2002 춘계학술발표회 논문집
한국원자력학회

SIS-RT를 위한 문서 추적성 분석 방법론의 개발
Development of a Document Traceability Analysis Methodology for SIS-RT

김만철, 유영제, 성풍현
한국과학기술원
대전광역시 유성구 구성동 373-1

손대성
㈜ 액트
대전광역시 유성구 덕진동 150번지 한국원자력연구소내 창업보육센터109호

요약

본 연구에서는 고신뢰도 소프트웨어를 위한 확인 및 검증 도구인 SIS-RT에서 문서 추적성 평가를 위한 방법론을 제안하였다. 제안된 방법론은 정보검색에 이용되는 코사인벡터유사도공식에 기반하여 유사도를 계산하고, 역문헌빈도에 기반하여 주요단어(keyword)들을 추출한다. 제안된 방법론을 실제 문서에 적용해 본 결과 주어진 방법론에 의해서 두 문장 사이의 유사도를 계산하였을 때, 두 문장 사이의 유사도가 80% 이상일 경우 추적성이 있는 것으로 판단하여도 좋은 것으로 나타났으며, 유사도가 30%에서 80% 사이일 경우는 검토자의 판단이 필요한 것으로 나타났다. 비록 제안된 방법론에 따른 유사도와 추적성이 일치하지 않는 예외적인 경우가 있기도 하지만, 제안된 방법론은 추적성 평가에 매우 유용한 것으로 나타났다.

Abstract

In this study, we propose a document traceability analysis methodology for high-reliability software inspection tool, SIS-RT. The proposed methodology calculates the similarity between two sentences using the cosine vector similarity formula. The keywords are derived based on inverse document frequencies. When we apply the proposed methodology to real documents, it is found that a source sentence and a destination sentence can be determined to have traceability when the similarity between the two sentences is over 80%. In case the similarity between two sentences is over 30% but less 80%, it turns out that the decision of the inspector is required. Even though sometimes the trends between the similarity of two sentences and the traceability

of two sentences do not exactly matches, it is found that the methodology we propose in this paper is useful for traceability analysis.

I. 서론

원자력발전소의 보호시스템에 사용되는 디지털 시스템은 매우 높은 신뢰도를 요구한다. 고신뢰도의 소프트웨어를 얻기 위해서는 소프트웨어 전 단계에서 철저한 확인 및 검증 작업이 필요하나, 이는 상당한 비용과 시간을 요구한다. 소프트웨어 확인 및 검증 작업에 있어서 산업계에서 가장 널리 쓰이고 있는 방법으로는 인스펙션(inspection)을 들 수 있는데, 인스펙션은 효과적인 분석 방법임에도 불구하고 많은 인력을 필요로 하는 작업이다. 따라서, 효과적으로 확인 및 검증 작업을 수행하기 위해서는 이러한 인스펙션 작업을 지원하는 소프트웨어 도구가 필요한데, 이러한 목적을 만족시키기 위해여 개발된 소프트웨어 도구가 SIS-RT이다.

SIS-RT (Software Inspection Support and Requirements Traceability)는 원자력발전소의 안전성과 관련된 고신뢰도 소프트웨어의 자동화된 확인 및 검증(Verification and

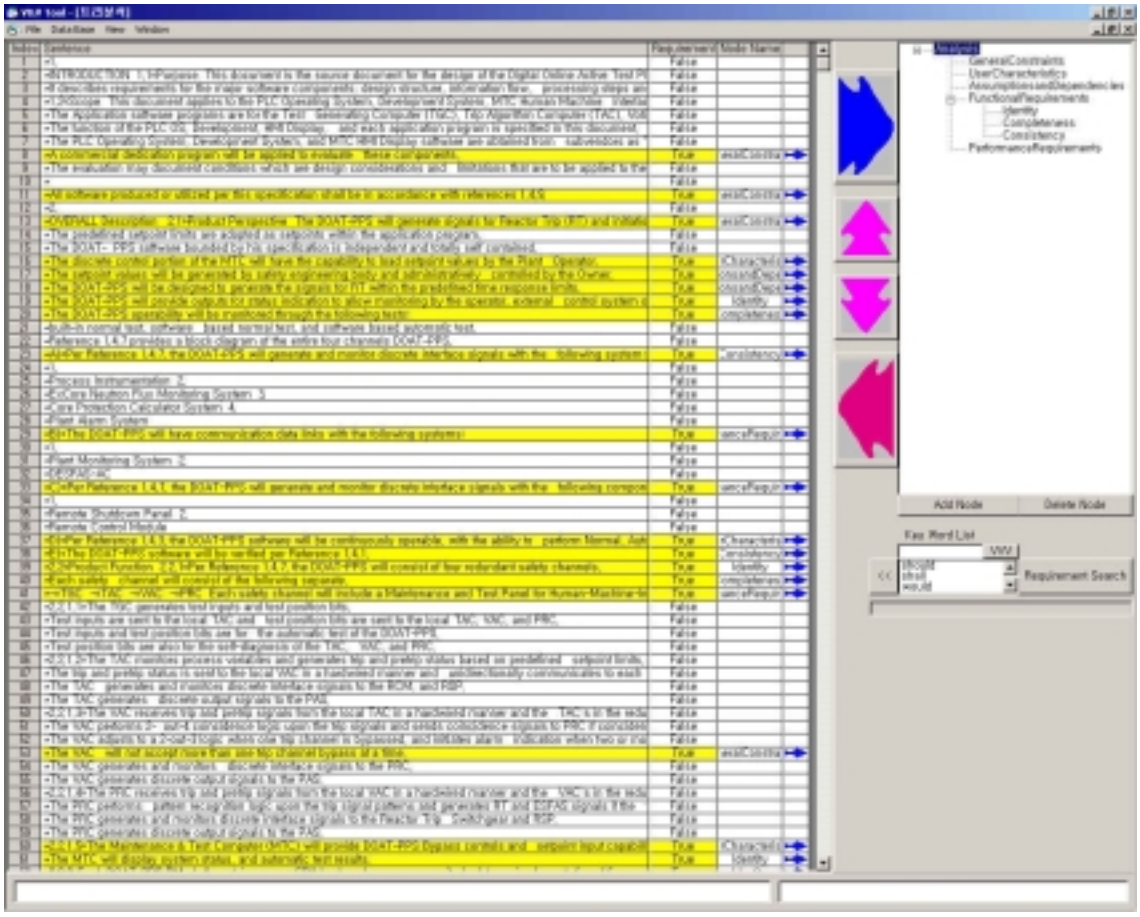

그림 1 SIS-RT에서 문서 트리 분석 화면

Validation) 도구이다. 그림 1, 2는 SIS-RT의 두 창인 문서 트리 분석과 문서 추적성 분석 화면을 보여주고 있다. 현재 SIS-RT는 문서 트리 분석에서 소프트웨어 개발 단계 중 생성된 문서에서 요구사항 도출, 체크리스트 구성, 분석된 문서 생성을 지원하고 있으며, 문서 추적성 분석에서는 여러 문서 사이에서의 응집력을 높일 수 있도록 문서들 사이의 추적성 분석을 지원하고 있다. 이와 함께 SCR(Software Cost Reduction)정형기법을 이용한 정형적 요구사항 분석도구를 개발하여, 현재 SCR 명세(specification)를 작성할 수 있는 도구가 마련되어 있다. 또한, 인스펙션을 웹 상에서도 가능하도록 하기 위해서 웹지원 홈페이지가 제작되어서, 여러 사람들이 웹상에서 문서를 공유하고 검토하면서 의견을 개진할 수 있도록 지원하고 있다.

결국, SIS-RT는 문서 분석 도구, 문서 추적성 도구, SCR 정형적분석 도구, 그리고 웹지원 홈페이지의 4가지 요소를 통합하여, 원자력발전소 보호시스템과 같은 매우 높은 안전성과 신뢰도를 요구하는 소프트웨어에 대한 확인 및 검증을 지원하는 도구이다.

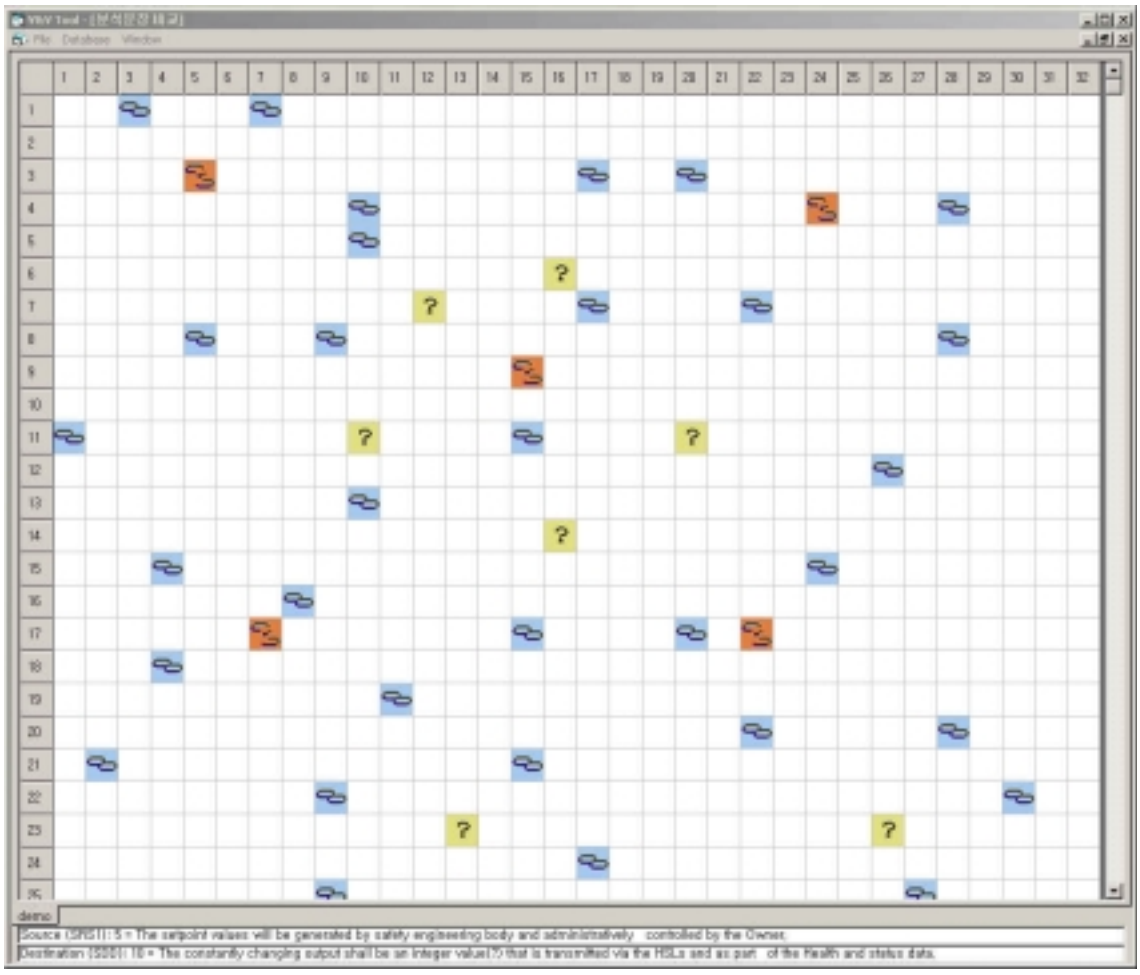

그림 2 SIS-RT에서 문서 추적성 분석 화면

II. 해결해야 할 문제

그림 2는 현재 작성된 SIS-RT에서 '문서 추적성 분석'을 실행했을 때의 모습이다. 문서 추적성 분석이란, 기존의 Functional Requirement (FR)에서 Software Requirement Specification (SRS)를 작성하였을 때, FR에 나타나있는 모든 내용이 SRS에도 나타나는가를 검사하기 위한 확인 및 검증 활동을 말한다. 현재는 FR이나 SRS 모두 자연어(natural language)로 쓰여지고 있으므로, 위에서 언급한 문서 추적성 분석을 수행하는데 있어서는 SRS에서 어느 부분에 FR의 어느 부분과 관련된 내용이 기술되어 있는가 하는 문제가 중요한 문제로 부각이 된다. 이에 따라, SIS-RT에서는 FR과 SRS에 있는 각각의 문장을 표의 형태로 나열한 뒤 (그림 2), 각 문장 사이의 연관관계를 정밀검사(inspection)를 수행하는 검사자(inspector)가 표시할 수 있도록 되어 있다. 이러한 SIS-RT가 KNICS과제에서 성능향상(upgrade)을 하는 단계에서, 기존에 검사자가 일일이 수행해 주어야 하는 연관관계에 대한 표시를 컴퓨터가 반자동화(semi-automation)해줄 수 있도록 수정이 될 계획이다.

이러한 일을 수행함에 있어서 중요하게 고려되는 것이 바로 FR중의 한 문장과 SRS중의 한 문장 사이에 가지는 연관관계를 정량적으로 평가하는 것이다.

III. 자연언어처리 (Natural Language Processing)와 정보검색(Information Retrieval)

자연언어처리는 음성인식까지를 포함하는 넓은 영역을 생각할 수도 있으나, 일반적으로는 텍스트(text)형태로 컴퓨터에 입력된 자연언어문장에 대해 수행하는 문장의 이해(natural language understanding)와 이를 바탕으로 하는 문장작성(text generation)까지의 영역을 의미한다. 이러한 자연언어처리는 주로 기계번역(machine translation)을 중심으로 발전해 왔으나, 그 이외에도 정보의 검색이나 대화시스템의 분야로도 발전을 하였다. 위의 문제를 해결하는데 있어서, 문헌조사(literature survey)는 자연언어처리 분야에서 정보검색의 분야에 초점을 맞추었다.

문헌조사에 있어서 자연언어의 정보검색에 초점을 맞춘 이유는 정보검색이 사용자가 입력한 자연언어를 바탕으로 해서 그에 대해 각 문헌에 대한 순위부여를 하는 것을 중심으로 발전되어 왔고, 이것이 현재 우리가 해결해야 할 문제와 매우 큰 유사성을 갖기 때문이었다.

1957년에 Luhn[1]이 처음으로 자연어를 이용한 정보검색에 있어서 문서들 내부의 특정한 용어와 사용자의 질의(query)에 포함되어 있는 용어들을 비교함으로써 자동적으로 정보검색을 수행해 줄 수 있는 시스템을 만들 수 있다고 하였다. 즉, 검색의 대상이 되는 문서들에 대한 분석을 통해서 그 문서들에게 중요하게 생각되는 용어(term)들을 추출해 내면 (이러한 용어들이 n개라고 가정할 때), 각각의 문서들은 각 용어의 포함 여부에 따라서 다음과 같은 용어벡터(term vector)의 형태로 표현이 가

능하다.

$$D = (t_1, t_2, t_3, \Lambda\ , t_n) \tag{1}$$

여기서, D : 특정한 문서에 대한 용어벡터

t_k : 문서가 특정한 용어를 포함하는가의 여부를 0 또는 1로 표현 (k = 1, 2,⋯, n)

이에 대해서, 정보요구(information request), 또는 질의어(query)도 이와 마찬가지로 용어벡터의 형태로 표현이 가능해진다.

$$Q = (q_1, q_2, q_3, \Lambda\ , q_n) \tag{2}$$

여기서, Q: 질의어에 대한 용어벡터

q_k : 질의어가 특정한 용어를 포함하는가의 여부를 0 또는 1로 표현 (k = 1, 2,⋯, n)

가장 간단하게 유사성을 정의하는 방법은 두 문서 사이에 함께 존재하는 용어의 개수로 정의하는 것으로 이 경우 유사성은 다음의 식으로 표현이 된다.

$$Similarity(D,Q) = \sum_{k=1}^{n} t_k q_k \tag{3}$$

하지만, 실제적으로는 각각의 용어들에 대해 서로 다른 가중치(weighting factor)를 줌으로써, 단순히 0과 1만을 이용하는 경우 보다 더 효과적으로 유사성을 측정하게 된다. 그림 3은 이렇게 용어들에 대해 가중치를 주는 것에 대한 한 가지 예를 나타낸 것이다. 이러한 가중치는 여러 가지 방식으로 줄 수 있으나 벡터의 정상화(normalization)를 이용하는 경우가 가장 대표적이며, 이 경우에 대해 문서와 질의어 사이의 유사성은 다음의 코사인벡터유사성공식(cosine vector similarity formula)에 의해 주어지게 된다. [3]

$$Similarity(D,Q) = \frac{\sum_{k=1}^{n} w_{qk} w_{dk}}{\sum_{k=1}^{n} (w_{qk})^2 \cdot \sum_{k=1}^{n} (w_{dk})^2} \tag{4}$$

여기서, w_{qk}: 질의어의 k번째 용어에 대한 weighting factor

w_{dk}: 문서의 k번째 용어에 대한 weighting factor (k = 1, 2,⋯, n)

Factors information help human operation retrieval systems

Query	human factors in information retrieval systems
VECTOR	(1 1 0 1 0 1 1)
Record1	containing human, factors, information, retrieval
VECTOR	(1 1 0 1 0 1 0)
Record2	containing human, factors, help, systems
VECTOR	(1 0 1 1 0 0 1)
Record3	containing factors, operation, systems
VECTOR	(1 0 0 0 1 0 1)

SIMPLE MATCH		WEIGHTED MATCH	
QUERY	(1 1 0 1 0 1 1)	QUERY	(1 1 0 1 0 1 1)
Rec 1	(1 1 0 1 0 1 0)	Rec 1	(2 3 0 5 0 3 0)
	(1 1 0 1 0 1 0) = 4		(2 3 0 5 0 3 0) = 13
QUERY	(1 1 0 1 0 1 1)	QUERY	(1 1 0 1 0 1 1)
Rec 2	(1 0 1 1 0 0 1)	Rec 2	(2 0 4 5 0 0 1)
	(1 0 1 1 0 0 1) = 3		(2 0 0 5 0 0 1) = 8
QUERY	(1 1 0 1 0 1 1)	QUERY	(1 1 0 1 0 1 1)
Rec 3	(1 0 0 0 1 0 1)	Rec 3	(2 0 0 0 2 0 1)
	(1 0 0 0 0 0 1) = 2		(2 0 0 0 0 0 1) = 3

그림 3 Term Weighting에 대한 간단한 예 [2]

이제 해결해야 할 문제는 어떠한 용어들을 선정할 것인가 하는 문제와 선정된 각각의 용어에 줄 가중치를 어떻게 선정할 것인가 하는 것이다. 우선 어떠한 용어를 선정할 것인가 하는 문제에 앞서서 용어를 선정하는 방식에 대한 고려 또한 필요할 것이다. 하나의 예로, plant protection system과 같이 세 단어로 구성되어서 하나의 의미를 갖는 용어에 대해 이를 세 개의 용어로 구분해서 고려할 것인지, 아니면 하나의 단어로 구분할 것인가 하는 것 또한 고려해야 할 대상이 되는 것이다. 당연히 이를 하나의 용어로 구성하는 것이 두 문장을 비교하는데 있어서 더 효율적인 결과를 줄 수 있을 것으로 예상된다. 하지만, 이를 위해서는 소프트웨어가 이러한 종류의 용어를 하나의 용어로 인식할 수 있도록 하기 위한 용어사전이 준비가 되어야

할 것이지만, 아직은 이러한 용어사전을 구하는 일이 그리 여의치 못한 실정이다. 이러한 용어사전을 자동으로 생성하는 알고리즘들 역시 많이 제안되어 왔지만, Salton은 약 25년간 자료검색과 관련된 문헌들을 조사한 결과에 의해 일반적인 경우에 대해서는 단순히 하나의 단어로 하나의 용어를 정의하는 것이 가장 효율적인 것이라는 결론은 내렸다. [3] 하지만, 현재 고려대상이 되고 있는 문서들이 지극히 한정적인 범위 내에서 쓰이고 있다는 점을 고려할 때, 몇몇 중요한 단어에 대해서는 위와 같은 여러 단어로 구성된 용어에 대한 사전을 구성하는 방안에 역시 고려의 대상이 되고 있다.

둘째, 각각의 용어에 대한 가중치를 선정하는 방법에 대해서는 자료검색의 경우에 있어서는 세 가지 항목을 고려하였다. 그 세가지는 용어빈도항목(term-frequency component), 전체용어빈도항목(collection-frequency component), 그리고 정상화항목(normalization component)이다. 용어빈도항목이란, 질의어에 있는 하나의 용어에 대해서 각각의 문서에서 그 용어가 몇번이나 나타났는가 하는 것을 고려하는 것으로, 질의어에 있는 하나의 용어가 더 많이 나타나는 문서가 더 많은 연관성을 가지고, 이에 따라 더 높은 유사성을 가져서 최종적으로 더 높은 우선순위를 부여받아야 한다는 개념에 있다. 이는 해결하고자 하는 문제가 FR에 있는 하나의 문장이 SRS에 있는 어떤 문장과 가장 깊은 유사성을 가지고, 그 유사성의 정도에 따라 SRS의 각 문장별로 우선순위를 부여해야 하는 가에 있는가 하는 것과 일치하는 면은 있으나, 하나의 문장에서 동일한 용어가 두 번 이상 출현할 가능성이 그리 많지 않을 것이라고 판단되므로, 최종적으로 용어빈도항목에 대한 고려는 배제하기로 결정하였다.

전체용어빈도항목이라, 자료검색에 있어서는 전체 문서들에 대해서 그 출현하는 빈도가 적은 단어일수록 높은 가중치를 받아야 한다는 개념이다. 예를 들면, a나 the와 같이 문서 중에 일반적으로 많이 포함되어 있는 단어보다는 ‘pressurizer’나 ‘condenser’와 같이 흔히 쓰이지 않는 단어와 질의어에 있는 단어가 일치했을 때, 더 높은 우선순위를 부여 받을 수 있도록 가중치를 배정해야 한다는 것이다. 이는 흔히 역문헌빈도(Inverse Document Frequency, IDF)를 통해 정량적으로 정의되는데, 정의하는 방법에는 다음의 세 가지 종류가 있다.

$$IDF = \log_2 \frac{N}{n} + 1 \quad [4] \tag{5}$$

$$IDF = \log_2 \frac{\max n}{n_i} + 1 \quad [5,6] \tag{6}$$

$$IDF = \log_2 \frac{N - n}{n_i} \quad [7] \tag{7}$$

여기서, N = 장서내의 문헌수

n = 장서내의 특정 용어에 대한 문헌수

maxn = 장서에서 용어에 대한 최대 빈도수

이러한 역문헌빈도 이 외에도 잡음측정이나 엔트로피 측정을 이용하는 방법도 제안되었다.

표준잡음 = maxnoise – 잡음 (8)

$$\text{잡음}_i = \sum_{k=1}^{N} \frac{Freq_{ik}}{TFreq_i} \log_2 \frac{TFreq_i}{Freq_{ik}} \quad [8] \tag{9}$$

$$\text{엔트로피}_i = 1 - \frac{\sum_{k=1}^{N} \frac{Freq_{ik}}{TFreq_i} \log_2 \frac{TFreq_i}{Freq_{ik}}}{\log_2 N} \quad [9] \tag{10}$$

여기서, N = 장서내의 문헌수

maxnoise = 장서에서 가장 높은 용어 잡음

$Freq_{ik}$ = 문헌 k에서 용어 i에 대한 빈도수

$Tfreq_i$ = 장서내의 용어 i에 대한 총 빈도수

마지막으로, 정상화항목은 문서들 사이에 서로 그 크기가 다른 것을 보상해 주기 위해 고려되는 것이다. 예를 들면, 문서A는 매우 짧고, 문서B는 매우 길다고 가정할 때, 확률적으로 문서B가 문서A에 비해서 사용자가 입력한 질의어를 포함할 가능성이 더 커지게 된다. 이는 결국 거의 대부분의 질의어에 대해서 길이가 긴 문서B가 길이가 짧은 문서A에 비해 더 높은 우선순위를 부여받을 확률이 높다는 것을 의미한다. 이러한 것을 보상해 주기 위해서, 개념적으로는 가중치를 문서의 길이로 나누어주어서 모든 문서를 동일한 조건에서 검색이 되도록 하는 것이 필요하다. 이러한 문서의 서로 다른 길이에 대한 보상을 위한 정상화항목은 위의 코사인벡터유사성공식(4)에 이미 고려가 되어 있다.

IV. 제안되는 SIS-RT 도구개선 방향

그림 4는 문장비교 및 정보검색을 위한 전체적인 흐름도를 보여주고 있다. 정보검색의 경우 사용자가 입력하는 자연어로 된 질의어가 ‘문장1’이 되고, 문서에 있는 요구사항을 나타내는 각종 문장들이 ‘문장2’가 된다. 문장비교의 경우에는 바로 ‘문장1’과 ‘문장2’가 비교의 대상이 되는 두 문장이 된다. 이렇게 자연어로 입력된 두 문장은 동의어사전(thesaurus)에 따라 처리된 문장인 ‘문장I’과 ‘문장II’로 변형이 되고, 이 두 문장은 유사성을 정량화 할 수 있는 알고리즘에 의해 그 유사성이 정량적으로 평가된다. 이러한 유사성의 정량적 평가에는 그림 4에 나타나있는 바와 같이, 문장 구조의 특성을 이용한 방법론 또한 고려될 수 있을 것이다. 이렇게 정량적

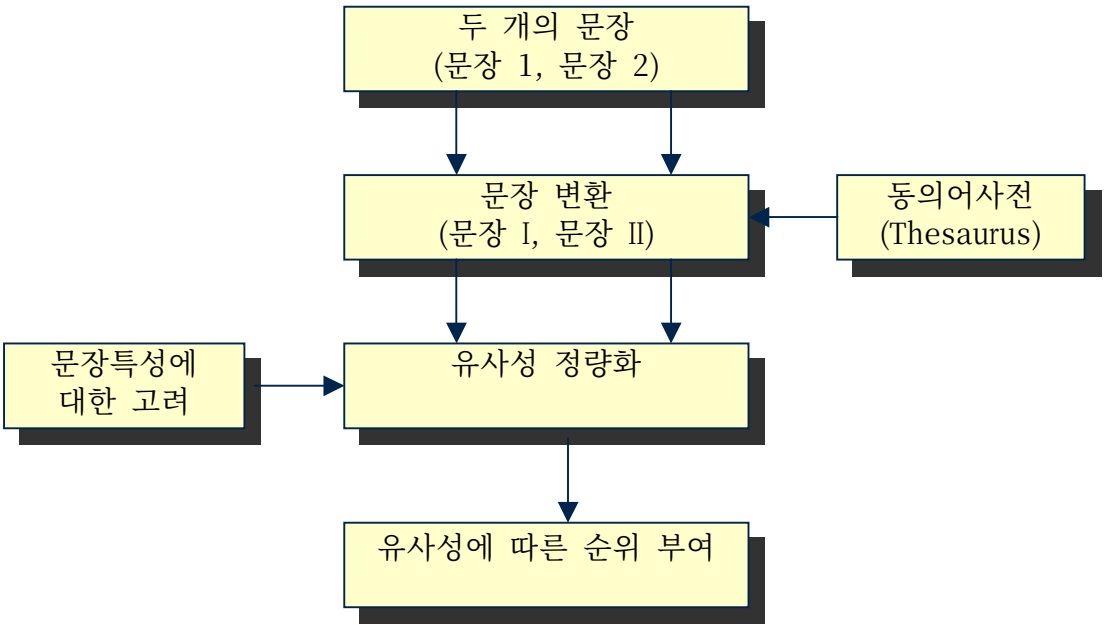

그림 4 문장비교 및 정보검색을 위한 흐름도

으로 평가된 유사성을 바탕으로 하나의 '문장1'과 여러 '문장II'들에 대해서 그 우선순위를 부여하게 된다.

문장의 유사성 정량화에 대해서는 위에서 언급한 고려사항들을 바탕으로 다음의 방법론을 제안한다.

(1) 대상이 되는 두 문서에 공통적으로 포함되는 용어들을 검색하여, 역문헌빈도에 따라 용어벡터의 대상이 되는 용어들을 결정한다.
(2) 두 문서 각각에 대해 역문헌빈도를 결정한다.
(3) 비교의 대상이 되는 두 문장에 대해 용어벡터를 결정하고, 코사인벡터유사성공식에 따라 두 문장의 유사성을 비교한다.

그림 4에 나타나 있는 바와 같이 위의 유사성 정량화 방법론 이외에도 동의어사전(thesaurus)의 활용 및 문서내에서의 문장의 특성에 기반하는 방법론 또한 함께 고려될 수 있을 것이다.

V. 적용 결과

그림 5는 실제 문서에 대해서 제안된 문서 추적성 분석 방법론을 적용한 결과를 보여주고 있다. 그림에서 세로로 1에서 21까지 나타난 수치는 비교하고자 하는 두 문서 중 전단계문서(source)에서 요구사항으로 판정된 문장들에 대한 일련번호를 나

타낸 것이다. 마찬가지로 가로로 120에서 152까지 나타난 수치는 후단계문서(destination)에서 요구사항으로 판정된 문장들에 대한 일련번호를 나타낸 것이다. 그림 5에서 창 아래부분에 사용자의 마우스 클릭에 의해서 선택된 전단계문서의 문장과 후단계문서의 문장이 나타나 있음을 알 수 있다. 현재는 전단계문서의 17번 문장과 후단계문서의 143번 문장이 나타나있다.

그림 5에서 화면 전체에 나타나는 표 안의 수치들은 전단계문서의 한 문장과 후단계문서의 한 문장 사이의 유사성을 나타내고 있다. 한 예로 현재 선택되어있는 전단계문서의 17번 문장과 후단계문서의 143번 문장 사이에의 유사도는 99.2%인 것으로 판정되었다.

실제 문서에 제안된 방법론은 적용해 본 결과 유사도가 80%이상으로 나타난 문장들의 경우 두 문장이 거의 일치하여 SIS-RT 도구가 자동으로 추적성 성립을 판정하여도 무방하다는 결론을 얻을 수 있었다. 다음은 이와 같은 문장에 대한 한 예를 나타내고 있다.

전단계문서 : If there is bypass condition, all AMS outputs except bypass information shall be automatically blocked.

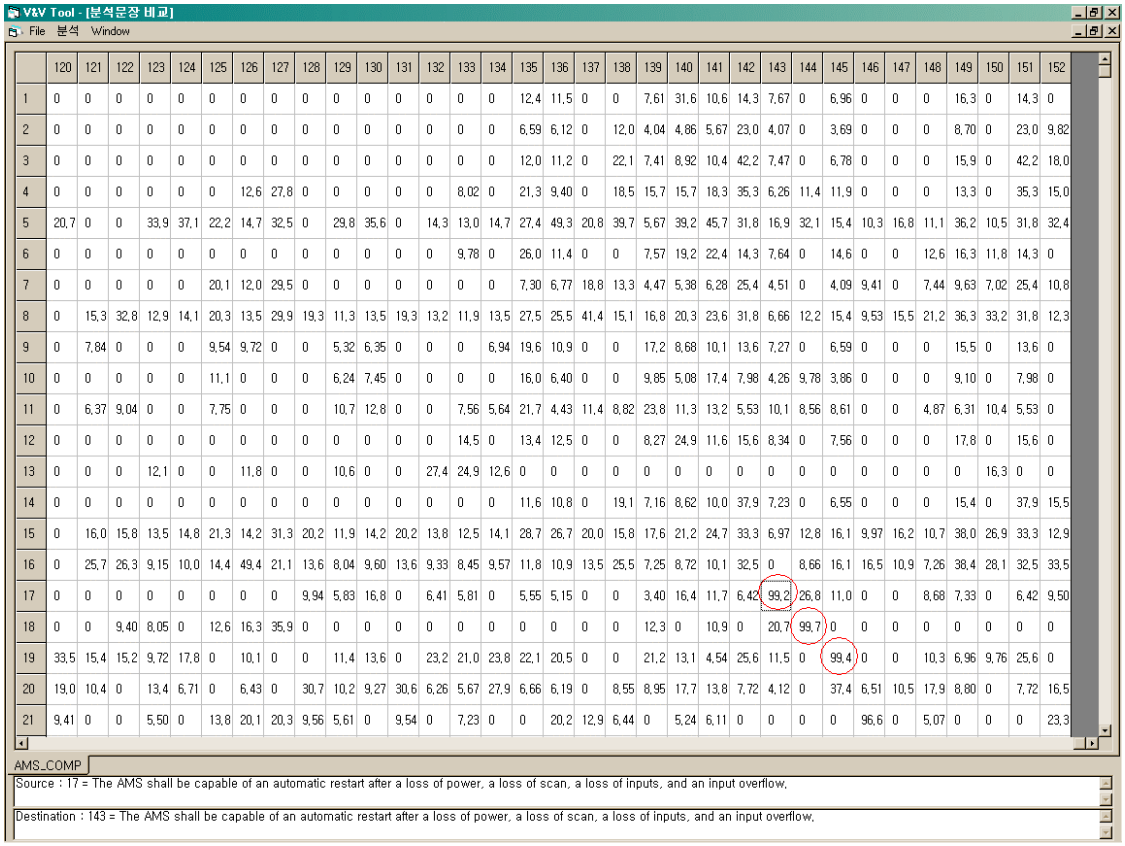

그림 5 SIS-RT에서 문서 추적성 분석 방법론 적용의 결과

후단계문서 : All AMS outputs except operating bypass information shall be automatically blocked.
유사도 = 81.4%

제안된 방법론에 따라 유사도를 계산하였을 때, 유사도가 30~80%의 결과는 검토자에 의한 추가적인 검토가 필요한 것으로 판단되었다.

제안된 방법은 문장에서 각 단어가 가지는 의미까지 모든 것을 고려할 수 있는 방법론은 아니다. 따라서, 경우에 따라서는 실제 추적성이 더 큰 두 문장이 제안된 방법에 따라 유사도를 계산하면 더 낮게 나오는 경우도 발생한다. 다음은 그에 대한 한 예이다.

(예제 1)
전단계문장 : Hysteresis shall be considered in programming for the AMS application program.
후단계문장 : To prevent signal oscillation, the bistable algorithm shall utilize a predefined value for hysteresis function.
유사도=26.6%

(예제 2)
전단계문장 : Hysteresis shall be considered in programming for the AMS application program.
후단계문장 : Each CPM will contain a functionally identical application program.
유사도=44.2%

위의 두 예제에서 실제 예제1의 경우가 두 문장사이의 추적성이 더 크지만, 제안된 방법론에 의해 계산된 유사도는 예제2가 더 크게 나타나는 것을 알 수 있다. 하지만, 이러한 경우는 극히 드물게 나타나는 경우로, 전체적으로 제안된 방법론은 추적성 평가에 있어서 매우 유용한 것으로 판단된다.

VI 결론

본 연구에서는 고신뢰도 소프트웨어를 위한 확인 및 검증 도구인 SIS-RT에서 문서 추적성 평가를 위한 방법론을 제안하였다. 제안된 방법론은 문서 사이의 추적성 분석에 유용한 것으로 나타났으며, 현재 SIS-RT에 적용되어 있는 상태이다.

References

[1] H. P. Luhn, “A Statistical Approach to the Mechanized Encoding and Searching of Literary Information”, *IBM Journal of Research and Development*, vol. 1, no. 4, pp.309-317, 1957

[2] 류근호, 김진호 공역, *정보검색*, 시그마프레스, 대한민국, 1995

[3] G. Salton and C. Buckley, “Term-Weighting Approaches in Automatics Text Retrieval”, *Information Processing and Management*, vol. 25, no. 5, pp.513-23, 1988

[4] K. Sparck Jones, “A Statistical Interpretation of Term Specificity and Its Application in Retrieval”, *J. Documentation*, vol. 28, no.1. pp.11-20, 1972

[5] K. Sparck Jones, “Experiments in Relevance Weighting of Search Terms”, *Information Processing and management*, vol.15, no.3, pp.133-144, 1979

[6] K. Sparck Jones, “Search Term Relevance Weighting Geven Little Relevance Information”, *J. Documentation*, vol.35, no.1, pp.30-48, 1979

[7] W. B. Croft and D. J. Harper, “Using Probabilistic Models of a Document Retrieval Without Relevance Information”, Documentation, vol.35, no.4, pp.385-395, 1979