말뭉치 정규화와 의미 규칙 기반 요구사항 정제를 통한 원인-결과 그래프 자동 생성

장우성*, 김영철*
*홍익대학교 소프트웨어공학연구실
e-mail:{jang, bob}@selab.hongik.ac.kr

Automatic Generation of Cause-Effect Graph through Refining Requirements Specifications based on Semantic rules with Corpus Normalization

Woo Sung Jang*, R.Young Chul Kim*
*SE Lab., Dept of Software and Communications Engineering, Hongik University

요 약

현실적으로 요구사항의 불명료성은 테스트 케이스 추출에 어려움을 초래한다. 명료한 요구사항 기반의 사용자 승인 테스트는 소프트웨어의 올바른 품질을 증가시키고, 유지보수 비용을 감소시킨다. 하지만 중소기업에서는 촉박한 개발 기간, 테스트 도구 구매 비용의 부담, 낮은 테스트 기술 레벨 등의 이유로 좋은 품질의 테스트를 수행하기가 힘들다. 이러한 문제점의 해결을 위해 말뭉치 정규화를 이용한 의미 규칙으로 불명료한 요구사항을 간결하고 명료한 요구사항으로 변경하기 위한 메커니즘을 제안한다. 또한 이를 원인-결과 그래프 자동 생성하는 방법을 제안한다. 이는 원인-결과 그래프를 통해 테스트케이스를 최대한 생성하는 기초가 될 수 있다.

1. 서론

최근 소프트웨어 시장에서, 소프트웨어 테스트에 대한 중요성은 계속 늘어나고 있다. 대기업은 다양한 테스트 솔루션을 적용하여 소프트웨어를 검증하지만, 국내 중소기업의 소프트웨어 테스트 환경은 개발자의 테스트에 대한 이해 부족, 테스트케이스 작성 시간 부족 등의 이유로 좋은 품질의 테스트 적용이 미비한 실정이다[1].

테스트 자동화는 테스트 케이스 작성에 필요한 시간적 비용을 줄이고, 사람의 실수로 인해 발생할 수 있는 잘못된 테스트케이스 생성을 방지할 수 있다. 하지만 추상적인 요구사항에 대한 테스트 자동화는 사람이 수행하는 수작업에 비해 정확도가 떨어지기 때문에, 테스트 수행 과정에서 사람의 개입으로 인한 테스트 비용 및 시간이 증가된다.

본 논문은 이러한 문제점을 해결을 위한 방법으로써, 말뭉치 정규화를 이용한 의미 규칙으로 개선된 요구사항 명세서로부터 원인-결과 그래프 자동 생성 방법을 제안한다. 원인-결과 그래프는 요구사항의 모호성 및 불일치를 쉽게 명세 및 감지할 수 있고, 블랙박스 기반 테스트 케이스를 쉽게 자동 생성할 수 있다[2]. 요구사항 문장에서 원인, 결과, 요소 결합 정보를 식별하는 규칙을 정의하고, 식별된 요소를 이용하여 원인-결과 그래프를 만든다면, 테스트케이스를 만들기 위한 원인-결과 그래프를 생성할 수 있다.

본 논문의 구성은 다음과 같다. 2장은 관련 연구로써, 기존 한글 문장 정규화 및 의미 분석 방법을 언급한다. 3장은 말뭉치 정규화를 이용한 의미 규칙으로 개선된 요구사항 명세서로부터 원인-결과 그래프 자동 생성 방법을 설명한다. 4장은 적용 사례를 언급한다. 5장은 결론 및 향후 연구를 언급한다.

2. 관련 연구

2.1. 구문 관계 지식 추출을 위한 말뭉치 정규화 [3][4]

구문 관계 지식은 문장 성분 간의 문법적 관계에 대한 지식을 의미한다. 자연어 처리를 위해서는 문장 내 구성 성분 간의 관계 구성을 위해 정확한 조사의 사용이 중요하다. 하지만 실제 문장에서는 다음과 같은 문제점이 발생한다.

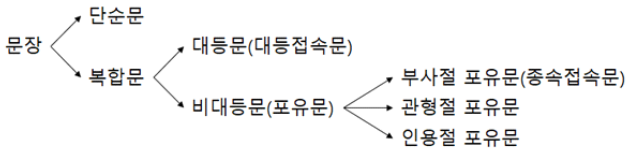
(가) 격이동에 의한 조사 변경 현상
(나) 복합어의 이형태 표현 현상
(다) 선행문과 후행문의 공통 성분 생략 현상

이러한 문제점을 해결하기 위한 올바른 구문 관계 지식 추출을 위한 말뭉치 정규화 과정은 다음과 같다.

(가) 격이동 복원
(나) 복합어의 중복된 형태 통일
(다) 접속문의 생략된 주어 복구

2.2. 의미적 관점에서 대등 접속문과 종속 접속문의 분류[5]

접속문은 의미론적 관점에서 대등 접속문과 종속 접속문으로 나눌 수 있으며, 내포문은 명사절 내포문과 관형사절 내포문, 부사절 내포문으로 각각의 하위 체계를 가진다. 문장의 체계에 대한 전체적인 구조는 아래의 <그림 1>과 같다.



(그림 1) 문장의 체계

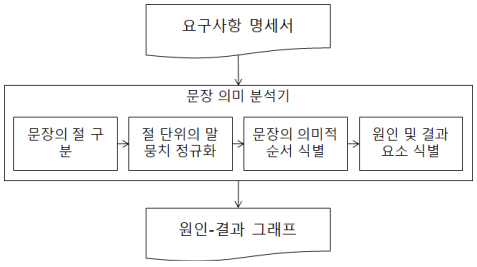
각 관계에서 사용되는 연결어미를 분류 했을 때, 병렬관계에 해당되는 연결 어미는 '-고', '-(으)며', '-(으)면서' 등이다. 이 외의 다른 관계도 다양한 연결 어미를 가진다.

2.3. 원인-결과 그래프

원인-결과 그래프는 자연어 요구사항을 형식적인 표기법으로 전환하는데 효과적이다. 요구사항에서 소프트웨어의 최대 테스트 영역을 포함하는 최소한의 테스트케이스를 설계하는데 용이하다. 원인-결과 그래프를 사용하여 작성된 테스트케이스는 최소한의 테스트케이스를 이용하여 100%의 커버리지를 포함할 수 있다[6].

3. 말뭉치 정규화를 이용한 의미 규칙으로 개선된 요구사항 명세서로부터 원인-결과 그래프 자동 생성 방법

말뭉치 정규화를 이용한 의미 규칙으로 개선된 요구사항 명세서로부터 원인-결과 그래프 자동 생성하는 방법은 아래의 <그림 2>와 같다.



(그림 2) 요구사항으로부터 원인-결과 그래프 생성 과정

요구사항 명세서의 요구사항들을 하나씩 분류하여 문장 의미 분석기에 입력한다. 문장 의미 분석기는 4개의 단계을 거쳐 원인-결과 그래프를 생성한다.

문장의 절 구분 단계에서는 문장을 구성하는 절들을 식별한다.

절 단위의 말뭉치 정규화 단계에서는 절 내의 말뭉치들을 정규화하여 절의 의미를 명확하게 표현한다.

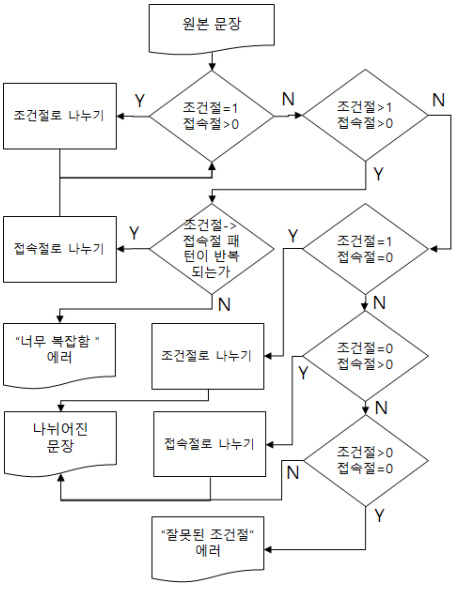
문장의 의미적 순서 식별 단계에서는 절들의 의미적 순서를 식별한다.

원인 및 결과 요소 식별 단계에서는 의미적 순서로 나열된 절들로부터 원인 및 결과 요소를 식별한다.

3.1. 문장의 절 구분

접속문의 생략된 주어 복구 방법[3][4], 접속문 분류 방법[5]을 사용하여 문장에서 접속문, 조건문을 식별한다. 그리고 의미가 잘못 해석되는 문장들은 데이터 사전에 입력하여 예외사항으로 등록 후 따로 식별한다.

문장에서 접속문과 조건문 식별 시, 접속문을 먼저 식별하는지, 또는 조건문을 먼저 식별하는지에 따라 문장의 의미가 달라진다. 본 논문에서는 아래 <그림 3>의 알고리즘을 통해 접속문 및 조건문 우선 식별 순위를 파악한다.



(그림 3) 문장의 절 구분 알고리즘

3.2. 절 단위의 말뭉치 정규화

문장이 절 단위로 나뉘었다면, 격이동 복원, 복합어의 중복된 형태 통일 방법, 접속문의 생략된 주어 복구 방법[3][4]을 적용하여 각 절의 문장들을 정규화 한다. 의미가 잘못 해석되는 문장들은 데이터 사전에 입력하여 예외사항으로 등록 후 처리한다.

3.3. 문장의 의미적 순서 식별

앞의 과정들을 거쳐 가장 작은 단위로 나뉘어지고, 정규화 된 절들의 순서를 식별한다. 이 의미적 순서는 원인-결과 그래프에서 AND/OR/NOT 연산 중 먼저 수행할 연산을 결정한다.

3.4. 원인 및 결과 요소 식별

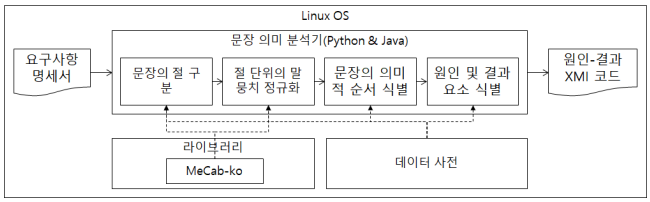
앞서 나뉘어진 절에서 원인, 결과, 결합 정보를 식별한다. 접속절은 결합 정보가 되고, 조건절은 원인 및 결과가 된다. 예외적인 접속절은 데이터 사전에 등록한다.

4. 적용 사례

4.1. 원인-결과 그래프 자동 생성 환경 구축

원인-결과 그래프 자동 생성 환경은 아래의 <그림 4>와 같다. 문장 의미 분석기는 한글 요구사항을 입력 받아, 원인-결과 그래프의 정보가 담긴 XMI 파일을 생성한다.

문장 의미 분석기는 리눅스 환경에서 실행되고, Java 프로그램으로 구현되며, 내부적으로 Pyhon 프로그램을 실행한다. 문장의 절 구분, 그리고 절단위의 말뭉치 정규화 과정에는 문장의 형태소 분석 과정이 필요하다. 문장의 형태소 분석을 위해 MeCab-ko 라이브러리를 사용한다. MeCab-ko는 리눅스 기반의 라이브러리이고, Python에서 동작한다. 데이터 사전에는 요구사항에서 주로 사용되는 문법적 예외사항이 저장된다.



(그림 4) 구축된 환경

4.2. 원인-결과 그래프 자동 생성 수행 결과

<표 4>와 같은 문장을 문장 의미 분석기에 입력한 경우, 아래 <표 5>와 같은 XMI 코드를 생성한다.

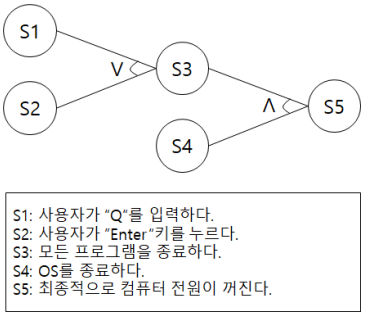
<표 4> 문장 의미 분석기에 입력한 문장

사용자가 "Q"를 입력하거나, "Enter"키를 누르면, 모든 프로그램이 종료되고, OS가 종료된 후, 최종적으로 컴퓨터 전원이 꺼진다.

<표 5> 생성된 원인-결과 그래프 XMI 코드

```
<?xml version = "1.0" encoding = "ISO-8859-1"?>
<cem:CauseEffect xmi:version="2.0" xmlns:xmi = "http://www.omg.org/XMI" xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance" xmlns:cem = "http://cem/1.0">
<element xmi:id="e1" value="사용자가 Q를 입력하다."/>
<element xmi:id="e2" value="사용자가 Enter키를 누르다."/>
<element xmi:id="e3" value="모든 프로그램을 종료하다."/>
<element xmi:id="e4" value="OS를 종료하다."/>
<element xmi:id="e5" value="최종적으로 컴퓨터 전원이 꺼진다."/>
<connector xmi:id="con1" type="OR">
<cause href="//@element.1" />
<cause href="//@element.2" />
<effect href="//@element.3" />
</connector>
<connector xmi:id="con2" type="AND">
<cause href="//@element.3" />
<cause href="//@element.4" />
<effect href="//@element.5" />
</connector>
</cem:CauseEffect>
```

<표 5>의 XMI 코드를 해석하여 가시화 한 결과는 <그림 5>와 같다.



(그림 5) 가시화 된 원인-결과 그래프

5. 결론 및 향후 연구

본 논문은 요구사항 명세서로부터 원인-결과 그래프 자동 생성을 위한 방법을 제안한다. 자연어 요구사항 문장에서 절을 구분하고, 구분된 절에서 조건절, 결과절, 선행절, 후행절을 식별하여 원인-결과 그래프를 생성한다.

원인-결과 그래프를 이용하면, 블랙박스 기반의 테스트케이스를 자동 생성할 수 있기 때문에, 요구사항으로부터 원인-결과 그래프를 자동 생성한다면, 쉽게 테스트케이스를 자동 생성할 수 있는 기반을 제공할 수 있다.

하지만 한국어 문장에서 원인 및 결과를 식별하면, 중복되는 의미를 가지는 원인 및 결과가 존재한다. 향후 연구로써, 중복되는 의미를 가지는 원인-결과 요소들을 통합하고, 자동 생성되는 원인-결과 그래프의 정확도를 측정하기 위한 방법을 연구할 예정이다.

ACKNOWLEDGEMENT

본 논문은 2019년도 산업통상자원부의'창의산업융합 특성화 인재양성사업'(과제번호 N0000717)과 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1D1A3B03035421).

참고문헌

[1] 권오승, 홍사능, "로그 기반 효과적 반복 테스트", 한국경영정보학회 추계학술대회, pp.685-690, 2009.

[2] Amit Paradkar, K.C. Tai, M.A. Vouk, Specification based testing .ng cause effect graphs, Annals of Software Engineering, January 1997, Volume 4, Issue 1, pp 133 - 157.

[3] 조정미, 조영환, 김길창, "구문 관계 지식 추출을 위한 코퍼스 정규화에 대한 연구", 한국정보과학회 언어공학연구회, pp.207-215, 1996.

[4] 조정미, 김길창, "구문 관계 정보 추출을 위한 말뭉치의 정규화에 대한 연구", 한국인지과학회, Vol.7, No.2, pp.39-56, 1996.

[5] 김기성, "현대몽골어와 한국어의 접속문 비교연구", 한국몽골학회, 몽골학 27권 0호, pp.151-186, 2009.

[6] Myers, Glenford J. (1979). The Art of Software Testing. John Wiley & Sons. ISBN 0-471-04328-1.