(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0019245
(43) 공개일자 2012년03월06일

(51) 국제특허분류(Int. Cl.)
G06F 17/27 (2006.01) *G06F 17/21* (2006.01)
G06F 17/10 (2006.01)
(21) 출원번호 10-2010-0082549
(22) 출원일자 2010년08월25일
심사청구일자 2010년08월25일

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
김선호
서울특별시 양천구 목동서로 130, 목동신시가지아파트 404-805 (목동)
윤준태
서울특별시 양천구 목동서로 70, 목동아파트 220동 1508호 (목동)
(뒷면에 계속)
(74) 대리인
특허법인충현

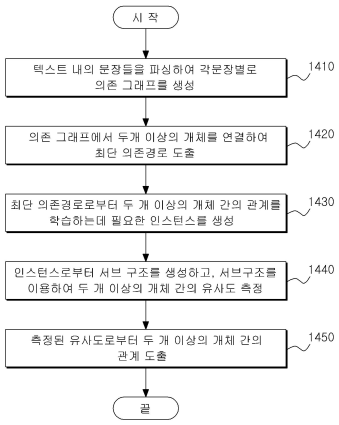
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 **바이오 텍스트 데이터로부터 개체 간의 관계를 도출하는 방법**

(57) 요 약

본 발명은 바이오 텍스트 데이터로부터 개체 간의 관계를 도출하는 방법에 관한 것으로서, 텍스트 내의 문장들을 파싱하여 각 문장별로 단어들 간의 문법적 의존 그래프를 생성하고, 생성된 의존 그래프에서 두 개 이상의 개체를 연결하는 최단 의존 경로를 도출하고, 최단 의존 경로로부터 두 개 이상의 개체 간의 관계를 학습하는데 필요한 인스턴스를 생성하고, 생성된 인스턴스들로부터 서브 구조들을 생성하고, 생성된 서브 구조들을 이용하여 두 개 이상의 개체 간의 유사도를 측정하고, 측정된 유사도로부터 두 개 이상의 개체 간의 관계를 도출함으로써, raw 텍스트 데이터로부터 양질의 개체 간의 관계를 도출하여 자동으로 새로운 정보를 업데이트하고, 추출할 수 있는 수단을 제공할 수 있다.

대 표 도 - 도14



(72) 발명자

박석

서울특별시 종로구 평창12길 16-6, 2동 202호 (평창동, 평창삼호빌라)

서정연

서울특별시 서초구 서초중앙로 206, 삼호가든 3차 B동 203호 (반포동)

이 발명을 지원한 국가연구개발사업

과제고유번호	핵06A1903
부처명	교육과학기술부
연구사업명	2단계 두뇌 한국(BK) 21 사업
연구과제명	PERVASIVE 컴퓨팅 기반의 의료시스템을 위한 S/W 인력 양성팀
주관기관	서강대학교 산학협력단
연구기간	2010.03.01 ~ 2011.02.28이 발명을 지원한 국가연구개발사업
과제고유번호	2009-0070211
부처명	교육과학기술부
연구사업명	기초 연구 지원 사업 여성 과학자 부문
연구과제명	자연어 처리를 이용한 바이오 텍스트 마이닝
주관기관	서강대학교 산학협력단
연구기간	2010.03.01 ~ 2011.02.28

특허청구의 범위

청구항 1

텍스트 내의 문장들을 파싱하여 각 문장별로 단어들 간의 문법적 의존 그래프를 생성하는 단계;

상기 생성된 의존 그래프에서 두 개 이상의 개체를 연결하는 최단 의존 경로를 도출하는 단계;

상기 최단 의존 경로로부터 상기 두 개 이상의 개체 간의 관계를 학습하는데 필요한 인스턴스를 생성하는 단계;

상기 생성된 인스턴스들로부터 서브 구조들을 생성하고, 상기 생성된 서브 구조들을 이용하여 상기 두 개 이상의 개체 간의 유사도를 측정하는 단계; 및

상기 측정된 유사도로부터 상기 두 개 이상의 개체 간의 관계를 도출하는 단계를 포함하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 2

제 1 항에 있어서,

상기 인스턴스는 상기 두 개 이상의 개체를 잇는 문장 내의 단어들과 상기 단어들 간의 문법적 의존 관계들을 스트링으로 나타낸 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 3

제 1 항에 있어서,

상기 두 개 이상의 개체 간의 유사도를 측정하는 단계는,

상기 인스턴스들의 쌍을 유사도로 맵핑하는 유사도 함수인 커널을 이용하여 상기 두 개 이상의 개체 간의 유사도를 측정하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 4

제 1 항에 있어서,

상기 생성된 인스턴스들 간의 공통적인 서브 구조가 인접한 서브 스트링인 경우, 인접하지 않은 서브 스트링인 경우보다 더 많은 가중치를 부여하여 상기 두 개 이상의 개체 간의 유사도를 측정하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 5

제 1 항에 있어서,

상기 생성된 인스턴스들 간의 공통적인 서브 구조가 인접하지 않은 서브 스트링인 경우, 갭의 크기와는 상관없이 같은 가중치를 부여하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 6

제 1 항에 있어서,

상기 생성된 인스턴스들로부터 생성되는 서브 구조는 에지로 시작하여 에지로 끝나는 워크인 e-walk를 포함하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 7

제 6 항에 있어서,

버텍스로 시작하여 버텍스로 끝나는 워크인 v-walk보다 상기 e-walk에 더 많은 가중치를 부여하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 8

제 1 항에 있어서,

노드-에지, 에지-노드, 노드, 또는 에지 간의 매칭을 가능하도록 하는 부분 매칭을 허용하여 상기 두 개 이상의 개체 간의 공통적인 서브 구조를 생성하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 9

제 1 항에 있어서,

상기 최단 의존성 경로는 방향성이 있는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 10

제 1 항에 있어서,

상기 생성된 인스턴스들 간의 공통적인 서브 구조는 인접한 e-walk 서브 구조, 인접한 v-walk 서브 구조, 및 인접하지 않은 서브 구조를 포함하고, 각각의 서브 구조에 다른 가중치를 부여하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 11

제 1 항에 있어서,

상기 두 개 이상의 개체 간의 유사도를 측정하는 단계는,

상기 인스턴스들로부터 생성된 서브 구조들이 공통되는 정도에 따라서 상기 두 개 이상의 개체 간의 유사도를 측정하는 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 12

제 1 항에 있어서,

상기 개체는 단백질 개체인 것을 특징으로 하는 텍스트로부터 개체 간의 관계를 도출하는 방법.

청구항 13

제 1 항 내지 제 12 항 중에 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

명 세 서

기 술 분 야

[0001] 본 발명은 바이오 텍스트 데이터로부터 생물학 개체(biomedical named entity) 간의 관계를 도출하는 방법에 관한 것으로서, 더욱 상세하게는 워크(walk), 인접, 및 비인접 의존 경로에 따라 가중치를 다르게 부여하는 가중치 서브 시퀀스 커널을 이용하여 바이오 텍스트 데이터로부터 개체 간의 관계를 도출하는 방법에 관한 것이다.

배 경 기 술

[0002] 생물학 개체 중 하나인 단백질 간의 상호작용 네트워크를 구축하는 것은 여러 가지 인간의 생명 현상을 규명에 있어서 중요하다. 그러나, 많은 유용한 단백질 간의 관계들이 데이터베이스에 누락되어 있으며, 가공되지 않은 텍스트 내에 그대로 남아 있다. 따라서, 가공되지 않은 일반 텍스트에 숨어져 있는 단백질들 간의 상호작용 관계를 자동으로 도출하려는 연구가 바이오 인포매틱스 분야에서 중요한 이슈 중 하나이다.

[0003] 최근 생의학 분야는 기술적 진보와 인간 게놈 프로젝트의 성공에 따른 방대한 양의 게놈 데이터에 의해 급속히 확대, 발전하고 있으며, 그와 관련된 새로운 수많은 정보들이 쏟아져 나오고 있다. 또한 그에 따른 방대한 생의학 데이터의 수요를 감당하기 위해 다양한 데이터베이스가 개발되었다.

[0004] 하지만, 대부분의 데이터베이스들은 여전히 큐레이터에 의존하고 있다. 따라서 과학자와 큐레이터들이 기존 연구들의 새로운 발견에 쉽게 대응하고 단백질 간의 상호작용 정보를 도출하기 위해서는 고성능 정보 추출 마이닝

도구의 개발이 절실하다.

[0005] 생의학 문서에서 정보를 추출하는 것은 보통 문서상에 나타나는 단백질, 유전자, 병명 등과 같은 생물학 개체명들을 인식하고, 그들 간에 존재하는 여러 가지 관계를 추출하는 단계를 거친다. 여기서 개체 간에 존재하는 관계(relation)는 상호작용(interaction)으로도 해석할 수 있으며, 예를 들어 특정 병과 약물을 개체로 든다면 "효과 있음", "없음", "부작용", "관련 없음" 등이 두 개체 간의 상호작용의 예라고 볼 수 있다.

[0006] 정보 추출 중 바이오 텍스트에 존재하는 여러 가지 개체명을 인식하는 것은 상당한 발전이 있었던 반면에, 개체 간의 관계 또는 상호작용을 도출하는 것은 여전히 어려운 문제로 인식되고 있어, 이에 대한 연구가 필요한 실정이다.

발명의 내용

해결하려는 과제

[0007] 따라서, 본 발명이 해결하고자 하는 과제는 생물학 텍스트 데이터로부터 단백질 개체들 간의 관계를 자동으로 추출하는 다양한 커널 방법을 이용하여, 바이오 텍스트 데이터로부터 개체 간의 관계를 도출하는 방법을 제공하는 것이다.

[0008] 또한, 바이오 텍스트 데이터로부터 개체 간의 관계를 도출하는 방법을 컴퓨터상에서 실행 가능한 시스템을 제공하고자 한다.

과제의 해결 수단

[0009] 본 발명은 상기된 과제를 달성하기 위하여 생물학 텍스트 데이터로부터 각 문장을 파싱하여 단어들 간의 문법적 의존 그래프로 표현하고, 의존 그래프에서 두 개체를 연결하는 최단 의존 경로를 도출한 다음, 이로부터 관계 학습에 필요한 인스턴스(예제)를 추출한다.

[0010] 이러한 인스턴스들로부터 두 개체 간의 관계를 학습해 나갈 때 인스턴스들 간의 유사도 벡터를 이용하여 각 관계를 대표하는 지지 벡터를 생성하는 SVM (support vector machine)기법을 사용하며, 새로운 개체간 관계를 포함하는 그래프 입력에 대해 학습된 클래스를 할당한다. 본 발명의 실시예로서, 관계 클래스는 두 개체간이 TA(taget-agent) 관계로 상호작용 있음, AT(agent-target) 관계로 상호작용 있음, 상호작용 없음으로 구분하였는데 이에 제한적이지 않고 더 복잡한 관계 클래스를 가진 데이터에서도 학습이 가능하다.

[0011] 과제 해결의 주요한 수단은 유사도 커널이며, 커널 정의 시 기본이 되는 두 인스턴스 서브 그래프들 사이의 유사도는 두 그래프의 내부에 공통적인 부분 하위 구조(서브 구조)가 얼마나 같은 지에 의해서 파악된다.

[0012] 문장의 구조적인 면을 고려하여 유전적 상호 작용을 도출하는데 있어서 상호작용을 가지는 문장은 얼마나 공통적인 구조를 포함하는 지를 파악하여 문제 해결의 키로 사용한다.

[0013] 커널 정의시 매칭되는 하부 구조를 (1) e-walk (2) v-walk (3) 부분 매칭, (4) 비인접 경로로 구분하여 매칭시 각각의 다른 가중치를 부여하여 구조적 유사도를 계산하였다.

[0014] 본 발명의 실시 예에 의하면, 공통적인 서브 그래프에 가중치를 부여할 때 두 단어간의 관계를 표현하는 어휘적 워크인 v-walk 형태보다는 특정 단어의 구문적 워크를 표현하는 e-walk에 더 많은 가중치를 부여하는 것이 바람직하였으며, 인접한 서브 그래프와 서로 거리를 두고 떨어져서 연결된 서브 그래프의 가중치를 다르게 부여하는 것이 바람직하였다. 또한 부분 매칭을 허용하여 노드(단어)-에지(문법 관계) 또는 에지-노드 및 노드, 혹은 에지 간의 매칭도 가능하게 하여 공통적인 서브 그래프를 산출하였으며 최단 의존 경로에는 방향성이 존재하는 방향성 그래프 (directed graph)를 가정하는 것이 바람직하였다.

[0015] 상기 다른 기술적 과제를 해결하기 위하여, 본 발명은 상기된 텍스트 데이터로부터 개체 간의 관계를 도출하는 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체를 제공한다.

발명의 효과

[0016] 본 발명에 따르면, raw 텍스트 데이터로부터 양질의 개체 간의 관계를 도출하여 큐레이터 및 생의학자들에게 자동으로 새로운 정보를 업데이트하고, 추출할 수 있는 수단으로 활용하게 할 수 있다.

도면의 간단한 설명

[0017] 도 1은 개체들이 서로 멀리 떨어져 있는 문장들을 갖는 생물학 텍스트의 일례를 도시한 것이다.

도 2는 v-walk 커널에서 사용되는 서브 구조들을 도시한 것이다.

도 3은 e-walk 커널에서 사용되는 서브 구조들을 도시한 것이다.

도 4는 종래의 의존 커널에서 고려된 서브구조를 나타낸 것이다.

도 5는 확장된 의존 커널에서 고려된 서브구조를 나타낸 것이다.

도 6은 스펙트럼 커널에서의 서브 구조를 도시한 것이다.

도 7은 고정 길이 서브 시퀀스 커널에서의 서브 구조를 도시한 것이고, 도 8은 도 7의 일부 구조를 나타낸 것이다.

도 9는 주어진 문장에 대하여 개체 간의 의존 그래프를 도시한 것이다.

도 10은 "ywhE"와 "sigF"를 연결하는 최단 의존 경로를 나타낸다.

도 11은 도 10의 최단 의존 경로를 어휘 의존 경로 스트링과 구조 의존 경로 스트링으로 표현한 것이다.

도 12는 단어 의존 리스트와 품사 의존 리스트를 도시한 것이다.

도 13은 도 9에 도시된 문장의 최단 경로 스트링 인스턴스를 도시한 것이다.

도 14는 본 발명의 일 실시 예에 따른 텍스트로부터 개체 간의 관계를 도출하는 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0018] 본 발명은 상기된 과제를 달성하기 위하여 1) 생물학 텍스트 데이터로부터 각 문장을 파싱하여 단어들 간의 문법적 의존 그래프를 생성하는 단계; 2) 생성된 의존 그래프에서 두 개체를 연결하는 최단 의존 경로를 도출하는 단계; 3) 도출된 최단 의존 경로에서 유용한 하부 구조들을 표현하는 단계; 4) 최단 의존 경로들로부터 공통적인 서브 구조들을 이용하여 유사도 벡터를 만드는 단계로 이 단계는 커널 정의 단계로 표현할 수 있으며, 5) 유사도 벡터들을 이용하여 개체간 해당 관계를 대표하는 지지 벡터를 생성하는 SVM(support vector machine) 커널 생성/학습하는 단계; 6) 커널로부터 새로운 데이터(개체간의 의존 경로)에 대한 관계를 분류하는 단계를 포함한다.

[0019] 이때 유사도 커널 정의 단계인 4)의 경우, 두 서브 그래프 내부의 부분 구조가 얼마나 공통적인지를 측정하기 위해 그래프의 개체간 의존 경로를 스트링으로 표현하고 하부 구조들의 형태에 따라 서로 다르게 가중치를 부여하는 등 여러가지 커널을 정의하여 개체 간의 관계를 더욱 효과적으로 파악하는 방법을 제공한다.

[0020] 본 발명의 실시예에서는 1) 확장된 의존 커널, 2) 스펙트럼 커널, 3) 거리-기반 가중치 서브 시퀀스 커널 4) 워크-기반 가중치 서브 시퀀스 커널을 개시하고 있다.

[0021] 이하, 실시 예를 들어 본 발명을 더욱 상세하게 설명한다. 그러나 이들 실시 예는 본 발명을 보다 구체적으로 설명하기 위한 것으로, 본 발명의 범위가 단백질 개체간의 관계 추출 및 도식화된 예에 제한되지 않는다는 것을 명시한다.

[0022] 본 발명의 응용 범위는 생물학 개체간 관계 추출뿐만 아니라, 일반 텍스트 상의 개체간 관계 추출에도 확대 적용될 수 있다.

[0023] 도 1 내지 도 13은 여러 가지 커널 정의(유사도 계산) 시에 고려되는 그 내부적인 서브 구조들에 대해 보여준다.

[0024] 도 1은 개체가 멀리 떨어져 있는 문장들을 갖는 생의학 텍스트의 일례를 도시한 것이다.

[0025] 도 1의 윗 문장은 긴 범위에 걸쳐 두 개체간의 관계가 존재하는 복잡한 문장의 예이고, 도 1의 아래 그래프는 긴 범위에 걸친 관계를 갖는 문장의 최단 경로를 도시한 그래프이다. 최단 경로를 두꺼운 화살표로 도시하고 있다.

[0026] 특정 타입의 매칭 규칙 또는 패턴을 핵심적인 개체간 관계로 발견하기 위한 연산으로서, 일반적으로 문장을 단어, 품사(parts of speech), 또는 구문 요소(syntactic constituent)의 열 또는 시퀀스(sequence) 형태로 표현하여 관계를 잘 설명할 수 있는 열 규칙을 찾으려 시도하여 왔는데, 이러한 규칙에 기반한 관계 추출은 높은 정

확도를 가지고 직관적으로 쉬운 방법론을 제공할 수 있으나, 파악할 수 있는 범위가 지극히 제한적으로 낮은 리콜(recall rate)을 보이며, 의미적 또는 구문적인 의역(paraphrase)과 멀리 떨어져 있는 개체 간 관계를 파악해 내기가 어렵다.

[0027] 대안 중에 하나로서, 다양한 커널 방법들이 이러한 개체 간 관계를 추출하는데 사용될 수 있다. 특히, 이러한 커널 방법들은 구문적인 파싱 트리와 의존 구조와 같은 많은 구조적 데이터를 학습하는데 좋은 방법을 제시한다. 커널들은 보통 예제(instance)들이 공유하는 공통된 서브 구조들에 기초하여 예제들로부터 구조적 유사성을 도출하도록 고안된다.

[0028] 본 발명의 특이점은 기존 방법들과 다르게 공통된 서브 구조를 분류하고 유사도 계산시 그 특성을 반영한 가중치를 다르게 주어서 카운트를 계산한다는 점이다.

[0029] 이하에서는, 관련된 문장들을 정렬하여 더욱 일반화된 패턴 학습 방법의 일례를 제공하는 발명에 대해서 설명한다.

[0030] 본 발명은 인스턴스 간의 구조적 유사성 비교에 의한 개체간 관계를 학습해 나가는데 이러한 방법은 특성(feature) 벡터를 명시적으로 열거하지 않고, 복잡한 하부 구조간의 비교를 재귀적으로 계산하여 효과적인 연산을 시도할 수 있는 장점이 있다.

[0031] 본 발명에서 하나의 인스턴스는 문장 내에 등장하는 두 개체 사이의 최단 경로를 의미하며, 인스턴스는 두 개체를 잇는 문장 내의 단어들과 그들의 문법적 의존 관계들의 스트링으로 볼 수 있다. 이들 인스턴스로부터 단백질 개체들 사이 상호작용인 PPI(protein-protein interaction)를 도출하기 위하여 여러 가지 커널을 제시한다. 각 커널은 각각 다른 방식으로 구조적 유사성 정보를 도출하고, 문장 내의 유전자들 사이 또는 단백질 사이의 관계를 발견한다.

[0032] 1) 인스턴스 표현에 사용되는 데이터 구조를 먼저 설명하면 다음과 같다

[0033] 커널들은 모두 문장의 구조적 파싱 트리를 의존 그래프 형태로 표현하고 표현된 그래프로부터 두 개체를 연결하는 최단 거리 경로를 추출하여 개체간의 관계를 표현하는 방식의 데이터 구조를 사용한다.

[0034] 이때, 문장에 대한 파싱 그래프에서 단어/품사는 그래프의 노드(node)에 해당되고, 두 개의 노드 간의 문법적(구조적) 의존 관계는 그래프에서의 에지(edge)에 해당한다. 즉, 에지는 헤드 노드(단어)와 종속 노드(단어)의 문법적 의존 관계를 표현하며, 헤드 노드와 종속 노드 간의 의존 관계 타입은 다음과 같은 7개의 문법 관계로 표현하였다: appos(동격), comp_prep(전치사구), mod(수식어), mod_att(부가적 수식어), neg(부정어), obj(목적어), suj(주어).

[0035] 이때, 두 개체 사이의 관계를 모델링하기 위해 필수적인 구조를 방향성이 있는 최단 의존 경로로 표현한다. 이때, 두 개체를 이어주는 단일 경로와 개체 사이의 등위 접속사의 구문적인 의존까지 확대하여 고려한다.

[0036] 두 개의 개체가 접속사에 의하여 결합될 때, "NE1 conj_and NE2"와 같은 경우가 있기 때문에 최단 경로만 가지고 관계를 결정하는 것은 불충분하다.

[0037] 바인딩, 연관, 재조합, 또는 상호작용 문맥으로 동시에 일어나는 몇몇 경우를 제외하고 대부분의 경우에는 상호작용이 일어나지 않는다.

[0038] 따라서, 단일 최단 경로는 "interaction between ~ and ~", "association between ~ and ~", 및 "recombinant ~ and ~"와 같은 경로에서 단서가 되는 문맥 단어를 포함하도록 확장된다. 또한, 부정어 표현이 서술어 근방에 위치할 때, 경로 상에서 서술어 앞에 부정어 표시를 부가한다.

[0039] 2) 구조간 유사성을 파악하기 위해 커널에서 고려하고 있는 하부(서브) 구조들을 살펴보면 다음과 같다.

[0040] 첫 번째로 본 발명의 실시 예에서 기존의 의존 커널의 단점을 개선하기 위해 사용되는 서브 구조인 e-walk이다. 여기서 워크(walk)는 노드 혹은 버텍스(vertex)와 에지(edge)가 교대로 발생하는 열(sequence)로 정의될 수 있다.

[0041] 각 노드는 단어에 해당하며 에지는 연결된 단어 간의 문법적 관계를 의미한다. 즉, $v \in V$이고, $e \in E$라고 하

면, 워크는 다음의 형태로 v_i, $e_{i,i+1}$, v_{i+1}, $e_{i+1,i+2}$,..., v_{i+n-1}, 버텍스로 시작하여 버텍스로 끝나도록 표현할 수 있다.

[0042] 여기서, V는 버텍스(노드)의 집합, E는 에지(문법 관계)의 집합이고, i와 n은 임의의 자연수이다.

[0043] 이때, 버텍스로 시작하여 버텍스로 끝나는 워크를 버텍스 워크(이하, v-walk)라고 한다. 마찬가지로 하면, 에지로 시작하여 에지로 끝나는 워크를 에지 워크(이하, e-walk)라고 정의한다.

[0044] 즉, v-walk 특성은 (단어1, 문법 관계, 단어2) 또는 (품사1, 문법 관계, 품사2)로 구성되고, e-walk 특성은 (문법 관계1, 단어, 문법 관계2) 또는 (문법 관계1, 품사, 문법 관계2)로 구성된다.

[0045] 도 2는 v-walk 형태의 그래프 내의 서브 구조들을 도시한 것이다.

[0046] 도 3은 e-walk 형태의 그래프 내의 서브 구조들을 도시한 것이다.

[0047] 일반적으로 의존 커널에서는 개체간의 관계를 표현하고 있는 두 개의 관계 인스턴스들 간의 구조적 유사성은 서브 그래프에서의 v-walk (노드-에지-노드)를 비교함으로써, 재귀적으로 도출된다. 즉, 도 2에 도시된 v-walk만 비교되고, 도 3에 도시된 e-walk는 비교되지 않는다.

[0048] 그러나, 실험 결과 e-walk 특성은 실제적으로 v-walk보다 관계를 결정하는 데 있어서 더 중요한 역할을 수행하는 것을 알 수 있었다. 따라서 본 발명의 실시예에서는 이러한 특성을 커널 정의시 활용한다.

[0049] 도 11을 참조하면, "sub(UP)-control-comp_by(DN)"으로 표현되는 e-walk는 서술어 동사 "control"의 문장 성분 구조를 나타내는 것이다.

[0050] 이 경우에 서술어 동사 "control"의 하나의 문장 성분은 "주어"가 되는 것이고, 다른 하나의 문장 성분은 by로 표시되는 전치사 성분이다.

[0051] 만일 다른 두 개체 간의 관계를 표현하는 그래프의 e-walk가 서술어 "control"에 대하여 "sub(UP)-control-comp_by(DN)"의 구조로 되어 있다면, 이 구조 역시 두 개의 개체는 서로 유전적 관계를 가질 가능성이 클 것이다.

[0052] 의미적인 측면에서의 서술어와 수식어구(modifier) 간의 관계는 개체 간의 관계 추출에 있어서, 매우 도움이 된다. 따라서, 본 발명의 실시 예에서는 e-walk를 이용하여 문맥상의 구문 구조를 다루기로 한다.

[0053] 두 번째로 본 발명의 실시 예에서는 의존 커널의 단점을 개선하기 위해 하부 구조인 부분 경로 매칭을 사용한다.

[0054] 도 4는 종래의 의존 커널에서 고려된 서브구조를 나타낸 것이다.

[0055] 도 5는 확장된 의존 커널에서 고려된 서브구조를 나타낸 것이다. 도 5는 도 4의 구조를 포함하여 구성된다.

[0056] 도 5를 참조하면, 완전 경로 매칭(complete path match)에서는 "subj(UP) stimulate" 또는 "subj(UP) stimulate obj(DN)"과 같은 부분이 서브구조 비교에서 배제된다. 따라서, 두 개의 루트 노드들과 루트 노드들의 직접적인 자식 노드들이 같고, 노드들 사이의 의존 관계가 같을 때만 두 개의 서브 그래프는 매칭된다.

[0057] 완전 경로 매칭은 적어도 2개의 단어로 서로 연결된 서브 그래프로 구성된다. 그러나 두 개의 그래프 사이에서 일련의 노드-에지-노드가 매칭되는 경우는 드물다.

[0058] 따라서, 본 발명에서는 도 5에서와 같은 불완전한 그래프인 부분 구조를 추가적으로 고려하며, 이들에 대해서도 부분 경로 매칭(partial path match)을 허용하였다.

[0059] 세 번째로 본 발명의 실시 예에서는 의존 커널의 단점을 개선하기 위해 하부 구조인 비인접 경로를 사용한다.

[0060] 도 6은 스펙트럼 커널에서의 서브 구조를 도시한 것이고, 도 7은 고정 길이 서브 시퀀스 커널에서의 서브 구조를 도시한 것이다. 도 7은 도 8을 포함하여 구성된다.

[0061] 커널은 최단 경로 상에서 인접한 의존 서브구조들을 내부적으로 고려한다. 그러나 비인접 서브구조들이 유전적 상호작용을 도출하는 데 있어서 중요한 경우가 있다.

[0062] 예를 들어, 도 7에 도시된 "simulate ~ comp_from (DN) promoter"와 같이 두 개체 사이의 비인접 관계는 유전적 상호작용에 효과를 나타낼 수 있다. 이러한 서브구조들을 비인접 경로(non-contiguous path)라고 한다.

[0063] 또한, 도 9에 도시된 바와 같이 그래프 에지에 방향 정보와 서술어를 부가하여 관계 규명에 도움을 줄 수 있게 하였다.

[0064] 도 9는 주어진 문장에 대하여 개체 간의 의존 그래프를 도시한 것이다.

[0065] 도 9를 참조하면, 굵게 표시된 화살표는 방향성이 있는 최단 의존 경로 그래프를 도시한 것이다.

[0066] 파싱된 그래프에서 두 개체 사이의 최단 경로를 갖는 상호작용 인스턴스를 표현하기 위해서, 의존 그래프를 방향성이 없는 그래프로 변형하되, 에지들이 어느 방향으로도 진행될 수 있도록 한다. 이는 모든 구문적 관계가 구문 헤드로 향하기 때문이다. 그러나 그래프 상에서 원래의 관계 방향을 유지하기 위하여 헤드로 향하는 에지이면 "UP"으로, 헤드에서 종속적인 노드로 향하면 "DN"으로 표현한다.

[0067] 한편, 경로 상의 어느 한 단어에 "PRED"라는 라벨이 있는 경우 그 단어에 연결된 왼쪽 또는 오른쪽 에지의 방향이 변화되는 것을 나타낸다. 이는 의미적인 관계를 결정하는데 중요한 것은 서술어가 필요로 하는 역할을 채우는 의미적인 요소를 찾는 것이므로, 서술서 표지가 주어진 문장에서의 서술어 인자 구조에 대한 정보를 제공하는 특성이 될 수 있을 것이다.

[0068] 도 10은 "ywhE"와 "sigF"를 연결하는 최단 의존 경로를 나타낸다.

[0069] 도 10을 참조하면, 방향성이 있는 의존 관계에 의해 연결되고, 단어의 시퀀스 또는 품사의 시퀀스로 정의되는 최단 경로 스트링이 도시되어 있다.

[0070] 도 10은 의존 경로는 도 11과 같이 의존 경로 스트링의 형태로 표현될 수도 있고, 도 12와 같이 의존 리스트 형태로도 표현될 수 있다.

[0071] 도 12는 단어 의존 리스트와 품사 의존 리스트를 도시한 것이다.

[0072] 단어 의존 리스트는 단어와 단어 간의 의존 관계를 표현하고 품사 의존 리스트는 품사들간의 의존 관계를 표현한다.

[0073] 도 13은 도 9에 도시된 문장의 최단 경로 스트링 인스턴스를 도시한 것이다.

[0074] 모든 개체 쌍 인스턴스들의 라벨들은 "TA"(target-agent), "AT"(agent-target), 및 "O"(no interaction)으로 표현되며, TA는 두 개체가 target-agent의 상호작용을 가지고 있음을 의미하고, AT는 agent-target의 상호작용을, O은 상호작용이 존재하지 않음을 의미한다. 이는 우리가 학습해야 할 출력값(output)에 해당된다. 도 13은 도 9에 도시된 문장으로부터 추출된 몇 개의 인스턴스들의 예이다.

[0075] 3) 다음으로는 위에서 설명한 하부 구조들을 이용해 구조간 유사성 파악을 위한 커널 설계가 어떻게 이루어지는지를 살펴보면 다음과 같다.

[0076] 커널은 앞서 설명하였듯이 일반적으로 인스턴스들의 쌍을 유사성 점수로 맵핑하는 유사성 함수를 의미한다. 즉, 어느 인스턴스들의 공간 X에 대하여 커널 K는 함수 $K: X \times X \rightarrow [0,\infty]$로 표현될 수 있다. 따라서, 커널 방법의 인스턴스들은 유사성 행렬로 표현된다. 또한, 일반적으로 커널은 인스턴스 사이의 내적을 사용하여 쉽게 계산될 수 있다.

[0077] 각 커널은 넓게는 다음의 수학식 1에 따라 커널 공간에 대응하여 정규화될 수 있다.

수학식 1

[0078]
$$K(s,t) = \frac{K(s,t)}{\sqrt{K(s,s)K(t,t)}}$$

[0079] 본 발명에서는 생물학 개체간 관계를 추출하는데 있어서, 최단 의존 경로 상에서 1) 확장된 의존 커널, 2) 스펙트럼 커널, 3) 거리-기반 가중치 서브 시퀀스 커널 4) 워크-기반 가중치 서브 시퀀스 커널로 커널 내부에 고려하는 서브 구조의 타입들을 점진적으로 늘려가면서 가장 효과적인 커널이 어떤 것인지를 측정하였다.

[0080] 결과적으로 워크-가중치 부여 부분열 커널(walk-weighted subsequence kernel)에서 여러 가지 PPI 데이터 셋에 대하여 좋은 결과를 얻을 수 있었다.

[0081] 이러한 워크-가중치 부여 부분열 커널은 작은 양의 트레이닝 데이터로부터 효과적으로 개체 추출에 필요한 구조적인 면들을 학습하였다. 또한, 문장에서의 위치, 의미적인 역할, 및 어휘 특성들과 같은 파라미터의 중요도를 가중치를 변화시켜 구별할 수 있다.

[0082] [확장된 의존 커널]

[0083] 먼저 확장된 의존 커널 정의를 설명하면 의존 커널(dependency kernel)은 두 그래프 간의 구조적 유사성을 공통되는 서브 그래프들을 카운팅함으로써, 직접 산출한다. 두 개의 노드가 같고, 노드에 연결된 자식 노드들이 같으며, 부모 노드에서 자식 노드로의 의존 타입이 같으면, 공통되는 서브 그래프로 판단할 수 있다.

[0084] 한편, e-walk를 이용하되, 완전한 경로 매칭과 더불어 부분 경로 매칭도 작은 가중치를 주어서 함께 고려한다. 이를 확장된 의존 커널(extended dependency kernel)이라고 한다.

[0085] 부분 매칭은 e-walk는 물론 v-walk를 포함하여, 하나의 단어/품사 매칭, 노드-에지 매칭 또는 에지-노드 매칭으로 이루어질 수 있다. 부분 매칭 시에 매칭 유형에 따라 다르게 가중치가 부여된다.

[0086] 노드 x에 대하여, word(x)는 노드 x에 있는 단어, POS(x)는 노드 x의 품사를 의미한다.

[0087] $children_w(n)$은 단어 n과 의존 관계를 가지는 단어 리스트를 나타내고, $children_p(p)$은 품사 p와 의존 관계를 가지는 품사 의존 리스트를 나타낸다. $children_w(n)$은 단어 n과 의존관계를 지니는(관계, 단어)의 집합이고, 이것은 단어 n에 직접 연결되어 있는 자녀 노드들을 의미한다. $children_p(n)$은 n과 의존 관계를 지니는 단어들의 (관계, 품사)의 집합이고, 이것은 n의 품사 p에 직접 연결되어 있는 자녀 품사 노드들을 의미한다.

[0088] $sc_w(n_1,n_2)$는 단어 n_1과 단어 n_2 사이에 있는 두 개의 서브 그래프 간의 공통되는 의존 리스트를 나타낸다. $sc_p(p_1,p_2)$는 품사 p_1과 품사 p_2 사이에 있는 두 개의 서브 그래프 간의 공통되는 의존 리스트를 나타낸다.

[0089] 따라서, $sc_w(n_1,n_2)$와 $sc_p(p_1,p_2)$는 다음의 수학식 2와 같이 나타낼 수 있다.

수학식 2

[0090]
$$sc_w(n_1,n_2) = \{(x,y) \mid (relation,x) \in children_w(n_1), (relation,y) \in children_w(n_2), word(x) = word(y)\}$$
$$sc_p(p_1,p_2) = \{(x,y) \mid (relation,x) \in children_p(p_1), (relation,y) \in children_p(p_2), POS(x) = POS(y)\}$$

[0091] 즉, 두 개의 노드 x,y의 직접적인 자식 노드가 같은 단어이고, x,y의 부모 노드인 n_1,n_2와 동일한 의존 관계이면, (x,y)는 $sc_w(n_1,n_2)$인 집합의 구성요소이다.

[0092] $subcat_w(x)$는 단어 x의 왼쪽 에지와 오른쪽 에지로 구성된 단어 x의 하위 범주화(subcategorization) 정보를 나타낸다. 이는 e-walk와 같은 정보이다.

[0093] 매칭 함수 $C_w(n_1,n_2)$는 n_1, n_2에 기반을 둔 서브 그래프들이 얼마나 공통되는 의존리스트를 포함하는 지와 관련된 가중치가 부여된 카운트 숫자이다. 유사성(또는 유사도)은 루트 노드로부터 시작하여 집합 $sc_w(n_1,n_2)$에 있는 공통적인 의존 자식 단어 쌍들에 대하여 반복적으로 재계산된다.

[0094] 따라서, 매칭 함수 $C_w(n_1,n_2)$는 다음의 수학식 3과 같이 나타낼 수 있다.

수학식 3

- $if\ word(n_1) = word(n_2)\ and\ sc_w(n_1, n_2) \in \{\}\ and\ subcat_w(n_1) = subcat_w(n_2)$
  $then\ C_w(n_1, n_2) = 3.0$
- $else\ if\ word(n_1) = word(n_2)\ and\ children_w(n_1) \in \{\}\ or\ children_w(n_2) \in \{\}$
  $then\ C_w(n_1, n_2) = 1.0$
  $else\ if\ word(n_1) \neq word(n_2)\ or\ children_w(n_1) \in \{\}\ or\ children_w(n_2) \in \{\}$
  $then\ C_w(n_1, n_2) = 0.0$
- $else\ if\ word(n_1) = word(n_2)\ and\ sc_w(n_1, n_2) \neq \{\}\ and\ subcat_w(n_1) = subcat_w(n_2)$
  $then\ C_w(n_1, n_2) = \prod_{(x,y) \in sc_w(n_1,n_2)} (1.5 * (C_w(x, y) + 2)) - 1$
- $else\ if\ word(n_1) = word(n_2) and\ sc_w(n_1, n_2) \neq \{\}\ and\ subcat_w(n_1) \neq subcat_w(n_2)$
  $then\ C_w(n_1, n_2) = \prod_{(x,y) \in sc_w(n_1,n_2)} (C_w(x, y) + 2) - 1$

[0095]

[0096] 구조적 중요성을 고려하여 공통되는 서브 그래프들을 카운팅하기 위해서, 매칭 함수가 설계된다.

[0097] 수학식 3을 참조하면, 공통되는 의존 자식 단어 쌍들의 집합이 비어 있고, 두 노드들이 같은 서브 카테고리 값을 갖고 있으면, 매칭 함수 $C_w(n_1,n_2)$는 3.0을 반환한다.

[0098] n_1 또는 n_2의 자식 노드들이 없지만, 두 노드들이 같은 단어를 갖는 경우, 매칭 함수 $C_w(n_1,n_2)$는 1.0을 반환한다. n_1 또는 n_2의 자식 노드들이 없고, 두 노드들이 다른 단어를 갖는 경우에 매칭 함수 $C_w(n_1,n_2)$는 0을 반환한다.

[0099] 집합 $sc_w(n_1,n_2)$에 있는 공통적인 의존 단어 쌍들에 대하여 재귀적으로 매칭 함수 $C_w(n_1,n_2)$를 호출할 수 있다. 이때, 두 노드가 같은 서브 카테고리 정보를 공유하는 경우 매칭 함수 $C_w(n_1,n_2)$는 더 큰 값의 가중치가 부여될 것이다.

[0100] 이상에서는 두 그래프간의 유사성을 규명하는 데 있어서, 단어 의존 리스트가 공통적인 경우에 대하여 살펴보았고, 품사 의존 리스트에 대하여도 마찬가지로 적용할 수 있다.

[0101] 매칭 함수 $C_p(p_1,p_2)$가 품사 p_1과 p_2를 루트로 하는 품사 의존 서브 그래프에 적용된다. 그러나 구문 의존 경로의 경우에는 서브 카테고리 정보가 수학식 4와 같이 배제된다.

수학식 4

- $else\ if\ pos(p_1) = pos(p_2)\ and\ sc_p(p_1, p_2) \in \{\}\ and\ subcat_p(p_1) = subcat_p(p_2)$
  $then\ C_p(p_1, p_2) = 1.0$
- $else\ if\ pos(p_1) = pos(p_2)\ and\ sc_p(p_1, p_2) \neq \{\}\ and\ subcat_p(p_1) = subcat_p(p_2)$
  $then\ C_p(p_1, p_2) = \prod_{(x,y) \in sc_p(p_1,p_2)} (1.0 * (C_p(x, y) + 2)) - 1$

[0102]

[0103] 매칭 함수 $C_w(n_1,n_2)$는 어휘 서브 그래프와 관련이 있고, 매칭 함수 $C_p(p_1,p_2)$는 형태적 구문적인(morpho-syntactic) 서브 그래프와 관련이 있는 다른 성질을 갖고 있으므로, $C_w(n_1,n_2)$는 $C_p(p_1,p_2)$보다 가중치가 더 많이 부여된다.

[0104] 따라서, 결과적으로 확장된 의존 커널은 구문적인 의존과 어휘적 의존의 결합에 의해서 두 그래프의 유사성을 다음의 수학식 5와 같이 평가하는 것이다.

수학식 5

$$K_D(d_1,d_2) = \sum_{n_1 \in N_1, n_2 \in N_2} C_w(n_1,n_2) + \sum_{p_1 \in P_1, p_2 \in P_2} C_p(p_1,p_2)$$

[0105]

[0106] 수학식 5를 참조하면, 두 그래프 d_1과 d_2 사이에서 매칭되는 모든 노드들의 매칭 함수의 값을 평가한 것이다. 즉, 두 그래프 d_1과 d_2 사이의 공통되는 단어 의존 서브 그래프와 품사 의존 서브 그래프의 합을 계산한 것이다.

[0107] 이상과 같이 부분 경로 매칭과 서브 카테고리화 정보를 사용하는 확장된 의존 커널을 사용하여 기존의 LLL 데이터 셋에 대해 실험한 결과 F-스코어는 60.4에서 69.4로 개선되었다.

[0108] [스펙트럼 커널]

[0109] 이하에서는 비인접 서브 그래프를 다루는 스트링 커널을 살펴보기로 한다.

[0110] 스트링 커널은 텍스트 문서들을 내부에 포함된 부분 문자열을 이용하여 두 문서가 얼마나 유사한지 판단하는 데 주로 사용되어 왔다. 즉, 두 문서가 포함하는 공통되는 부분 문자열이 많으면 많을수록, 문서끼리 더욱더 유사하다고 판단할 수 있다.

[0111] 일반적으로 스트링(문자열)은 유한한 알파벳에서 선택된 유한한 기호 열(symbol sequence)을 의미한다. 스트링 커널은 두 스트링이 하부 구조로 서브 시퀀스(subsequence)나 서브 스트링(substring)을 얼마나 공통적으로 가지고 있는지를 가지고 유사도를 판별하는데 부분 문자열을 의미하는 서브 시퀀스와 서브 스트링에는 정의 상의 차이가 있다.

[0112] $s_1\ s_2 \ldots s_n$로 표현되는 스트링에 대하여 스트링 내에서 서로 인접한 서브 스트링은 $s_i s_{i+1} \ldots s_{j-1} s_j$로 인접 표현이다.

[0113] 반면에 서브 시퀀스는 임의의 스트링을 의미하는 것으로, $s_i s_j \ldots s_k$로 표현될 수 있고, 부분 문자열이 인접할 수도 있고, 인접하지 않을 수도 있다.

[0114] 스트링 커널 중 가장 기본적인 커널은 스펙트럼 커널로서 p-길이 인접 서브 스트링을 공통적으로 얼마나 갖고 있는지를 계산함으로써, 두 문자열 스트링을 비교하는 방법이다. 이를 차수 p 또는 p 스펙트럼 커널이라고 한다.

[0115] $K_s(s_1,s_2)$는 두 개의 최단 경로 스트링 s_1, s_2 사이의 공통적인 p 길이의 서브 스트링의 수를 나타내는 것이다. 이를 수학적으로 기술하면 다음과 같다.

수학식 6

$$K_s(s_1,s_2) = \sum_{i=1}^{|s_1|-p+1} \sum_{j=1}^{|s_2|-p+1} K_p(s_1(i:i+p), s_2(j:j+p))$$

$$K_p(s,t) = \begin{cases} 1 \ if \ \ s = t \\ 0 \ otherwise \end{cases}$$

[0116]

[0117] 여기서, 스트링 s(i:i+p)는 스트링 s의 p 길이의 서브 스트링 $s_i \ldots s_{i+p}$를 의미한다. 스펙트럼 커널의 차수 p를 3이라고 하고, 어휘 의존 경로 스트링 K_s와 구문 의존 경로 스트링 K_s를 합하여, 공통되는 서브 스트링을 카운팅한다.

[0118] 스펙트럼 커널은 도 6에 도시된 바와 같은 서브 구조들을 이용한다.

[0119] 스펙트럼 커널은 LLL 데이터에 대하여 70.5의 F-스코어를 가지므로, 이전의 확장된 의존 커널의 성능보다 우수하다.

[0120] [거리-기반 가중치 서브 시퀀스 커널]

[0121] 그런데, 스펙트럼 커널에서 "stimulate_obj(DN)~comp_from(DN)"과 같은 서브 구조들은 서브 구조들 사이에 거리(gap)가 존재하기 때문에 하부 구조 비교에서 제외된다.

[0122] 따라서, 스펙트럼 커널과는 달리 문자 사이의 거리(갭)를 허용하는 서브 시퀀스 커널을 살펴보기로 한다. 즉, 몇 개의 문자들이 두 개의 매칭되는 서브 시퀀스 사이에 끼어들어 위치할 수 있는 경우이다.

[0123] 이러한 경우는 도 7에 도시된 고정 길이 서브 시퀀스 커널의 서브 구조들로 나타내어질 수 있다.

[0124] 서브 구조 "stimulate-obj(DN)~comp_from(DN)"는 "stimulate-obj(DN)-any other noun-comp_from(DN)"과 같은 구와 매칭될 수 있다. 이 경우 "transcription" 대신에 다른 명사들이 포함되어도 무관하다.

[0125] 이러한 커널의 장점은 스트링에 존재하는 긴 범위의 의존을 찾아낼 수 있다는 점이다.

[0126] 스펙트럼 커널에서와 마찬가지로, 고정 길이의 서브 시퀀스만을 고려하여 특성 스페이스의 차원을 감소시킨다. 먼저 p 길이의 모든 서브 시퀀스들의 세트를 A^p로 정의하고, 스트링 $s=s_1s_2...s_{|p|}$의 길이를 $|p|$로 나타낸다. 굵게 표시된 **i**는 인덱스 시퀀스 $i_1...i_{|u|}$를 나타내고, 스트링 s의 서브 시퀀스 u는 u=s[**i**]로 나타낸다. 즉, u는 **i**로 인덱싱되는 위치에 있는 s의 서브 시퀀스이고, $s_{i1}s_{i2}...s_{i|u|}$로 표현될 수 있다.

[0127] 특성 정합 함수(feature coordination function) $\phi_u(s)$는 서브 스트링 u가 입력 스트링 s에서 인접 및 비인접 서브 시퀀스로서 몇 번 발생하는지를 카운트하는데 사용된다.

[0128] I_p가 길이 p의 인덱스 시퀀스 세트이라고 할 때, $\phi_u^p(s)$는 길이 p의 서브 스트링 u가 몇 번 발생하는지 카운트하는데 사용된다.

[0129] 같은 길이 p를 갖으며, 비교 대상이 되는 스트링 s와 t를 가정하고, 특성 스페이스는 최단 경로 스트링으로부터 선택된 길이 p의 모든 서브 시퀀스들로 이루어진다고 하면, 스트링 s와 t 사이의 전체 내적은 다음의 수학식 7과 같이 표현될 수 있다.

수학식 7

[0130]
$$\varphi_u^{\mathrm{P}}(s) = |\{\mathbf{i} : u = s(\mathbf{i})\}|, u \in A^{\mathrm{P}}$$

$$K_{\mathrm{p}}(s,t) = <\varphi_u(s), \varphi_u(t)> = \sum_{u \in \mathrm{A}^{\mathrm{P}}} \varphi_u^{\mathrm{P}}(s)\,\varphi_u^{\mathrm{P}}(t)$$

$$= \sum_{u \in \mathrm{A}^{\mathrm{P}}} \sum_{(\mathbf{i},\mathbf{j}):u=s(\mathbf{i})=t(\mathbf{j})} 1$$

$$= \sum_{(i,j) \in I_{\mathrm{p}} \times I_{\mathrm{P}}:s(\mathbf{i})=t(\mathbf{j})} 1$$

[0131] 길이 p는 3으로 선택하고, LLL 데이터 세트에 대하여 F-스코어를 구해 보면, 스펙트럼 커널보다는 우수하지만, 워크 커널과 같은 77.5의 값이 나왔다.

[0132] 서브 시퀀스 커널에서 모든 서브 구조들은 같은 가중치를 가지고 카운트된다. 즉, 서브 시퀀스가 연속적이든, 비연속적이든 또는 서브 시퀀스가 얼마나 퍼져 있는지는 가중치에 영향을 미치지 않는다.

[0133] 한편, 서브 시퀀스 커널의 서브 시퀀스의 연속성의 정도를 반영하기 위해 거리 가중치 (gap-weighted) 고정 길

이 서브 시퀀스 커널을 이하 설명하기로 한다.

[0134] 거리-가중치 고정 길이 서브 시퀀스 커널에서는 "stimulate-obj(DN) -transcription"과 "stimulate~comp_from(DN)-promoter"인 두 개의 서브 구조들은 다른 가중치를 갖는다. 즉 인접, 비인접 서브 시퀀스를 구분하여 스트링 내부 문자들 사이에 들어가는 갭의 크기를 반영하기 위해 감쇠 팩터 $\lambda(0<\lambda\leq 1)$을 사용한다. 인접하지 않은 서브 시퀀스 매칭은 반영 정도를 낮춘다. 즉, 서브 시퀀스의 시작과 끝이 멀리 떨어져 있을수록 두 단어 사이에 다른 단어들이 많이 포함되어 있을수록 작은 가중치를 갖도록 카운트 값을 낮춘다. 특성 정합 함수(feature coordination function) $\phi_u(s)$는 다음의 수학식 8과 같이 서브 시퀀스에 가중치 부여된 카운팅으로 변환된다.

수학식 8

[0135]
$$\varphi_u(s) = \sum_{\mathbf{i}:u=s[\mathbf{i}]} \lambda^{l(\mathbf{i})}$$
$$l(\mathbf{i}) = i_{|u|} - i_1 - 1 \quad (\mathbf{i} = (i_1 \cdots i_{|u|}))$$

[0136] 여기서, 갭의 전체 길이에 의해 가중치를 낮게 하여 카운트된다.

[0137] $l(\mathbf{i})$는 인덱스 $\mathbf{i}$의 길이를 나타낸다.

[0138] 두 개의 서브 시퀀스들 사이의 유사성 값은 서브 시퀀스들이 얼마나 떨어져 있는지를 나타내면서, $l(\mathbf{i})$와 $l(\mathbf{j})$의 팩터만큼 줄어들게 된다.

[0139] A^p에 대하여 두 개의 스트링 s와 t 사이의 내적은 발생 빈도와 길이에 따라 가중치가 부여된 공통 고정 길이 서브 시퀀스들의 모든 합이며, 다음의 수학식 9와 같이 나타낼 수 있다.

수학식 9

[0140]
$$K_p(s,t) = \sum_{u\in A^p} <\varphi_u(s)\,\varphi_u(t)> = \sum_{u\in A^p} \sum_{\mathbf{i}:u=s(\mathbf{i})} \lambda^{l(\mathbf{i})} \sum_{\mathbf{j}:u=t(\mathbf{j})} \lambda^{l(\mathbf{j})}$$
$$= \sum_{u\in A^p} \sum_{\mathbf{i}:u=s(\mathbf{i})} \sum_{\mathbf{j}:u=t(\mathbf{j})} \lambda^{l(\mathbf{i})+l(\mathbf{j})}$$

[0141] 즉, 본 커널 함수는 두 스트링 사이의 p 심볼들로 이루어진 매칭된 서브 시퀀스들을 모두 계산하고, 각각의 서브 시퀀스들은 그 길이에 따라 가중치가 부여된다.

[0142] 감쇠 팩터 λ를 0.5로 하고, 인덱스 세트는 $U=A^3$(최단 경로 스트링에서의 3개의 노드 또는 에지 열)로 한정한다.

[0143] 만일 λ를 1로 하는 경우, 모든 서브 시퀀스들의 가중치는 l과 무관하게 1이 될 것이다. 이런 경우는 커널이 모든 공통 서브 시퀀스들을 1로 동일하게 카운팅하는 고정 길이 서브 시퀀스 커널과 같다.

[0144] 그러나, 각 서브 시퀀스의 의존 거리에 따른 광범위한 가중치를 부여한다고 하여도, F-스코어는 70.2로, 서브 시퀀스 커널보다 작은 값을 갖는다. 따라서, 서브 스트링에 갭에 따라 가중치를 부여하는 것은 비효율적임을 알 수 있다.

[0145] [워크-기반 가중치 서브 시퀀스 커널]

[0146] 이상에서 살펴본, 거리-가중치 고정 길이 서브 시퀀스 커널을 개선하기 위해 워크 가중치 고정 길이 서브 시퀀

스 커널을 이하 개시하기로 한다.

[0147] 거리-가중치 고정 길이 서브 시퀀스 커널과 마찬가지로 워크-가중치 고정 길이 서브 시퀀스 커널은 각 서브 시퀀스에 대하여 다른 가중치를 부여하되, 갭의 길이에 따라 서브 시퀀스의 가중치를 부여하는 것이 아니라 서브 시퀀스의 타입에 따라 가중치를 부여한다.

[0148] 인접한 서브 시퀀스는 밀착한 상태에서 최단 경로 스트링에 더 많은 영향을 가지므로, 인접하지 않은 서브 시퀀스보다는 인접한 서브 시퀀스에 더 많은 가중치를 부여한다.

[0149] 또한, 정보의 의미적 역할을 강조하기 위해 v-walk보다는 e-walk에 더 많은 가중치를 부여한다.

[0150] 워크 가중치 고정 길이 서브 시퀀스 커널에 길이 3의 서브 시퀀스를 고려하면 다음의 수학식 10과 같이 표현될 수 있다.

수학식 10

$$\varphi_u^p(s) = \sum_{\mathbf{i}:u=s(\mathbf{i})} \lambda_u, u \in \mathrm{A}^p$$

$$K_p(s,t) = \sum_{u \in \mathrm{A}^p} \varphi_u^p(s)\, \varphi_u^p(t)$$

$$if\ l(\mathbf{i}) = 3\ and\ (\mathbf{i}_1 \%2) == 0 \quad then\ \lambda_u = 3.0 \quad (e-walk)$$

$$else\ if\ l(\mathbf{i}) = 3\ and (\mathbf{i}_1 \%2) != 0\ then\ \lambda_u = 2.0 \quad (v-walk)$$

$$else\ \lambda_u = \sum_{\mathbf{i}:u=s[\mathbf{i}]} \lambda^{l(\mathbf{i})},\ l(\mathbf{i}) = i_{|u|} - i_1 - 1 \quad (\mathbf{i} = (i_1 \cdots i_{|u|}))$$

[0151]

[0152] 수학식 10을 참조하면, 공통적인 인접 e-walk 서브 스트링에 대하여는 3.0의 가중치를 부여하고, 공통적인 인접 v-walk 서브 스트링에 대하여는 2.0을 부여한다.

[0153] 인접하지 않은 서브 시퀀스들의 경우 문자열 갭에 따른 가중치를 부여한다. 이때 갭의 크기와는 상관없이 1.0을 비인접 서브 시퀀스에 부여한다.

[0154] 위와 같이 가중치를 부여한 결과 워크-가중치 고정 길이 서브 시퀀스 커널은 LLL 데이터에 대하여 유전 관계를 추출하는 데 있어서, 82.1의 F-스코어를 나타낸다.

[0155] 따라서, 공통적인 서브 시퀀스들의 타입에 따라 가중치가 부여된 스트링 커널을 사용하는 것이 매우 효율적임을 알 수 있다.

[0156] 도 14는 본 발명의 일 실시 예에 따른 텍스트로부터 개체 간의 관계를 도출하는 방법의 흐름도이다.

[0157] 1410 단계에서 텍스트 내의 문장들을 파싱하여 각 문장별로 단어들 간의 문법적 의존 그래프를 생성한다. 도 9를 참조하면, 하나의 예시 문장 및 개체 간의 의존 그래프가 도시되어 있다.

[0158] 1420 단계에서 생성된 의존 그래프에서 두 개 이상의 개체를 연결하는 최단 의존 경로를 도출한다. 도 10을 참조하면, "ywhE"와 "sigF"를 연결하는 최단 의존 경로가 도시되어 있다. 최단 의존 경로는 방향성이 있는 것이 바람직하다.

[0159] 1430 단계에서 최단 의존 경로로부터 두 개 이상의 개체 간의 관계를 학습하는데 필요한 인스턴스를 생성한다. 인스턴스란, 두 개 이상의 개체를 잇는 문장 내의 단어들과 상기 단어들 간의 문법적 의존 관계들을 나타낸 스트링이다.

[0160] 1440 단계에서 인스턴스들로부터 서브 구조들을 생성하고, 상기 생성된 서브 구조들을 이용하여 상기 두 개 이상의 개체 간의 유사도를 측정한다.

[0161] 서브 구조는 e-walk, v-walk, 부분 매칭, 비인접 경로를 포함한다. 또한, 인스턴스들의 쌍을 유사도로 맵핑하는 유사도 함수인 커널을 이용하여 유사도를 측정한다.

[0162] 1450 단계에서 측정된 유사도로부터 상기 두 개 이상의 개체 간의 관계를 도출한다.

[0163] 정리하면, 본 발명의 실시 예에서는 공통적인 서브 그래프가 인접한 서브 시퀀스인 경우 인접하지 않은 서브 시퀀스인 경우보다 더 많은 가중치를 부여한다. 또한, 인접하지 않은 서브 시퀀스의 경우 거리(갭)의 크기와는 상관없이 같은 가중치를 부여하도록 한다.

[0164] 또한, 공통적인 서브 그래프의 서브 시퀀스는 인접한 e-walk 서브 시퀀스, 인접한 v-walk 서브 시퀀스, 및 인접하지 않은 서브 시퀀스로 이루어지고, 각각의 서브 시퀀스에 다른 가중치를 부여할 수 있다. 그리고, 정보의 의미적 역할을 강조하기 위해 v-walk보다 e-walk에 더 많은 가중치를 부여한다.

[0165] e-walk는 어휘적 워크 또는 구문적 워크로 분류되며, 어휘적 워크는 어휘 단어 및 어휘 단어 간의 의존 관계로 구성되고, 구문적 워크는 품사 및 품사 간의 의존 관계로 구성된다.

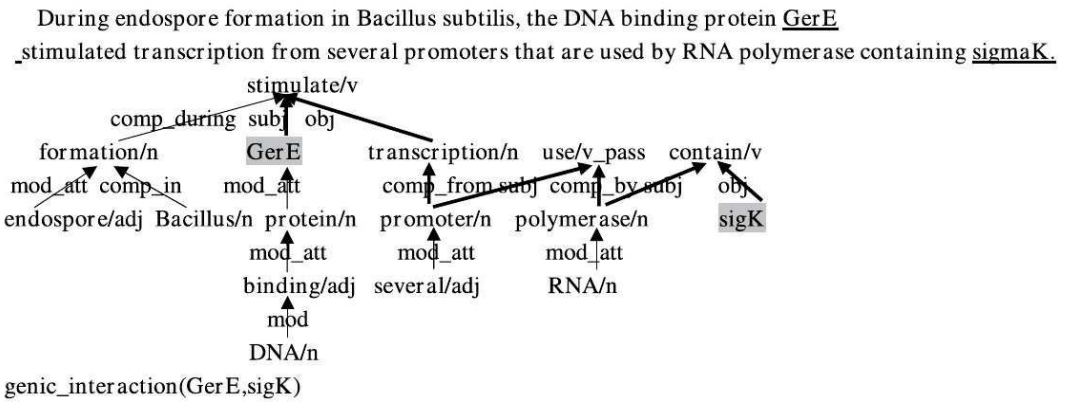
[0166] 본 발명의 실시 예들은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0167] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시 예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.

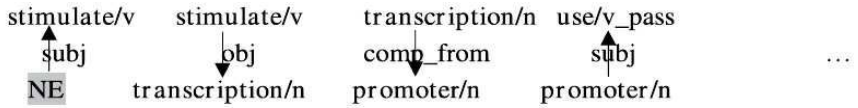
[0168] 따라서, 본 발명의 사상은 설명된 실시 예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고 할 것이다.

도면

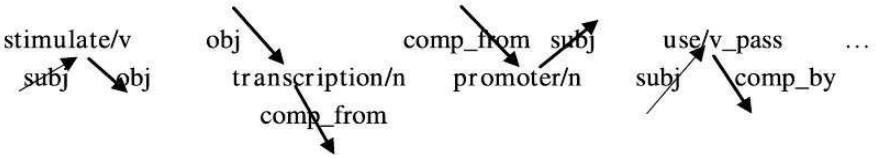
도면1



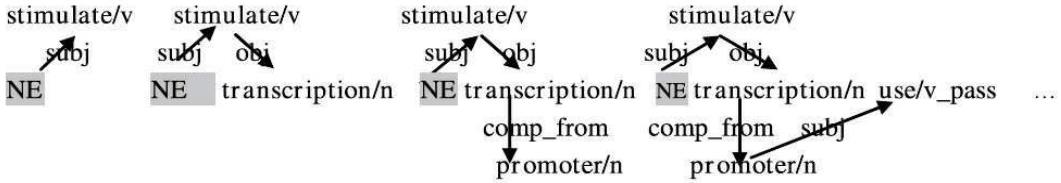
도면2



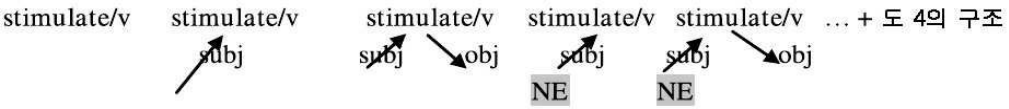
도면3



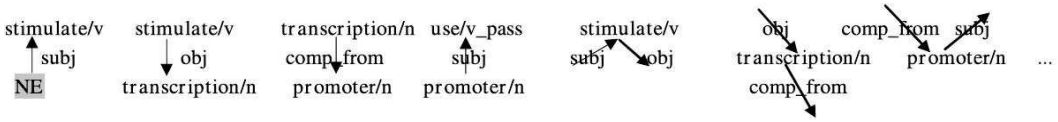
도면4



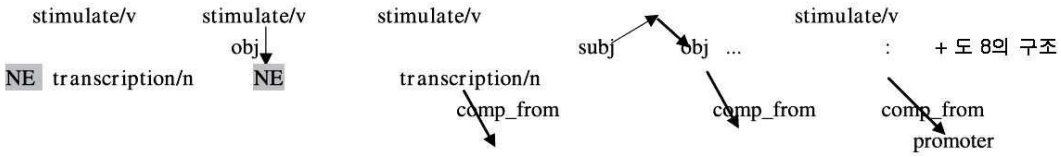
도면5



도면6



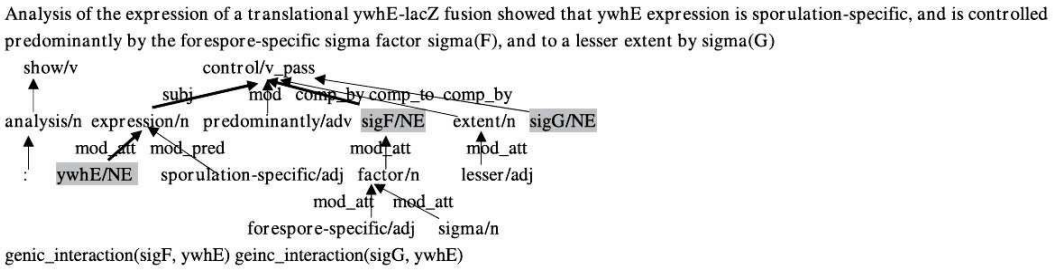
도면7



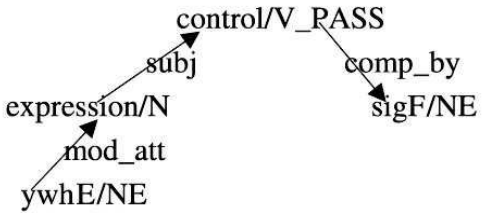
도면8

str=NE1_subj(UP)_stimulate
str=NE1_subj(UP) ~ obj(DN)
str=NE1_subj(UP) ~ transcription
str=NE1_subj(UP) ~ comp_from(DN)
str=NE1_subj(UP) ~ promoter
str=NE1_subj(UP) ~ subj(UP)
str=NE1_subj(UP) ~ use
str=NE1_subj(UP) ~ comp_by(DN)
str=NE1_subj(UP) ~ polymerase
str=NE1_subj(UP) ~ contain

도면9



도면10



도면11

NE1	mod_att(UP)	expression	sub(UP)	control	comp_by(DN)	NE2
NE1	mod_att(UP)	N	sub(UP)	V_PASS/PRED	comp_by(DN)	NE2

도면12

Node	Node Relation set
NE1	{ }
expression	{(mod_att, NE1)}
control	{(subj,expression),(comp_by,NE2)}
NE2	{ }

POS	POS Relation set
NE1	{ }
N	{(mod_att, NE1)}
V_PASS	{(subj,N),(comp_by,NE2)}
NE2	{ }

도면13

TA	NE/NE	mod_att(UP)	expression/N	subj(UP)	control/V_PASS_PRED	comp_by(DN)	NE/NE	(ywhE-SigF)
TA	NE/NE	mod_att(UP)	expression/N	subj(UP)	control/V_PASS_PRED	comp_by(DN)	NE/NE	(ywhE-SigG)
0	NE/NE	comp_by(UP)	control/V_PASS_PRED	comp_by(DN)	NE/NE			(SigF-SigG)

도면14

