동 1604호

서현태
경기도 안양시 만안구 박달로 403, 한일유앤아이아파트 101동 1502호

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126082
과제번호	2020-0-01361-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	인공지능대학원지원(연세대학교)
기 여 율	1/2
과제수행기관명	연세대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711135204
과제번호	2020R1A4A3079947
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	집단연구지원(R&D)
연구과제명	휴먼-AI 협업 프로그래밍 플랫폼 기술 연구실
기 여 율	1/2
과제수행기관명	연세대학교
연구기간	2021.06.01 ~ 2022.02.28

명 세 서

청구범위

청구항 1

특정 프로그램에 관한 소스코드 및 요약문을 입력받는 프로그램 입력부;

상기 소스코드를 분석하여 간선과 노드로 구성된 프로그램 의존 그래프를 생성하는 그래프 생성부;

상기 소스코드 및 상기 요약문을 기초로 적어도 하나의 어휘사전을 생성하는 어휘사전 생성부; 및

동작 과정에서 상기 적어도 하나의 어휘사전이 적용되는 인코더(Encoder)와 디코더(Decoder)를 포함하고, 상기 소스코드에 대응되는 소스코드 임베딩을 기초로 상기 노드의 어텐션을 산출하여 소스코드 요약문을 생성하는 소스코드 요약문 생성부;를 포함하고,

상기 인코더(Encoder)는 트랜스포머(transformer) 과정의 출력이 그래프 합성곱 과정의 입력에 연결되도록 구현되는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 2

제1항에 있어서, 상기 그래프 생성부는

상기 소스코드의 코드 블록을 상기 노드로 구성하고 데이터 플로우 또는 제어 플로우로 정의되는 상기 노드 간의 연결을 상기 간선으로 구성하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 3

제1항에 있어서, 상기 어휘사전 생성부는

최대 빈도수를 기준으로 상기 어휘사전에 있는 어휘 개수를 상위 N 개로 제한하고, 상기 N은 상기 소스코드의 크기에 비례하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 4

제1항에 있어서, 상기 소스코드 요약문 생성부는

상기 소스코드의 입력 토큰들에 대한 워드 임베딩을 통해 각 입력 토큰 별로 각각이 해당 토큰의 의미 정보와 위치 정보를 포함하는 토큰 임베딩 및 위치 임베딩을 생성하고, 상기 토큰 임베딩 및 상기 위치 임베딩 간의 합연산에 의해 생성된 임베딩 벡터를 상기 인코더의 입력으로 사용하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 5

제4항에 있어서, 상기 소스코드 요약문 생성부는

상기 소스코드 임베딩을 기초로 상기 입력 토큰들의 어텐션을 산출하여 상기 소스코드 내 자기학습을 강화하는 셀프 어텐션(self-attention) 과정과, 상기 셀프 어텐션 과정의 결과로서 획득된 노드 어텐션 벡터를 완전 연결 계층에 입력하여 노드 잠재표현 벡터를 생성하는 피드 포워드(feed forward) 과정을 포함하는 상기 트랜스포머 과정을 수행하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 6

제5항에 있어서, 상기 소스코드 요약문 생성부는

상기 그래프 합성곱 과정에서 상기 프로그램 의존 그래프의 간선 정보를 기초로 상기 노드 잠재표현 벡터 간의 합성 곱을 반복적으로 수행하여 상기 노드의 어텐션을 산출하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 7

제6항에 있어서, 상기 소스코드 요약문 생성부는

상기 노드의 어텐션과 상기 요약문을 상기 디코더에 입력하여 요약 어휘 임베딩을 생성하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 8

제7항에 있어서, 상기 소스코드 요약문 생성부는

상기 노드의 어텐션을 상기 디코더의 멀티 헤드 어텐션 과정의 입력으로 사용하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 9

제7항에 있어서, 상기 소스코드 요약문 생성부는

상기 요약 어휘 임베딩을 활성화 함수에 적용한 결과를 기초로 상기 소스코드 요약문을 생성하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치.

청구항 10

특정 프로그램의 소스코드를 분석하여 프로그램 의존 그래프를 생성하는 단계;

상기 소스코드 및 상기 특정 프로그램에 관한 요약문을 기초로 적어도 하나의 어휘사전을 생성하는 단계;

상기 소스코드에 대응되는 소스코드 임베딩을 생성하는 단계;

상기 소스코드 임베딩을 기초로 상기 프로그램 의존 그래프의 노드들에 관한 제1 노드 임베딩을 생성하는 단계;

상기 제1 노드 임베딩 및 상기 프로그램 의존 그래프의 간선 정보를 이용하여 그래프 어텐션(graph attention)을 수행하는 단계;

상기 그래프 어텐션에 따른 노드들의 제2 노드 임베딩 및 상기 요약문을 기초로 요약 어휘들에 관한 요약 어휘 임베딩을 생성하는 단계; 및

상기 요약 어휘 임베딩을 기초로 상기 소스코드에 관한 소스코드 요약문을 생성하는 단계;를 포함하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법.

청구항 11

제10항에 있어서, 상기 소스코드 임베딩을 생성하는 단계는

상기 소스코드의 입력 토큰들에 대한 워드 임베딩을 통해 각 입력 토큰 별로 각각이 해당 토큰의 의미 정보와 위치 정보를 포함하는 토큰 임베딩 및 위치 임베딩을 생성하는 단계; 및

상기 토큰 임베딩 및 상기 위치 임베딩 간의 합 연산을 통해 임베딩 벡터를 생성하는 단계를 포함하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법.

청구항 12

제11항에 있어서, 상기 제1 노드 임베딩을 생성하는 단계는

셀프 어텐션(self-attention) 과정을 통해 상기 소스코드 임베딩을 기초로 상기 입력 토큰들의 어텐션을 산출하여 상기 소스코드 내 자기학습을 강화하는 단계; 및

피드 포워드(feed forward) 과정을 통해 상기 셀프 어텐션 과정의 결과로서 획득된 노드 어텐션 벡터를 완전 연결 계층에 입력하여 노드 잠재표현 벡터를 생성하는 단계를 포함하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법.

청구항 13

제10항에 있어서, 상기 그래프 어텐션을 수행하는 단계는

상기 프로그램 의존 그래프의 간선 정보를 기초로 상기 제1 노드 임베딩 간의 합성 곱을 반복적으로 수행하는 단계를 포함하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법.

청구항 14

제10항에 있어서, 상기 소스코드 요약문을 생성하는 단계는

상기 요약 어휘 임베딩을 활성화 함수에 적용한 결과를 기초로 상기 소스코드 요약문을 생성하는 단계를 포함하는 것을 특징으로 하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 소스코드 요약 기술에 관한 것으로, 보다 상세하게는 기존의 트랜스포머 모델을 기반으로 그래프 처리 모델을 적용하여 프로그램 소스코드에 대한 자연어 요약문을 생성하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치 및 방법에 관한 것이다.

배 경 기 술

[0003] 소스코드(source code)의 요약은 코드에 관한 간단한 자연어(natural language) 설명을 생성하는 작업에 해당할 수 있다. 소스코드에 대한 간단한 설명은 프로그래머가 코드 자체를 읽을 필요없이 해당 코드의 동작과 전체 프로그램에서 해당 코드의 목적을 쉽게 이해하도록 할 수 있다.

[0004] 또한, 프로그래머는 소스코드에 대한 요약문을 통해 해당 코드의 동작에 대한 명확한 그림을 그릴 수 있고, 해당 코드의 구체적인 동작을 이해하는데 소요되는 시간을 절약할 수 있다.

[0005] 이에 따라, 자동 코드 요약 기술은 빠르게 연구되어 왔으며, 최근에는 AI, 자연어 처리(NLP) 및 마이닝(mining) 분야에서 신경망을 적용하여 소스코드의 요약문을 생성하고자 하는 시도가 존재한다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 한국공개특허 제10-2013-0116908호 (2013.10.24)

발명의 내용

해결하려는 과제

[0008] 소스코드와 추상구문트리를 이용하는 기존의 소스코드 요약생성 모델의 경우 구조정보 학습을 위한 별도의 인코더를 둠으로써 모델의 크기가 커져 훈련 효율성이 상대적으로 낮은 문제점이 있다. 또한, 소스코드의 제어흐름을 이용한 기존의 소스코드 요약생성 모델의 경우 제어흐름을 사전 학습하는 과정이 추가되어 이중훈련이 필요하고 요약문을 생성하는데 데이터흐름 정보를 포함하지 않는 문제점이 있다.

[0009] 본 발명에 따른 기존의 트랜스포머 모델을 기반으로 그래프 처리 모델을 적용하여 프로그램 소스코드에 대한 자연어 요약문을 생성하는 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치 및 방법을 제공하고자 한다.

과제의 해결 수단

[0011] 실시예들 중에서, 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치는 특정 프로그램에 관한 소스코드 및 요약문을 입력받는 프로그램 입력부; 상기 소스코드를 분석하여 간선과 노드로 구성된 프로그램 의존 그래프를 생성하는 그래프 생성부; 상기 소스코드 및 상기 요약문을 기초로 적어도 하나의 어휘사전을 생성하는 어휘사전 생성부; 및 동작 과정에서 상기 적어도 하나의 어휘사전이 적용되는 인코더(Encoder)와 디코더(Decoder)를 포함하고, 상기 소스코드에 대응되는 소스코드 임베딩을 기초로 상기 노드의 어텐션을 산출하여 소스코드 요약문을 생성하는 소스코드 요약문 생성부;를 포함한다.

[0012] 이때, 상기 인코더(Encoder)는 트랜스포머(transformer) 과정의 출력이 그래프 합성곱 과정의 입력에 연결되도록 구현될 수 있다.

[0013] 상기 그래프 생성부는 상기 소스코드의 코드 블록을 상기 노드로 구성하고 데이터 플로우 또는 제어 플로우로 정의되는 상기 노드 간의 연결을 상기 간선으로 구성할 수 있다.

[0014] 상기 어휘사전 생성부는 최대 빈도수를 기준으로 상기 어휘사전에 있는 어휘 개수를 상위 N 개로 제한할 수 있고, 상기 N은 상기 소스코드의 크기에 비례할 수 있다.

[0015] 상기 소스코드 요약문 생성부는 상기 소스코드의 입력 토큰들에 대한 워드 임베딩을 통해 각 입력 토큰 별로 각각이 해당 토큰의 의미 정보와 위치 정보를 포함하는 토큰 임베딩 및 위치 임베딩을 생성하고, 상기 토큰 임베딩 및 상기 위치 임베딩 간의 합 연산에 의해 생성된 임베딩 벡터를 상기 인코더의 입력으로 사용할 수 있다.

[0016] 상기 소스코드 요약문 생성부는 상기 소스코드 임베딩을 기초로 상기 입력 토큰들의 어텐션을 산출하여 상기 소스코드 내 자기학습을 강화하는 셀프 어텐션(self-attention) 과정과, 상기 셀프 어텐션 과정의 결과로서 획득된 노드 어텐션 벡터를 완전 연결 계층에 입력하여 노드 잠재표현 벡터를 생성하는 피드 포워드(feed forward) 과정을 포함하는 상기 트랜스포머 과정을 수행할 수 있다.

[0017] 상기 소스코드 요약문 생성부는 상기 그래프 합성곱 과정에서 상기 프로그램 의존 그래프의 간선 정보를 기초로 상기 노드 잠재표현 벡터 간의 합성 곱을 반복적으로 수행하여 상기 노드의 어텐션을 산출할 수 있다.

[0018] 상기 소스코드 요약문 생성부는 상기 노드의 어텐션과 상기 요약문을 상기 디코더에 입력하여 요약 어휘 임베딩을 생성할 수 있다.

[0019] 상기 소스코드 요약문 생성부는 상기 노드의 어텐션을 상기 디코더의 멀티 헤드 어텐션 과정의 입력으로 사용할 수 있다.

[0020] 상기 소스코드 요약문 생성부는 상기 요약 어휘 임베딩을 활성화 함수에 적용한 결과를 기초로 상기 소스코드 요약문을 생성할 수 있다.

[0021] 실시예들 중에서, 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법은 특정 프로그램의 소스코드를 분석하여 프로그램 의존 그래프를 생성하는 단계; 상기 소스코드 및 상기 특정 프로그램에 관한 요약문을 기초로 적어도 하나의 어휘사전을 생성하는 단계; 상기 소스코드에 대응되는 소스코드 임베딩을 생성하는 단계; 상기

소스코드 임베딩을 기초로 상기 프로그램 의존 그래프의 노드들에 관한 제1 노드 임베딩을 생성하는 단계; 상기 제1 노드 임베딩 및 상기 프로그램 의존 그래프의 간선 정보를 이용하여 그래프 어텐션(graph attention)을 수행하는 단계; 상기 그래프 어텐션에 따른 노드들의 제2 노드 임베딩 및 상기 요약문을 기초로 요약 어휘들에 관한 요약 어휘 임베딩을 생성하는 단계; 및 상기 요약 어휘 임베딩을 기초로 상기 소스코드에 관한 소스코드 요약문을 생성하는 단계;를 포함한다.

[0022] 상기 소스코드 임베딩을 생성하는 단계는 상기 소스코드의 입력 토큰들에 대한 워드 임베딩을 통해 각 입력 토큰 별로 각각이 해당 토큰의 의미 정보와 위치 정보를 포함하는 토큰 임베딩 및 위치 임베딩을 생성하는 단계; 및 상기 토큰 임베딩 및 상기 위치 임베딩 간의 합 연산을 통해 임베딩 벡터를 생성하는 단계를 포함할 수 있다.

[0023] 상기 제1 노드 임베딩을 생성하는 단계는 셀프 어텐션(self-attention) 과정을 통해 상기 소스코드 임베딩을 기초로 상기 입력 토큰들의 어텐션을 산출하여 상기 소스코드 내 자기학습을 강화하는 단계; 및 피드 포워드(feed forward) 과정을 통해 상기 셀프 어텐션 과정의 결과로서 획득된 노드 어텐션 벡터를 완전 연결 계층에 입력하여 노드 잠재표현 벡터를 생성하는 단계를 포함할 수 있다.

[0024] 상기 그래프 어텐션을 수행하는 단계는 상기 프로그램 의존 그래프의 간선 정보를 기초로 상기 제1 노드 임베딩 간의 합성 곱을 반복적으로 수행하는 단계를 포함할 수 있다.

[0025] 상기 소스코드 요약문을 생성하는 단계는 상기 요약 어휘 임베딩을 활성화 함수에 적용한 결과를 기초로 상기 소스코드 요약문을 생성하는 단계를 포함할 수 있다.

발명의 효과

[0027] 개시된 기술은 다음의 효과를 가질 수 있다. 다만, 특정 실시예가 다음의 효과를 전부 포함하여야 한다거나 다음의 효과만을 포함하여야 한다는 의미는 아니므로, 개시된 기술의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.

[0028] 본 발명에 따른 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치 및 방법은 기존의 트랜스포머 모델을 기반으로 그래프 처리 모델을 적용하여 프로그램 소스코드에 대한 자연어 요약문을 효과적으로 생성할 수 있다.

[0029] 본 발명에 따른 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 장치 및 방법은 트랜스포머 기반의 code-to-sequence 모델에 GCN 계층을 추가적으로 사용하여 소스코드의 시퀀스 정보뿐 아니라 그래프의 구조정보가 담긴 요약을 생성할 수 있다.

도면의 간단한 설명

[0031] 도 1은 본 발명에 따른 요약문 생성 시스템을 설명하는 도면이다.

도 2는 본 발명에 따른 요약문 생성 장치의 시스템 구성을 설명하는 도면이다.

도 3은 본 발명에 따른 요약문 생성 장치의 기능적 구성을 설명하는 도면이다.

도 4는 본 발명에 따른 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법을 설명하는 순서도이다.

도 5는 본 발명에 따른 요약문 생성 인공지능 프로그램을 설명하는 도면이다.

도 6은 본 발명에 따른 프로그램 의존 그래프를 설명하는 도면이다.

도 7은 본 발명에 따른 소스코드 토큰의 유형을 설명하는 도면이다.

도 8은 소스코드 입력에 따른 그래프 및 요약문 생성 과정의 일 실시예를 설명하는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0032] 본 발명에 관한 설명은 구조적 내지 기능적 설명을 위한 실시예에 불과하므로, 본 발명의 권리범위는 본문에 설명된 실시예에 의하여 제한되는 것으로 해석되어서는 아니 된다. 즉, 실시예는 다양한 변경이 가능하고 여러 가지 형태를 가질 수 있으므로 본 발명의 권리범위는 기술적 사상을 실현할 수 있는 균등물들을 포함하는 것으로 이해되어야 한다. 또한, 본 발명에서 제시된 목적 또는 효과는 특정 실시예가 이를 전부 포함하여야 한다거나 그러한 효과만을 포함하여야 한다는 의미는 아니므로, 본 발명의 권리범위는 이에 의하여 제한되는 것으로 이해

되어서는 아니 될 것이다.

[0033] 한편, 본 출원에서 서술되는 용어의 의미는 다음과 같이 이해되어야 할 것이다.

[0034] "제1", "제2" 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위한 것으로, 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.

[0035] 어떤 구성요소가 다른 구성요소에 "연결되어"있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결될 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어"있다고 언급된 때에는 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 한편, 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0036] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함하다"또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이며, 하나 또는 그 이상의 다른 특징이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0037] 각 단계들에 있어 식별부호(예를 들어, a, b, c 등)는 설명의 편의를 위하여 사용되는 것으로 식별부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않는 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0038] 본 발명은 컴퓨터가 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현될 수 있고, 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광 데이터 저장 장치 등이 있다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

[0039] 여기서 사용되는 모든 용어들은 다르게 정의되지 않는 한, 본 발명이 속하는 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한 이상적이거나 과도하게 형식적인 의미를 지니는 것으로 해석될 수 없다.

[0041] 도 1은 본 발명에 따른 요약문 생성 시스템을 설명하는 도면이다.

[0042] 도 1을 참조하면, 요약문 생성 시스템(100)은 사용자 단말(110), 요약문 생성 장치(130) 및 데이터베이스(150)를 포함할 수 있다.

[0043] 사용자 단말(110)은 사용자에 의해 운용되는 단말 장치에 해당할 수 있다. 예를 들어, 사용자는 사용자 단말(110)을 통해 특정 프로그램에 관한 소스코드와 요약문을 직접 작성할 수 있으며, 이에 관한 데이터를 입력하고 그 결과를 확인할 수 있다. 본 발명의 실시예에서 사용자는 하나 이상의 사용자로 이해될 수 있으며, 복수의 사용자들은 하나 이상의 사용자 그룹으로 구분될 수 있다.

[0044] 또한, 사용자 단말(110)은 요약문 생성 시스템(100)을 구성하는 하나의 장치로서 요약문 생성 장치(130)와 연동하여 동작하는 컴퓨팅 장치에 해당할 수 있다. 예를 들어, 사용자 단말(110)은 요약문 생성 장치(130)와 연결되어 동작 가능한 스마트폰, 노트북 또는 컴퓨터로 구현될 수 있으며, 반드시 이에 한정되지 않고, 태블릿 PC 등 포함하여 다양한 디바이스로도 구현될 수 있다. 또한, 사용자 단말(110)은 요약문 생성 장치(130)와 연동하기 위한 전용 프로그램 또는 어플리케이션(또는 앱, app)을 설치하여 실행할 수 있다.

[0045] 요약문 생성 장치(130)는 본 발명에 따른 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법을 수행하는 컴퓨터 또는 프로그램에 해당하는 서버로 구현될 수 있다. 또한, 요약문 생성 장치(130)는 사용자 단말(110)과 유선 네트워크 또는 블루투스, WiFi, LTE 등과 같은 무선 네트워크로 연결될 수 있고, 네트워크를 통해 사용자 단말(110)과 데이터를 송·수신할 수 있다.

[0046] 또한, 요약문 생성 장치(130)는 관련 동작을 수행하기 위하여 독립된 외부 시스템(도 1에 미도시함)과 연결되어 동작하도록 구현될 수 있다. 예를 들어, 요약문 생성 장치(130)는 포털 시스템, SNS 시스템, 클라우드 시스템

등과 연동하여 다양한 서비스를 제공하도록 구현될 수 있다.

[0047] 데이터베이스(150)는 요약문 생성 장치(130)의 동작 과정에서 필요한 다양한 정보들을 저장하는 저장장치에 해당할 수 있다. 예를 들어, 데이터베이스(150)는 소스코드와 이미지에 관한 정보를 저장할 수 있고, 학습을 위한 트랜스포머 모델이나 그래프 처리 모델에 관한 정보를 저장할 수 있으며, 반드시 이에 한정되지 않고, 요약문 생성 장치(130)가 본 발명에 따른 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법을 수행하는 과정에서 다양한 형태로 수집 또는 가공된 정보들을 저장할 수 있다.

[0048] 한편, 도 1에서, 데이터베이스(150)는 요약문 생성 장치(130)와 독립적인 장치로서 도시되어 있으나, 반드시 이에 한정되지 않고, 논리적인 저장장치로서 요약문 생성 장치(130)에 포함되어 구현될 수 있음은 물론이다.

[0050] 도 2는 본 발명에 따른 요약문 생성 장치의 시스템 구성을 설명하는 도면이다.

[0051] 도 2를 참조하면, 요약문 생성 장치(130)는 프로세서(210), 메모리(230), 사용자 입출력부(250) 및 네트워크 입출력부(270)를 포함할 수 있다.

[0052] 프로세서(210)는 본 발명에 따른 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 프로시저를 실행할 수 있고, 이러한 과정에서 읽혀지거나 작성되는 메모리(230)를 관리할 수 있으며, 메모리(230)에 있는 휘발성 메모리와 비휘발성 메모리 간의 동기화 시간을 스케줄 할 수 있다. 프로세서(210)는 요약문 생성 장치(130)의 동작 전반을 제어할 수 있고, 메모리(230), 사용자 입출력부(250) 및 네트워크 입출력부(270)와 전기적으로 연결되어 이들 간의 데이터 흐름을 제어할 수 있다. 프로세서(210)는 요약문 생성 장치(130)의 CPU(Central Processing Unit)로 구현될 수 있다.

[0053] 메모리(230)는 SSD(Solid State Disk) 또는 HDD(Hard Disk Drive)와 같은 비휘발성 메모리로 구현되어 요약문 생성 장치(130)에 필요한 데이터 전반을 저장하는데 사용되는 보조기억장치를 포함할 수 있고, RAM(Random Access Memory)과 같은 휘발성 메모리로 구현된 주기억장치를 포함할 수 있다. 또한, 메모리(230)는 전기적으로 연결된 프로세서(210)에 의해 실행됨으로써 본 발명에 따른 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법을 실행하는 명령들의 집합을 저장할 수 있다.

[0054] 사용자 입출력부(250)은 사용자 입력을 수신하기 위한 환경 및 사용자에게 특정 정보를 출력하기 위한 환경을 포함하고, 예를 들어, 터치 패드, 터치 스크린, 화상 키보드 또는 포인팅 장치와 같은 어댑터를 포함하는 입력장치 및 모니터 또는 터치 스크린과 같은 어댑터를 포함하는 출력장치를 포함할 수 있다. 일 실시예에서, 사용자 입출력부(250)은 원격 접속을 통해 접속되는 컴퓨팅 장치에 해당할 수 있고, 그러한 경우, 요약문 생성 장치(130)는 독립적인 서버로서 수행될 수 있다.

[0055] 네트워크 입출력부(270)은 네트워크를 통해 사용자 단말(110)과 연결되기 위한 통신 환경을 제공하고, 예를 들어, LAN(Local Area Network), MAN(Metropolitan Area Network), WAN(Wide Area Network) 및 VAN(Value Added Network) 등의 통신을 위한 어댑터를 포함할 수 있다. 또한, 네트워크 입출력부(270)는 데이터의 무선 전송을 위해 WiFi, 블루투스 등의 근거리 통신 기능이나 4G 이상의 무선 통신 기능을 제공하도록 구현될 수 있다.

[0057] 도 3은 본 발명에 따른 요약문 생성 장치의 기능적 구성을 설명하는 도면이다.

[0058] 도 3을 참조하면, 요약문 생성 장치(130)는 프로그램 입력부(310), 그래프 생성부(330), 어휘사전 생성부(350), 소스코드 요약문 생성부(370) 및 제어부(390)를 포함할 수 있다.

[0059] 프로그램 입력부(310)는 특정 프로그램에 관한 소스코드 및 요약문을 입력받을 수 있다. 소스코드(source code)는 프로그래밍 언어로 작성된 원시코드에 해당할 수 있으며, 특정 목적 달성을 위해 일련의 동작들을 정의하는 프로그래밍 언어의 시퀀스에 해당할 수 있다. 요약문은 특정 프로그램의 목적 및 동작을 설명하는 텍스트로서 자연어들의 집합에 해당할 수 있다. 프로그램 입력부(310)는 데이터베이스(150)에 저장된 소스코드와 요약문을 독출하여 메모리(230)에 저장할 수 있으며, 이를 기초로 다음 단계의 동작들이 수행될 수 있다. 또한, 프로그램 입력부(310)는 사용자 단말(110)과 연동하여 사용자에 의해 직접 작성되거나 또는 입력되는 소스코드와 요약문을 수신할 수도 있다.

[0060] 그래프 생성부(330)는 소스코드를 분석하여 간선과 노드로 구성된 프로그램 의존 그래프(Program Dependence Graph, PDG)를 생성할 수 있다. 프로그램 의존 그래프는 노드(node)와 간선(edge)으로 구성될 수 있으며, 노드 사이의 데이터 흐름 또는 제어 흐름을 표현하기 위해 방향 간선(directed edge)을 갖는 방향 그래프로 표현될 수 있다.

[0061] 여기에서, 프로그램 의존 그래프는 소스코드의 데이터 플로우(data flow)와 제어 플로우(control flow)에 따라 생성되는 그래프에 해당할 수 있다. 이때, 제어 플로우는 제어 의존 관계에 대응될 수 있고, 데이터 플로우는 데이터 의존 관계에 대응될 수 있다. 제어 의존 관계는 프로그램 실행 시의 순차적 진행 과정을 정의하기 위한 것으로, 예를 들어, B코드의 실행 여부가 A코드에 따라 결정되는 경우 B는 A에 의존관계가 있는 것으로 결정될 수 있다. 데이터 의존 관계는 제어 흐름에 영향을 받는 변수들 사이의 관계에 해당할 수 있으며, 프로그램의 실행순서에 따라 각 실행문에서 사용된 데이터의 흐름을 표현할 수 있다.

[0062] 일 실시예에서, 그래프 생성부(330)는 소스코드의 코드 블록을 노드로 구성하고 데이터 플로우 또는 제어 플로우로 정의되는 노드 간의 연결을 간선으로 구성할 수 있다. 보다 구체적으로, 그래프 생성부(330)는 소스코드를 코드 블록 단위로 분해할 수 있으며, 코드 블록은 특정 연산을 수행하는 소스코드의 부분코드로서 프로그램을 구성하는 프로시저에 대응될 수 있다. 그래프 생성부(330)는 소스코드 분석을 통해 도출된 코드 블록에 대응하여 노드를 생성할 수 있고, 코드 블록 간의 제어 플로우 또는 데이터 플로우를 기초로 노드 간의 연결을 간선으로 구성할 수 있다.

[0063] 한편, 프로그램 의존 그래프의 노드들과 제어 플로우에 대응되는 간선들은 소스코드에 관한 제어 흐름 그래프(Control Flow Graph, CFG)로 표현될 수 있으며, 프로그램 의존 그래프의 노드들과 데이터 플로우에 대응되는 간선들은 소스코드에 관한 데이터 흐름 그래프(Data Flow Graph, DFG)로 표현될 수 있다.

[0064] 어휘사전 생성부(350)는 소스코드 및 요약문을 기초로 적어도 하나의 어휘사전을 생성할 수 있다. 소스코드는 변수, 변수 유형, 키워드, 특수문자, 함수, 리터럴(literal) 등 다양한 유형의 토큰들을 포함할 수 있다. 어휘사전 생성부(350)는 입력받은 소스코드와 요약문을 분석하여 소스코드 요약문 생성 과정에 사용되는 어휘사전을 생성할 수 있다. 이때, 어휘사전에는 소스코드 어휘사전과 요약문 어휘사전이 포함될 수 있다. 소스코드 어휘사전은 소스코드에서 추출된 토큰(token)들의 집합으로 정의될 수 있고, 요약문 어휘사전은 요약문에서 추출된 워드(word)들의 집합으로 정의될 수 있다. 어휘사전 생성부(350)에 의해 생성된 어휘사전들은 이후 단계의 동작 과정에서 활용될 수 있다.

[0065] 일 실시예에서, 어휘사전 생성부(350)는 최대 빈도수를 기준으로 어휘사전에 있는 어휘 개수를 상위 N 개로 제한할 수 있다. 이때, N은 소스코드의 크기에 비례할 수 있다. 소스코드에 포함된 변수는 프로그래머의 지정 어휘에 따라 달라질 수 있다. 즉, 프로그램의 소스코드마다 서로 상이한 변수들이 정의되어 사용될 수 있으며, 이는 프로그램의 분석에 있어 구조적 정보를 고려하는데 방해가 될 수 있다. 따라서, 어휘사전 생성부(350)는 어휘의 전체 개수를 어휘의 최대 빈도수를 기준으로 제한할 수 있다. 이에 따라, 최대 빈도수를 기준으로 상위 N 개의 어휘들만 어휘사전에 포함될 수 있다. 이때, N은 소스코드의 크기에 비례하여 가변적으로 설정될 수 있다.

[0066] 소스코드 요약문 생성부(370)는 동작 과정에서 적어도 하나의 어휘사전이 적용되는 인코더(Encoder)와 디코더(Decoder)를 포함하고, 소스코드에 대응되는 소스코드 임베딩을 기초로 노드의 어텐션을 산출하여 소스코드 요약문을 생성할 수 있다. 여기에서, 소스코드 임베딩은 소스코드에 대응하는 벡터에 해당할 수 있으며, 소스코드의 토큰별 임베딩을 기초로 1차원 벡터로 표현될 수 있다. 또한, 소스코드 요약문 생성부(370)는 트랜스포머 모델을 기반으로 구현되는 인코더와 디코더를 포함할 수 있다. 소스코드 요약문 생성부(370)는 인코더를 통해 소스코드에 대응되는 특징을 벡터로 표현하고 디코더를 통해 벡터 표현에 대응되는 요약문을 생성할 수 있다. 특히, 소스코드 요약문 생성부(370)는 인코더에 그래프 정보를 학습하는 GCN 계층을 추가하여 인코더를 통해 소스코드에 대응하는 잠재표현을 생성하고, 디코더를 통해 노드의 어텐션에 따른 소스코드와의 관계를 분석하여 요약문을 출력할 수 있다.

[0067] 일 실시예에서, 소스코드 요약문 생성부(370)는 소스코드의 입력 토큰들에 대한 워드 임베딩을 통해 각 입력 토큰 별로 각각이 해당 토큰의 의미 정보와 위치 정보를 포함하는 토큰 임베딩 및 위치 임베딩을 생성하고, 토큰 임베딩 및 위치 임베딩 간의 합 연산에 의해 생성된 임베딩 벡터를 인코더의 입력으로 사용할 수 있다.

[0068] 먼저, 소스코드 요약문 생성부(370)는 입력 토큰과 위치 값을 인코더 신경망에 입력으로 제공하기 위해 워드 임베딩을 통해 입력 토큰에 대응되는 벡터를 생성할 수 있다. 토큰 임베딩은 소스코드의 각 입력 토큰에 대응되는 임베딩 벡터에 해당할 수 있으며, 위치 임베딩은 임베딩 벡터에 위치 정보를 추가하기 위한 포지셔널 인코딩(positional encoding)값에 해당할 수 있다. 일 실시예에서, 위치 임베딩은 사인(sin) 또는 코사인(cos) 함수를 통해 생성될 수 있다.

[0069] 이후, 소스코드 요약문 생성부(370)는 토큰 임베딩 및 위치 임베딩 간의 합 연산에 의해 생성된 임베딩 벡터를 인코더의 입력으로 사용할 수 있다. 예를 들어, 토큰의 길이가 m인 입력 토큰 $x = (x_1, x_2, \cdots, x_m)$에 대해 x의

토큰 임베딩이 $e_x = (e_{x1}, e_{x2}, \cdots, e_{xm})$이고, 위치 정보 p = (1, 2, …, m)에 대해 p의 위치 임베딩이 $e_p=(e_1, e_2, \cdots, e_m)$인 경우, 소스코드 요약문 생성부(370)는 다음의 수학식 1과 같이 두 벡터를 더해준 다음 최종적으로 인코더 신경망의 입력으로 사용할 수 있다.

[0070] [수학식 1]

[0071] $$e_x + e_p = (e_{x_1} + e_1, e_{x_2} + e_2, \ldots, e_{x_m} + e_m)$$

[0073] 일 실시예에서, 소스코드 요약문 생성부(370)는 소스코드 임베딩을 기초로 입력 토큰들의 어텐션을 산출하여 소스코드 내 자기학습을 강화하는 셀프 어텐션(self-attention) 과정과, 셀프 어텐션 과정의 결과로서 획득된 노드 어텐션 벡터를 완전 연결 계층에 입력하여 노드 잠재표현 벡터를 생성하는 피드 포워드(feed forward) 과정을 포함하는 트랜스포머 과정을 수행할 수 있다. 소스코드 요약문 생성부(370)는 트랜스포머 모델 기반의 인코더와 디코더를 포함할 수 있고, 트랜스포머 과정은 인코더 내부에서 수행될 수 있으며 셀프 어텐션 과정과 피드 포워드 과정을 포함할 수 있다.

[0074] 보다 구체적으로, 인코더 내의 트랜스포머는 셀프 어텐션(self-attention) 계층과 완전연결(Fully connected) 계층으로 구성될 수 있다. 이때, 셀프 어텐션 계층을 통해 상기의 셀프 어텐션 과정이 수행될 수 있으며, 완전연결 계층을 통해 상기의 피드 포워드 과정이 수행될 수 있다.

[0075] 셀프 어텐션 계층은 입력된 토큰 간 어텐션(attention)을 계산하여 소스코드 내 자기학습을 강화할 수 있다. 예를 들어, Q, K, V는 가중치가 곱해진 소스코드의 벡터(즉, 소스코드 임베딩)에 해당할 수 있으며, 자기학습이기에 모두 동일한 소스코드를 표현할 수 있다. Q와 K는 내적 후 소프트맥스(softmax)를 통해 활성화 정도가 산출될 수 있다. 그 후, 각 토큰은 자신의 활성화 정보와 연산되어 노트 어텐션 벡터를 출력할 수 있으며, 다음의 수학식 2와 같이 표현될 수 있다.

[0076] [수학식 2]

[0077] $$Attention(Q, K, V) = Softmax(\frac{QK^\top}{\sqrt{d_k}})V$$

[0079] 여기에서, Q, K 및 V는 각각 쿼리(Query), 키(Key), 값(Value)이고, d_k는 K벡터의 크기(차원수)이며, $K^\top$는 K 행렬의 전치행렬이다.

[0080] 또한, 셀프 어텐션 계층에서 출력된 노드 어텐션 벡터 o는 완전연결 계층에 입력될 수 있다. 완전연결 계층은 노드 어텐션 벡터 o를 학습 가중치 W_1과 연산한 후 ReLU를 통해 활성화할 수 있다. 완전연결 계층은 활성화된 정보를 다시 한번 학습 가중치 W_2와 연산한 후 입력과 같은 피처(feature)를 갖는 벡터를 출력할 수 있으며, 다음의 수학식 3과 같이 표현될 수 있다.

[0081] [수학식 3]

[0082] $$FFNN(o) = MAX(0, oW_1 + b_1)W_2 + b_2$$

[0084] 여기에서, FFNN은 피드 포워드 신경망이고, o는 노드 어텐션이며, 벡터 b_1 및 b_2는 바이어스 벡터(bias vector)이다.

[0085] 완전연결 계층에 의해 출력된 모든 벡터 T는 신경망 계층을 거쳐 벡터 N으로 집약될 수 있으며, N은 각 코드 블럭의 집약된 정보이자 하나의 노드를 의미할 수 있다. 즉, 다음의 수학식 4와 같이 표현될 수 있으며, N은 노드 잠재표현 벡터에 해당할 수 있다.

[0086] [수학식 4]

[0087] $$N = TW + b$$

[0089] 일 실시예에서, 소스코드 요약문 생성부(370)는 그래프 합성곱 과정에서 프로그램 의존 그래프의 간선 정보를

기초로 노드 잠재표현 벡터 간의 합성 곱을 반복적으로 수행하여 노드의 어텐션을 산출할 수 있다. 즉, 인코더 내의 트랜스포머 과정의 출력은 그래프 합성곱 과정의 입력으로 연결될 수 있다. 이에 따라, 소스코드 요약문 생성부(370)는 프로그램 의존 그래프의 간선 정보를 이용하여 각 노드에 대응되는 노드 잠재표현 벡터들 간의 연산을 수행할 수 있다.

[0090] 보다 구체적으로, M은 프로그램 의존 그래프의 간선(edge) 정보를 갖는 행렬(matrix)에 해당할 수 있으며, $M \in R^{node\ num \times node\ num}$의 형상을 가질 수 있다. 간선 정보가 1인 노드들의 관계는 '이웃(neighbor)'으로 결정될 수 있고, 해당 노드는 자신의 이웃 노드 $h_{neighbor}$들과 더해질 수 있다. 한편, 합성곱 과정에서 마스크(mask)를 통해 노드 값에 M 정보가 반영될 수 있으며, 데이터의 특징에 따라 합성곱 연산이 반복적으로 수행될 수도 있다. 해당 동작은 다음의 수학식 5와 같이 표현될 수 있다.

[0091] [수학식 5]

[0092] $$h = mask\left(sum(h_{neighbor}, h)\right)V$$

[0094] 여기에서, h는 특정 노드의 임베딩이고, $h_{neightbor}$은 노드 h의 이웃 노드의 임베딩이다.

[0095] 일 실시예에서, 소스코드 요약문 생성부(370)는 노드의 어텐션과 요약문을 디코더에 입력하여 요약 어휘 임베딩을 생성할 수 있다. 여기에서, 노드의 어텐션은 노드에 관한 어텐션 벡터에 해당할 수 있으며, 요약 어휘 임베딩은 소스코드 요약문 생성에 사용될 가능성이 있는 요약 어휘들의 임베딩 벡터에 해당할 수 있다. 소스코드 요약문 생성부(370)는 디코더를 통해 요약 어휘 별로 요약 어휘 임베딩을 생성할 수 있다.

[0096] 한편, 디코더는 멀티 헤드 셀프 어텐션 계층(multi head self-attention), 멀티 헤드 어텐션(multi head attention) 계층 및 피드 포워드(feed forward) 계층으로 구성될 수 있다. 멀티 헤드 셀프 어텐션 계층은 인코더의 셀프 어텐션 과정과 동일하게 수행될 수 있다. 멀티 헤드 어텐션 계층은 멀티 헤드 어텐션 과정을 수행할 수 있다. 피드 포워드 계층은 인코더의 피드 포워드 과정과 동일하게 수행될 수 있다.

[0097] 일 실시예에서, 소스코드 요약문 생성부(370)는 노드의 어텐션을 디코더의 멀티 헤드 어텐션 과정의 입력으로 사용할 수 있다. 디코더는 자연어 토큰과 인코더 출력을 입력으로 수신할 수 있으며, 자연어 토큰은 요약문에서 추출된 토큰들에 해당할 수 있다. 한편, 자연어 토큰은 띄어쓰기를 기준으로 추출될 수 있다. 디코더 또한 인코더와 동일하게 요약문에 관한 토큰 임베딩 및 위치 임베딩을 기초로 생성된 임베딩 벡터를 입력으로 수신할 수 있으며, 경우에 따라 임베딩 벡터는 행렬로 표현될 수 있다. 멀티 헤드 어텐션 과정은 K 및 Q에 대해 인코더의 출력, 즉 노드의 어텐션을 사용하는 반면, V는 자연어 토큰으로부터 생성된 잠재표현(또는 잠재표현 벡터)을 사용할 수 있다. 또한, 멀티 헤드 어텐션 과정은 헤드(head) 개수만큼 다수의 어텐션들을 동시에 학습하고 그 결과들을 서로 연결하는 과정을 통해 출력을 생성할 수 있다.

[0098] 일 실시예에서, 소스코드 요약문 생성부(370)는 요약 어휘 임베딩을 활성화 함수에 적용한 결과를 기초로 소스코드 요약문을 생성할 수 있다. 소스코드 요약문 생성부(370)는 디코더로부터 출력된 요약 어휘 임베딩을 활성화 함수(예를 들어, softmax 함수)에 입력한 결과를 이용하여 요약 어휘들 중에서 최종 출력될 어휘들을 결정할 수 있다. 결과적으로, 소스코드 요약문 생성부(370)는 최종 출력 어휘들로 구성된 소스코드 요약문을 출력으로 생성할 수 있다.

[0099] 제어부(390)는 요약문 생성 장치(130)의 전체적인 동작을 제어하고, 프로그램 입력부(310), 그래프 생성부(330), 어휘사전 생성부(350) 및 소스코드 요약문 생성부(370) 간의 제어 흐름 또는 데이터 흐름을 관리할 수 있다.

[0101] 도 4는 본 발명에 따른 인공지능 분석 기반 프로그램 소스코드의 요약문 생성 방법을 설명하는 순서도이다.

[0102] 도 4를 참조하면, 요약문 생성 장치(130)는 프로그램 입력부(310)를 통해 특정 프로그램에 관한 소스코드 및 요약문을 입력받을 수 있다. 이때, 프로그램 입력부(310)는 사용자 단말(110)과 연결되어 사용자에 의해 직접 작성된 소스코드와 요약문을 수신할 수도 있다.

[0103] 요약문 생성 장치(130)는 그래프 생성부(330)를 통해 소스코드를 분석하여 간선과 노드로 구성된 프로그램 의존 그래프를 생성할 수 있다(단계 S410). 소스코드에 대응되는 프로그램 의존 그래프는 소스코드의 부분 코드에 해당하는 코드 블록을 노드로 표현하고 코드 블록 간의 데이터 흐름 또는 제어 흐름을 간선으로 표현함으로써 생성될 수 있다.

[0104] 요약문 생성 장치(130)는 어휘사전 생성부(350)를 통해 소스코드 및 요약문을 기초로 적어도 하나의 어휘사전을 생성할 수 있다(단계 S420). 적어도 하나의 어휘사전은 소스코드 어휘사전 및 요약문 어휘사전을 포함할 수 있다. 소스코드 어휘사전은 소스코드의 토큰들로 구성될 수 있으며, 요약문 어휘사전은 요약문의 단어들로 구성될 수 있다. 어휘사전 생성부(350)는 필요에 따라 어휘사전의 크기, 즉 어휘사전에 포함되는 어휘들의 개수를 제한적으로 적용할 수 있다.

[0105] 요약문 생성 장치(130)는 소스코드 요약문 생성부(370)를 통해 소스코드에 대응되는 소스코드 임베딩을 생성하여 인코더에 입력할 수 있다(단계 S430). 즉, 소스코드 요약문 생성부(370)는 트랜스포머 모델 기반으로 구현된 인코더와 디코더를 포함할 수 있으며, 인코더와 디코더는 동작 과정에서 적어도 하나의 어휘사전을 적용할 수 있다. 이때, 소스코드 임베딩은 1차원 벡터로 변화되어 인코더에 입력될 수 있다.

[0106] 요약문 생성 장치(130)는 소스코드 요약문 생성부(370)의 인코더를 통해 소스코드 임베딩을 기초로 프로그램 의존 그래프의 노드들에 관한 제1 노드 임베딩을 생성할 수 있다(단계 S440). 이때, 해당 과정은 인코더의 트랜스포머 과정에 해당할 수 있으며, 제1 노드 임베딩은 노드 잠재표현 벡터에 해당할 수 있으며, 소스코드 내에 존재하는 노드들의 임베딩에 해당할 수 있다.

[0107] 요약문 생성 장치(130)는 소스코드 요약문 생성부(370)의 인코더를 통해 제1 노드 임베딩 및 프로그램 의존 그래프의 간선 정보를 이용하여 그래프 어텐션(graph attention)을 수행할 수 있다(단계 S450). 이때, 해당 과정은 인코더의 그래프 합성곱 과정에 해당할 수 있다. 즉, 소스코드 내 존재하는 노드들의 임베딩을 기초로 그래프 합성곱 과정이 수행된 결과 그래프 어텐션에 관한 노드들의 임베딩이 갱신될 수 있다.

[0108] 요약문 생성 장치(130)는 소스코드 요약문 생성부(370)의 디코더를 통해 그래프 어텐션에 따른 노드들의 제2 노드 임베딩 및 요약문을 기초로 요약 어휘들에 관한 요약 어휘 임베딩을 생성할 수 있다(단계 S460). 이때, 제2 노드 임베딩은 인코더의 출력으로서 노드의 어텐션에 해당할 수 있으며, 디코더는 인코더의 출력과 요약문을 입력으로 수신하여 요약 어휘 임베딩을 출력을 위한 신경망 학습을 수행할 수 있다.

[0109] 이후, 요약문 생성 장치(130)는 소스코드 요약문 생성부(370)를 통해 요약 어휘 임베딩을 활성화 함수에 적용할 수 있고, 그 결과를 이용하여 소스코드에 관한 소스코드 요약문을 생성할 수 있다(단계 S470).

[0111] 도 5는 본 발명에 따른 요약문 생성 인공지능 프로그램을 설명하는 도면이다.

[0112] 도 5를 참조하면, 본 발명에 따른 요약문 생성 인공지능 방법은 프로그램을 통해 구현될 수 있다. 요약문 생성을 위한 인공지능 프로그램은 소스코드를 입력받아 자연어 요약문(510)을 출력하는 모델로 트랜스포머, 그래프 처리 모델인 GCN(graph convolutional network) 기반의 code2seq모델에 해당할 수 있다. 기존의 트랜스포머 모델은 두 개의 순환신경망이 각각 인코더-디코더의 역할을 수행한 반면, 본 발명에 따른 모델은 인코더(Encoder)에 그래프 정보를 학습하는 GCN 계층(도 5의 PDG GCN)이 추가된 형태로 구현될 수 있다. 즉, 입력된 소스코드는 1차원의 벡터로 변환되어 인코더(Encoder)에 입력될 수 있으며, 인코더(Encoder)는 학습된 잠재표현을 출력할 수 있다. 디코더(Decoder)는 인코더(Encoder)의 출력을 입력받아 소스코드와의 관계를 학습한 후 소스코드에 대응되는 요약문(510)을 출력할 수 있다.

[0114] 도 6은 본 발명에 따른 프로그램 의존 그래프를 설명하는 도면이다.

[0115] 도 6을 참조하면, 요약문 생성 장치(130)는 그래프 생성부(330)를 통해 소스코드를 분석하여 간선(630)과 노드(610)로 구성된 프로그램 의존 그래프를 생성할 수 있다. 프로그램 의존 그래프는 도 6과 같이 소스코드의 데이터 흐름(data flow)과 제어 흐름(control flow)에 따라 그래프를 생성할 수 있다. 그래프는 노드(610)들과 노드(610) 간의 간선(630)으로 구성될 수 있으며, 노드(610)는 부분 소스코드를 의미하고 간선(630)은 프로그램의 작동에 따라 부분 소스코드 간의 연결을 의미할 수 있다.

[0116] 또한, 간선(630)은 데이터 흐름과 제어 흐름을 나타내는 간선으로 분류될 수 있다. 데이터 흐름은 이전에 사용된 데이터의 노드를 기준으로 데이터가 다른 노드에서 사용되는 것을 의미할 수 있다. 제어 흐름은 소스코드가 실행되기 위해 거쳐야 하는 코드 흐름을 의미할 수 있다. 그래프 생성부(330)는 생성된 노드(610)를 기준으로 소스코드를 분리할 수 있고, 분리된 소스코드 간 간선(630) 정보를 데이터로 생성할 수 있다. 이후, 간선 데이터는 GCN 계층의 합성곱(convolution) 레이어에서 연결 노드 간 정보의 연산을 수행하기 위해 사용될 수 있다.

[0118] 도 7은 본 발명에 따른 소스코드 토큰의 유형을 설명하는 도면이다.

[0119] 도 7을 참조하면, 특정 프로그램에 관한 소스코드(710)는 변수, 변수 유형, 키워드, 특수문자, 함수 및 리터럴과 같은 다양한 종류의 토큰으로 구성될 수 있다. 이 중에서 변수 유형, 특수문자 등은 서로 다른 소스코드에서

공용으로 사용되는 어휘에 해당할 수 있다. 하지만, 변수는 프로그래머의 지정 어휘에 따라 달라질 수 있다. 즉, 소스코드(710)에 따라 무분별한 변수가 정의될 수 있고, 이는 프로그램의 분석에서 구조적 정보를 고려하는 데 방해가 될 수 있다. 따라서, 어휘의 전체 수는 어휘의 최대 빈도수를 기준으로 제한될 수 있다.

[0121] 도 8은 소스코드 입력에 따른 그래프 및 요약문 생성 과정의 일 실시예를 설명하는 도면이다.

[0122] 도 8을 참조하면, 요약문 생성 장치(130)는 특정 프로그램에 관한 소스코드(810)를 입력으로 수신할 수 있으며, 소스코드(810)를 분석하여 소스코드 요약문(850)을 생성할 수 있다.

[0123] 도 8에서, 소스코드(810)는 Main 함수를 정의하는 코드에 해당할 수 있고, 2번째 라인의 'System.out.println(“Hello World”);'은 java 표준 입출력 크래스인 System이라는 클래스의 out이라는 객체를 이용해서 println()메소드를 호출하고, 해당 메소드를 통해 괄호 안에 있는 입력값인 “Hello World”라는 텍스트를 화면에 출력하는 동작을 정의하는 부분 소스코드에 해당할 수 있다.

[0124] 또한, 3번째 라인의 'Boolean pasCall = true'는 Boolean 형 변수 passCall을 선언하고 true 값을 할당하는 동작을 정의하고, 4번째 라인의 'if(pasCall == true){'은 변수 pasCall의 값이 true 값인지를 비교하는 동작을 정의하며, 5번째 라인의 'pasCall = false'은 변수 pasCall에 false 값을 할당하는 동작을 정의할 수 있다.

[0125] 이에 따라, 요약문 생성 장치(130)는 소스코드(810)에 관한 소스코드 요약문(850)으로서 “change Boolean value if Boolean value is true”라는 텍스트를 출력할 수 있다.

[0126] 또한, 요약문 생성 장치(130)는 소스코드 요약문(850) 생성을 위하여 소스코드(810)에 대응되는 프로그램 의존 그래프(830)를 생성할 수 있다. 도 8에서, 요약문 생성 장치(130)는 소스코드(810)에서 3 ~ 5 번째 라인들의 코드 블록에 대응되는 노드들(831, 832 및 833)을 생성하고 노드들 사이를 데이터 흐름에 따른 간선(834)과 제어 흐름에 따른 간선(835)으로 연결하는 과정을 통해 프로그램 의존 그래프(830)를 생성할 수 있다.

[0128] 상기에서는 본 발명의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

부호의 설명

[0130] 100: 요약문 생성 시스템

110: 사용자 단말 130: 요약문 생성 장치

150: 데이터베이스

210: 프로세서 230: 메모리

250: 사용자 입출력부 270: 네트워크 입출력부

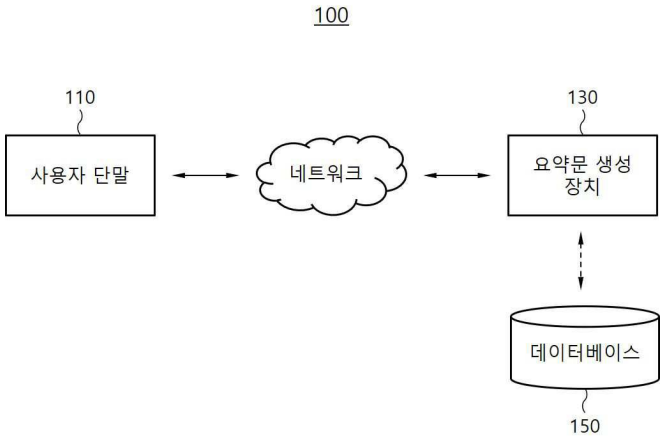
310: 프로그램 입력부 330: 그래프 생성부

350: 어휘사전 생성부 370: 소스코드 요약문 생성부

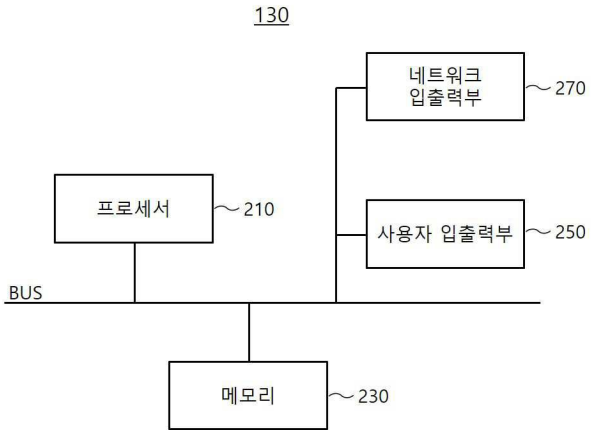
390: 제어부

도면

도면1

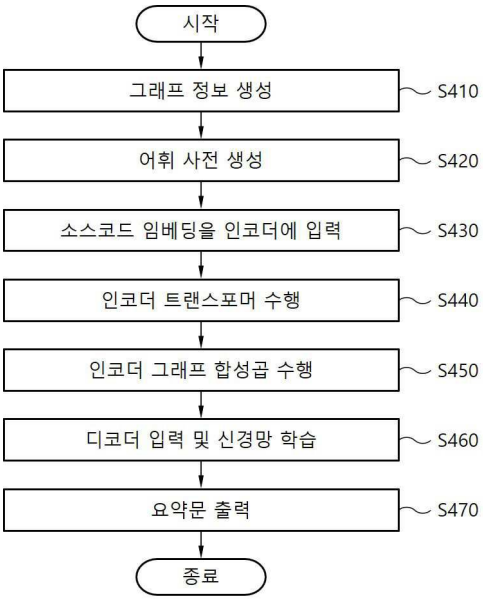


도면2

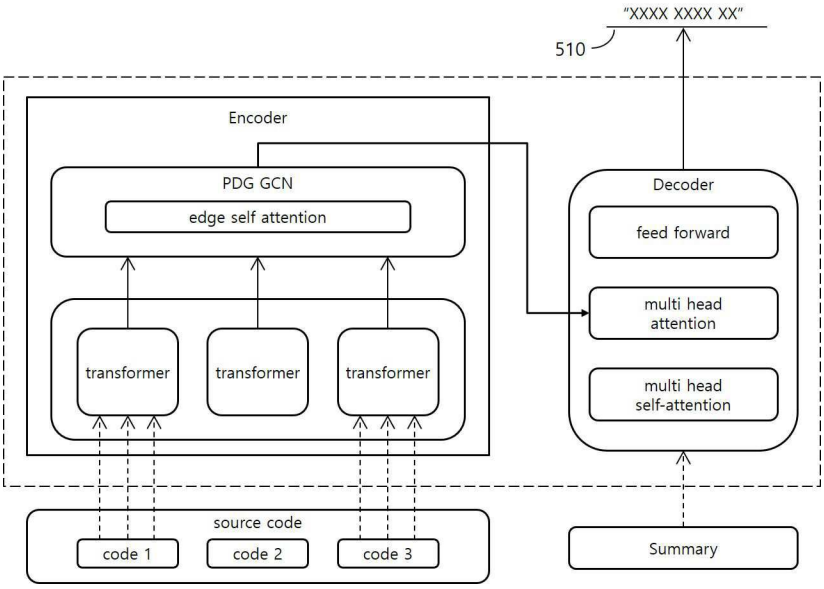


도면3

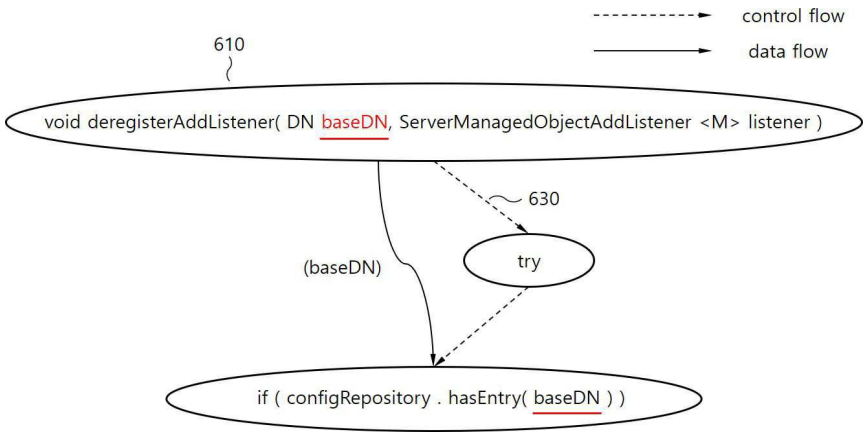
도면4


시작
그래프 정보 생성
S410
어휘 사전 생성
S420
소스코드 임베딩을 인코더에 입력
S430
인코더 트랜스포머 수행
S440
인코더 그래프 합성곱 수행
S450
디코더 입력 및 신경망 학습
S460
요약문 출력
S470
종료

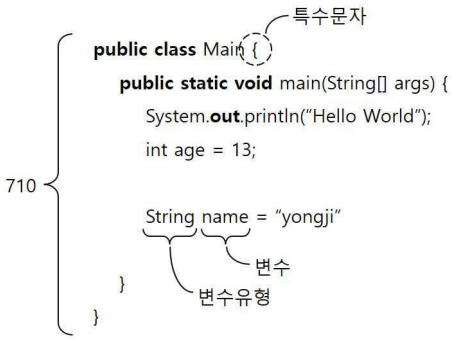
도면5


"XXXX XXXX XX"
510
Encoder
PDG GCN
edge self attention
transformer
transformer
transformer
Decoder
feed forward
multi head attention
multi head self-attention
source code
code 1
code 2
code 3
Summary

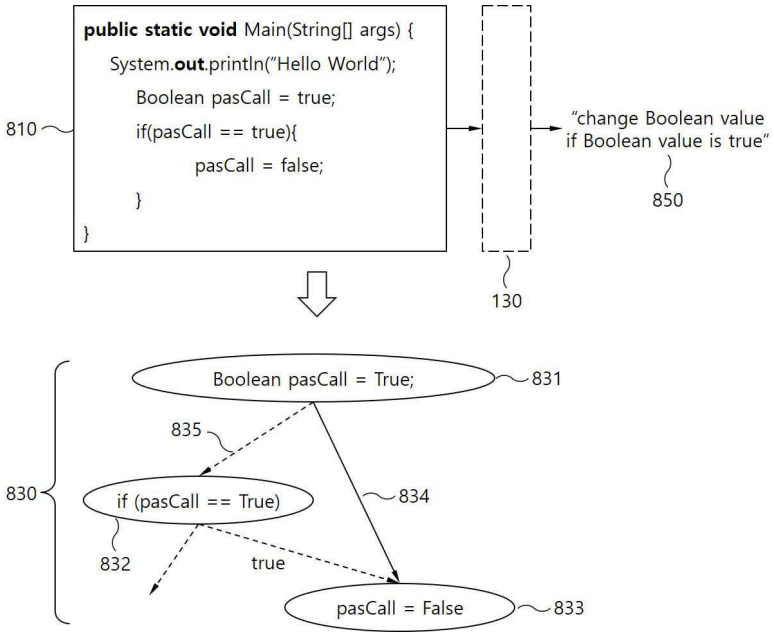
도면6


control flow
data flow
610
void deregisterAddListener(DN baseDN, ServerManagedObjectAddListener <M> listener)
630
try
(baseDN)
if (configRepository . hasEntry(baseDN))

도면7


특수문자
public class Main {
public static void main(String[] args) {
System.out.println("Hello World");
int age = 13;
String name = "yongji"
변수
변수유형
}
}
710

도면8


public static void Main(String[] args) {
System.out.println("Hello World");
Boolean pasCall = true;
if(pasCall == true){
pasCall = false;
}
}
810
"change Boolean value
if Boolean value is true"
850
130
Boolean pasCall = True;
831
835
834
if (pasCall == True)
832
true
pasCall = False
833
830