ISSN 2288-4866 (Print)
ISSN 2288-4882 (Online)
http://www.jiisonline.org
J Intell Inform Syst 2020 March: 26(1): 47~73 http://dx.doi.org/10.13088/jiis.2020.26.1.047

사례기반추론과 텍스트마이닝 기법을 활용한 KTX 차량고장 지능형 조치지원시스템 연구

이형일
한양대학교 일반대학원 비즈니스인포매틱스학과
(*usdt204@hanyang.ac.kr*)

김종우
한양대학교 경영대학 경영학부
(kjw@hanyang.ac.kr)

KTX 차량은 수많은 기계, 전기 장치 및 부품들로 구성되어 있는 하나의 시스템으로 차량의 유지보수에는 상당히 많은 전문성과 유지보수 작업자들의 경험을 필요로 한다. 차량 고장발생 시 유지보수자의 지식과 경험에 따라 문제 해결의 시간과 작업의 질적 차이가 발생하며 그에 따른 차량의 가용율이 달라진다. 일반적으로 문제 해결은 고장 매뉴얼을 기반으로 하지만 경험이 많고 능숙한 전문가의 경우는 이와 더불어 개인의 노하우를 접목하여 신속하게 진단하고 조치를 취한다. 이러한 지식은 암묵지 형태로 존재하기 때문에 후임자에게 완전히 전수되기 어려우며, 이를 위해 사례기반의 철도차량 전문가시스템을 개발하여 데이터화된 지식으로 바꾸려고 하는 연구들이 있어왔다. 하지만, 간선에 가장 많이 투입되고 있는 KTX 차량에 대한 연구나 텍스트의 특징을 추출하여 유사사례를 검색하는 시스템 개발은 아직 미비하다. 따라서, 본 연구에서는 이러한 차량 유지보수 전문가들의 노하우를 통해 수행된 고장들에 대한 진단과 조치 이력을 문제 해결의 사례로 활용하여 새롭게 발생하는 고장에 대한 조치가이드를 제공하는 지능형 조치지원시스템을 제안하고자 한다.

이를 위하여, 2015년부터 2017년동안 생성된 차량고장 데이터를 수집하여 사례베이스를 구축하였고, 차원축소 기법인 비음수 행렬 인수분해(NMF), 잠재의미분석(LSA), Doc2Vec을 통해 고장의 특징을 추출하여 벡터 간의 코사인 거리를 측정하는 방식으로 유사 사례를 검색하였으며, 위의 알고리즘에 의해 제안된 조치내역들 간 성능을 비교하였다. 분석결과, 고장 내역의 키워드가 적은 경우의 유사 사례 검색과 조치 제안은 코사인 유사도를 직접 적용하는 경우에도 좋은 성능을 낸다는 것을 알 수 있었고 차원 축소 기법들의 성능 비교를 통해 문맥적 의미를 보존하는 차원 축소 방식 중 Doc2Vec을 적용하는 것이 가장 좋은 성능을 나타낸다는 것을 알 수 있었다.

텍스트 마이닝 기술은 여러 분야에서 활용을 위한 연구들이 이루어지고 있는 추세이나, 본 연구에서 활용하고자 하는 분야처럼 전문적인 용어들이 다수이고 데이터에 대한 접근이 제한적인 환경에서 이러한 텍스트 데이터를 활용한 연구는 아직 부족한 실정이다. 본 연구는 이러한 관점에서 키워드 기반의 사례 검색을 보완하고자 텍스트 마이닝 기법을 접목하여 고장의 특징을 추출하는 방식으로 사례를 검색해 조치를 제안하는 지능형 진단 시스템을 제시하였다는 데에 의의가 있다. 이를 통해 현장에서 바로 사용 가능한 진단시스템을 단계적으로 개발하는데 기초자료로써 시사점을 제공할 수 있을 것으로 기대한다.

주제어 : KTX, 고장진단시스템, 사례기반추론, 차량고장, 텍스트마이닝

논문접수일 : 2019년 11월 14일 논문수정일 : 2020년 3월 10일 게재확정일 : 2020년 3월 14일
원고유형 : 일반논문 교신저자 : 김종우

1. 서론

우리나라 고속철도(Korea Train eXpress, KTX)는 2004년 4월 첫 운행을 시작한 이래로 2019년 4월에 개통 15주년을 맞이했다. KTX로 우리나라는 세계에서 5번째 고속철도 운영국이 되었으며, 15년간 100만회를 운행하면서 총 4억 2천만 Km를 달렸다. 하루 평균 300회 이상을 운행하며 개통 초기에 비해 점점 더 많은 차량이 투입되어 운행되고 있고, 차량의 노후화가 진행됨에 따라 차량의 유지관리에 대한 비용이 많은 비중을 차지하고 있다. 특히 철도차량은 수많은 기계, 전기 장치 및 부품들로 구성되어 있는 하나의 복잡한 시스템으로 고장 매뉴얼에서조차 KTX의 모든 고장에 대한 해결책을 제시하는데 한계를 가지고 있다고 서술하고 있다. 따라서, 차량 운행의 안정성 확보와 가용율을 높이기 위해서는 부품의 수명주기를 분석한 예방보수 체계 확립과 함께 유지보수 전문가의 확보가 필수적이다. 일반적으로 차량에 고장이 발생하면 고장 매뉴얼을 기반으로 진단의 올바른 방향을 찾고 계획을 구체화하여 문제 해결의 도움을 얻지만, 이 외에도 유지보수 작업자의 설비에 대한 이론적 지식과 오랜 시간 쌓아온 현장에서의 유지보수 경험이 접목되어 조치가 이루어진다. 이러한 경험 기반의 지식은 암묵지 형태로 존재하고, 유지보수는 지역적으로 산재되어 있는 차량기지에서 이루어지기 때문에 전문성 있는 개인의 노하우를 모아 전사적으로 전수하는 일은 쉽지 않다. 이러한 문제들을 해결하기 위해 고속차량의 유지보수 체계를 확립하여 주기적인 정비를 시행하고 있으며, 차량 전문가 양성을 위한 교육도 지속적으로 시행하고 있다. 또한 KTX차량의 장애통계 분석을 통해 원인을 파악하고 차량의 운행 패턴 변화에 맞춘 적정 유지관리 전략을 세우기 위한 시도도 있어왔고(Choi and Kim, 2014), 신뢰성 중심의 유지보수(Reliability Centered Maintenance, RCM) 체계, 스마트 팩토리 도입 등 지속적인 KTX차량의 장애 경감에 노력하고 있다. 여러가지 노력에도 불구하고 개통 이후 15년 동안 축적된 이 방대한 양의 데이터를 분석하고 활용하는 시스템은 미비하다. 이는 현재 차량 고장과 관련된 데이터는 차량의 차상 컴퓨터 시스템(On Board Computer System, OBCS)에서 전송하는 고장코드와 현장 전문가들이 실제로 진단하고 조치했던 경험을 데이터베이스에 저장하여 전산화 관리하고 있지만, 대부분의 속성들이 텍스트 기반의 비정형 데이터로 되어 있어 정형데이터를 처리하고 분석하는 것에 비해 자연어 처리(Natural Language Processing, NLP) 방안을 기반으로 한 텍스트 마이닝과 같은 상당한 기술을 필요로 한다.

텍스트 마이닝은 여러 산업분야에서 활용을 위한 연구들이 이루어지고 있는 추세로 국내 연구에서는 주로 특정 이슈와 관련한 동향 분석이나 토픽 모델링을 활용한 주제분류, 감성분석 위주로 연구가 이루어지고 있고, 연구 대상도 온라인 상의 기사 또는 제품 리뷰에 국한되어 있다. 이에 반해, 본 연구에서 활용하고자 하는 분야처럼 텍스트 데이터를 대상으로 하는 전문가시스템의 경우 도메인의 특수성으로 인해 전문 용어들의 사용이 빈번하며 데이터 접근에 어느 정도 제한이 있어 이러한 텍스트 데이터를 활용하고자 하는 연구는 미비한 실정이다. 점차 기업 내에서 활용하고자 하는 데이터가 비정형화 되어가는 추세에서 현장 전문가의 경험지식을 지식베이스로 관리하고 이를 실무적으로 활용할 수 있도록 전문가 시스템을 개발하고자 하

는 연구는 이러한 관점에서 중요한 분야 중 하나이다.

최근 철도분야에서도 단계적으로 텍스트 데이터를 활용하기 위한 기반 연구들이 이루어지고 있다. 철도사고를 기록한 텍스트 데이터에서 사고원인, 조치, 대책에 대한 키워드를 분석하고 원인과 대책의 연관관계 분석을 통해 철도사고 예방과 관련된 안전 정책을 마련하고자 최초로 텍스트 마이닝을 시도한 연구가 있었다(Eom, 2019). 또한 전기, 시설 등과 같은 철도시설물에 발생한 장애를 조치한 전문가들의 경험지식을 체계적으로 지식화 할 수 있도록 MCRDR (Multiple Classification Ripple Down Rule)의 방법론을 적용하여 사례기반의 전문가시스템 기반 모형을 제공한 연구도 있다(Ahn and Park, 2019). 그러나 아직까지 KTX 철도차량의 고장 내역과 조치에 대한 경험지식을 지식베이스로 구축하거나 이러한 지식베이스를 통해 구축된 텍스트 데이터를 실무적으로 활용하기 위한 전문가 시스템을 개발한 사례는 없다. 따라서 본 연구에서는 이러한 텍스트 기반의 데이터를 효율적으로 처리하는 텍스트 마이닝 기법을 활용하여 새롭게 발생하는 고장을 과거 사례로부터 검색하여 조치할 수 있도록 지원하는 지능형 고장 진단 방안을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 서론에 이어서 2장에서 본 연구에 활용된 관련 연구를 검토하며, 3장에서는 본 연구에서 제시하는 지능형 조치지원시스템의 연구방안을 제시한다. 이어서 4장에서 모델의 유용성을 확인하기 위한 실험 결과를 분석한다. 5장에서는 시사점을 서술하며, 마지막 6장에서는 결론과 본 연구의 한계점 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 사례기반추론(Case-based Reasoning, CBR)

사례기반추론이란 과거의 비슷한 경우에 대한 경험들을 인식하고, 그 경우들에 대한 지식을 현재의 문제에 적용하는 기법이다. 사례기반추론은 메모리기반추론(Memory-based Reasoning, MBR)이라고도 하며, 데이터의 형식에 관계없이 그대로 사용될 수 있기 때문에 다른 분석 기법으로 다루기 힘든 텍스트, 이미지, 음향 등의 비정형 데이터 처리에 강점을 가지고 있어 의료분야의 비정상 판독, 음악 조각으로 곡을 식별하는 등의 다양한 응용분야에 사용되고 있다(Yoon et al., 2013; Wang, 2003). 또한 새로운 데이터를 기존의 사례데이터에 단순히 추가함으로써 새로운 범주에 대한 분류나 추정이 가능하기 때문에 새로운 데이터가 추가되어도 모델을 개발하는 학습 과정이 필요하지 않다는 장점이 있다. 그에 비해 유사사례 검색을 위한 과정이 필요하기 때문에 분류나 예측을 위해 데이터에 대한 학습을 통하여 모델을 구축하는 신경망이나 의사결정나무와 비교하여, 사례베이스의 크기가 큰 경우에는 유사 사례 검색에 시간이 오래 걸린다는 단점이 있다. 또한 사례기반추론은 사례베이스로부터 유사 사례를 찾기 위한 거리함수를 정하는 방식에 따라 결과가 달라지기 때문에 적절한 거리함수를 찾기 위한 다양한 시도와 경험을 필요로 한다(Linoff and Berry, 2011). 사례기반추론의 문제해결 과정은 크게 검색(retrieve), 재사용(reuse), 수정(revise), 유지(retain)의 4단계로 구분된다(Aamodt and Plaza, 1994). 첫번째로 문제가 발생할 경우 검색 과정을 통해 과거 해결 사례 중 유

사 사례를 찾고, 두번째로 재사용 과정을 통해 유사 사례를 그대로 해결방안으로 사용하거나 수정하여 해결방안을 제시한다. 세번째로 교정 과정을 통해 제시된 해결방안의 타당성을 검증하고 타당하지 않은 경우 수정하여 개선된 방안을 제시하게 되며, 마지막으로 확정된 해결방안을 기존 사례 저장소에 추가하는 유지 과정을 거치게 된다.

사례베이스로부터 유사 사례를 검색하는 방법으로 최근접 이웃 검색(Nearest Neighbor Retrieval)이 대표적이며, 현재 문제와 사례베이스의 모든 사례의 유사도를 측정하여 가장 유사도가 높은 사례들을 검색하는 방법이다(Lee and Myoung, 2008). 유사도의 측정을 위한 거리함수로는 유클리디안 거리(Euclidean Distance), 맨하탄 거리(Manhattan Distance), 코사인 거리(Cosine Distance) 등이 사용되며, 본 연구에서는 유사도를 산출하기 위해 코사인 거리를 사용하여 유사 사례를 검색하였다. 자세한 방법은 3절에서 설명한다.

2.2 고장진단시스템

고장진단이란 시스템이 정상적인 기능을 하지 않을 때 그 원인을 조직적이고 체계적으로 규명하는 하나의 추론과정이다(Lee, 1992). 고장진단에 일반적으로 사용되는 기법에는 모델기반추론, 사례기반추론, 규칙기반추론(Rule-based Reasoning, RBR) 및 정성적 추론(Qualitative Reasoning, QR)이 있다(Lee, 1998). 고장진단의 대표적인 응용이 전문가 시스템이며, 대부분의 전문가 시스템의 경우 규칙기반 시스템이나 사례기반 시스템이 주를 이룬다. 규칙기반의 고장진단은 지식을 시스템이 이해할 수 있는 규칙의 형태로 표현 가능한 경우 잘 동작하는 특성이 있지만 시스템 반영을 위하여 모든 지식을 표현하는 것이 어렵다는 한계성 때문에 사례기반의 고장진단 방법을 사용하기도 한다. 사례기반추론은 규칙을 정형화하기 어렵거나 개념 정의가 잘 되어 있지 않은 경우에 사용하기 적절한 방법으로 실제 전동차와 도시철도차량의 고장진단에 사례기반추론을 접목하여 유사 사례를 검색할 수 있는 전문가 시스템을 개발하였다(Ahn and Park, 2006; Park, 2012). 이와 함께 사례베이스 증대에 따른 검색시간 지연 및 부적절한 사례 조회와 같은 사례기반추론의 한계점을 극복하기 위하여 인공신경망과 같은 규칙기반을 혼합한 하이브리드 방식의 고장진단을 통해 신뢰성을 높이고자 하는 방안도 제안되었다(Lee and Kim, 1998; Lee et al., 2006). 다음 <Table 1>은 기존 고장진단시스템과 관련된 연구 사례들을 정리한 표이다.

〈Table 1〉 Precedent Study on the Fault Diagnosis System

Author (Year)	Research overview
Kim et al. (2017)	Detecting Wind Turbine Blasde cracks by image classification using SVM
Song and Lim (2019)	Early detection of defects by diagnosing the condition of components using sensor data and predicting changes in condition
Jeon et al. (2015)	Suggesting the method of extracting and structured fault knowledge from a text-based fault analysis
Park et al. (2018)	Collecting text data by web crawling and classifying similar documents using Cosine Similarity
Ahn and Park (2019)	Establishing knowledge management model from failure-report

이상의 기존 선행 연구들을 데이터 구조 측면에서 본다면 주로 센서 데이터를 활용하는 정형 데이터 기반의 고장진단시스템과 텍스트 데이터를 활용하는 비정형 데이터 기반의 고장진단시스템 두 유형으로 나눠볼 수 있다. 전자의 경우 센서나 이미지를 통해 수집되는 데이터를 활용하여 기기나 장치의 상태변화 추이를 예측하고 이를 통해 결함의 조기감지 또는 이상상태를 판별하는 연구들을 진행하였으며, 후자의 경우 경험 지식을 체계적으로 데이터화 하여 사례기반의 전문가시스템을 구축하기위해 기반이 되는 텍스트 데이터에 대한 신뢰성을 높이는 방법론들을 연구하여 제안하였다.

센서 데이터를 활용하는 고장진단시스템의 경우 탐지나 예측과 같이 실용적 관점에서의 연구가 수행되어 왔으나, 비정형 기반의 고장진단시스템 관련 연구는 비교적 최근에 이루어지고 있는 추세이다. 고장진단분야에서 텍스트 데이터의 활용과 관련하여 수행된 주요 연구들은 저장하는 텍스트 데이터 자체의 신뢰성을 높이기 위한 방안으로 국한되어 있어, 아직까지 실무적으로 텍스트를 활용하기 위해 적절히 분석하는 방법을 제시하는 연구는 희소하다. 따라서 본 연구는 이러한 연구동향에서 고장진단 문제에 대해 텍스트 데이터를 활용하고자 시도하였다는 점이 기존 연구들과의 주요 차별점이다.

2.3 문서의 특징 추출

텍스트 기반의 비정형 데이터를 분석하기 위해서는 먼저 텍스트를 분석의 단위가 되는 토큰(token)으로 나눠야 한다. 한국어는 명사와 조사가 함께 사용되고, 용언에 여러 어미가 붙어 토큰화가 복잡하기 때문에 일반적으로 의미를 가지는 최소의 단위인 형태소(morpheme)를 분석하여 토큰화(tokenization)를 한다. 이를 단어-문서 행렬(Term-Document Matrix, TDM)을 통해 문서별로 나타난 단어의 빈도를 행렬 형태로 변환하게 되면 다양한 통계적 기법을 적용할 수 있게 된다. 하지만 단어-문서 행렬을 그대로 사용하게 되면 단어의 순서나 의미를 파악하지 않고 하나의 단어를 단순히 개별적인 하나의 차원으로 나타내기 때문에 문맥적 의미와 같은 특징들이 소실될 수 있는 단점이 있다. 이러한 문제점을 해결하기 위해 문서의 특징을 추출해 의미를 보존하면서 차원을 축소하는 다양한 방법들이 고안되었다. 대표적인 차원 축소 방법으로 비음수 행렬 인수분해(Non-negative Matrix Factorization, NMF), 특이값분해(Singular Value Decomposition, SVD) 등이 있고, 단어 임베딩의 분산 표현 방법으로 인공신경망 기법을 활용하는 Word2Vec을 문장 단위로 확장한 Doc2Vec과 Sent2Vec이 있다.

2.3.1 비음수 행렬 인수분해(Non-negative Matrix Factorization, NMF)

비음수 행렬 인수분해는 식 (1)과 같이 주어진 비음수 행렬(V)을 더 작은 크기의 비음수 가중치 행렬(W)과 비음수 특성 행렬(H) 두 가지 행렬의 곱으로 분해하는 알고리즘이며, 두 행렬의 곱이 원 행렬에 수렴할 때까지 식 (2)의 곱셈 갱신 규칙에 따라 두 행렬을 동시에 갱신한다(Lee and Seung, 1999; Heo and Jung, 2009).

$$V = WH \quad (V = m \times n, W = m \times r, H = r \times n) \tag{1}$$

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{(W^T V)_{\alpha\mu}}{(W^T W H)_{\alpha\mu}} \qquad W_{i\alpha} \leftarrow W_{i\alpha} \frac{(V H^T)_{i\alpha}}{(W H H^T)_{i\alpha}} \tag{2}$$

여기서 가중치 행렬은 문서와 특성 간의 관계를 나타내며, 문서별로 특성이 얼마나 반영되었는지를 알 수 있다. 특성 행렬은 특성에 대한 단어들의 관계를 나타내며, 어떤 단어들이 특성에 중요한 영향을 미치는지를 알 수 있다. 이런 이유로 비음수 행렬 인수분해를 사용하면 직관적으로 문서의 특성이 해석 가능하며, r의 수를 m과 n보다 작게 설정하기 때문에 특성을 보존하면서도 차원의 축소가 가능하다.

2.3.2 특이값분해(Singular Value Decomposition, SVD)

특이값분해는 식 (3)과 같이 주어진 행렬을 3개의 다른 행렬의 곱으로 분해하는 방식을 말한다.

$$A = U\sum V^T \tag{3}$$

$U: m \times m$ Orthogonal Matrix $(AA^T = U(\sum\sum^T)U^T)$

$V: n \times n$ Orthogonal Matrix $(A^TA = V(\sum^T\sum)V^T)$

$\sum: m \times n$ Diagonal Matrix

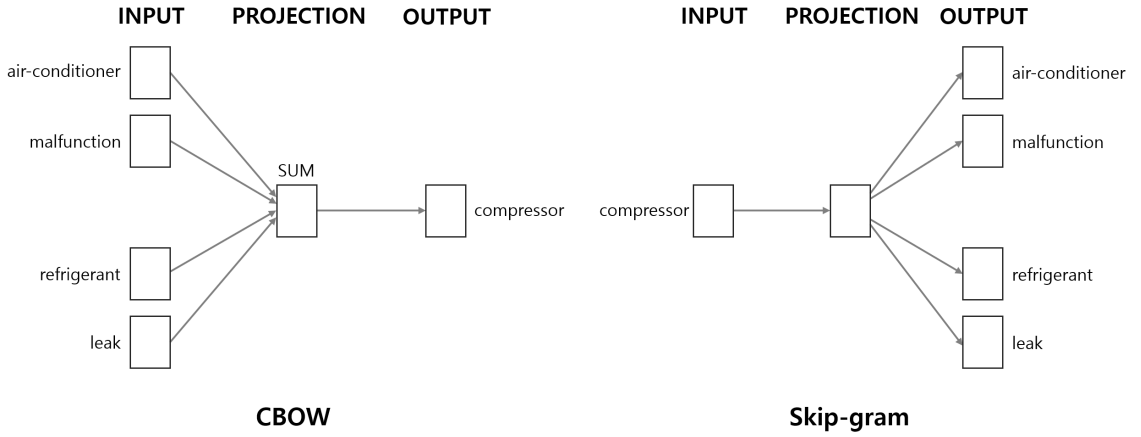
여기서 $\sum$는 대각 행렬로 대각선 상의 값을 제외한 나머지 원소가 0이고, 대각선 값은 고유값 분해를 통해 나오는 고유값의 제곱근(Square root)을 취한 값으로 특이값(Singular Value)이라고 부른다. 특이값분해를 통해 주어진 행렬이 U, V^T에 의해 방향이 바뀌며 특이값 만큼 크기가 변하기 때문에 차원 축소에 사용할 수 있다. 특이값분해는 각 행렬의 원소에 변화를 줌으로써 Thin SVD, Compact SVD, Truncated SVD과 같이 다양하게 변형되고 이 중 Truncated SVD를 단어-문서 행렬에 적용한 방식이 잠재의미분석(Latent Semantic Analysis, LSA)이다. Truncated SVD는 $\sum$행렬의 특이값 가운데 상위 일부만 추출한 형태이기 때문에 원래의 행렬로 복원이 되지 않지만 정보의 압축이 잘되는 특성이 있다. 따라서 잠재의미분석은 특이값 분해를 통해 차원을 줄이면서도 문서의 내재적 의미를 찾을 수 있어 유사 문서를 군집화 하는데 적용되고 있다(Park et al., 2017).

2.3.3 단어 임베딩(Word Embedding)

단어 임베딩은 문서에 포함되어 있는 단어들을 수치 값으로 변환하는 기법을 말하며, 대표적으로 Word2Vec, GloVe, FastText 등이 있다. Word2Vec은 2013년 구글에서 제안한 기법으로 CBOW (Continuous Bag-Of-Words)와 Skip-Gram 두 가지 방식이 있다. CBOW와 Skip-Gram은 모두 문서 내에 등장하는 단어를 대상 단어와 인접 단어로 하여 신경망을 학습시키는 방식이며, <Figure 1>에서 보는 바와 같이 CBOW는 인접 단어의 임베딩을 더해서 대상 단어를 예측하고, Skip-Gram은 대상 단어를 임베딩으로 인접 단어를 예측하도록 구성한다(Mikolov et al., 2013).

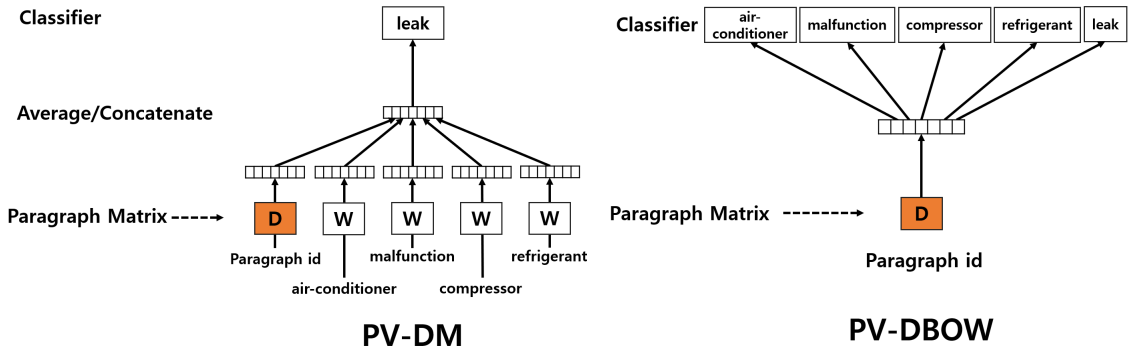
이에 착안하여 Word2Vec을 문장단위로 확장한 모델이 Paragraph Vector이며, Doc2Vec으로도 불린다. Word2Vec과 마찬가지로 PV-DM (Paragraph Vector with Distributed Memory)과 PV-DBOW (Paragraph Vector with Distributed Bag of Words)의 두가지 방식이 있으며 <Figure 2>와 같다(Le and Mikolov, 2014).

<Figure 2>에서 보는 것처럼 Paragraph id를 추가하여 문장을 하나의 id로하여 학습데이터에 추가하여 문장의 의미가 임베딩 되어진다. 이와 유사하게 Sent2Vec은 Word2Vec의 CBOW 모델을 문장 단위로 확장한 모델이며, CBOW와 달리 문



출처: Mikolov et al. (2014)

〈Figure 1〉 Word2Vec Architecture



출처: Le and Mikolov (2014)

〈Figure 2〉 Paragraph Vector Architecture

장의 모든 n-그램을 조합하여 학습하는 특징을 가진다(Park and Shin, 2018). 본 연구에서는 Doc2Vec의 PV-DBOW 방식을 적용하여 유사도 측정에 활용하였으며, 자세한 방법은 3절에서 설명한다.

3. 연구 방법

3.1 연구 프레임워크

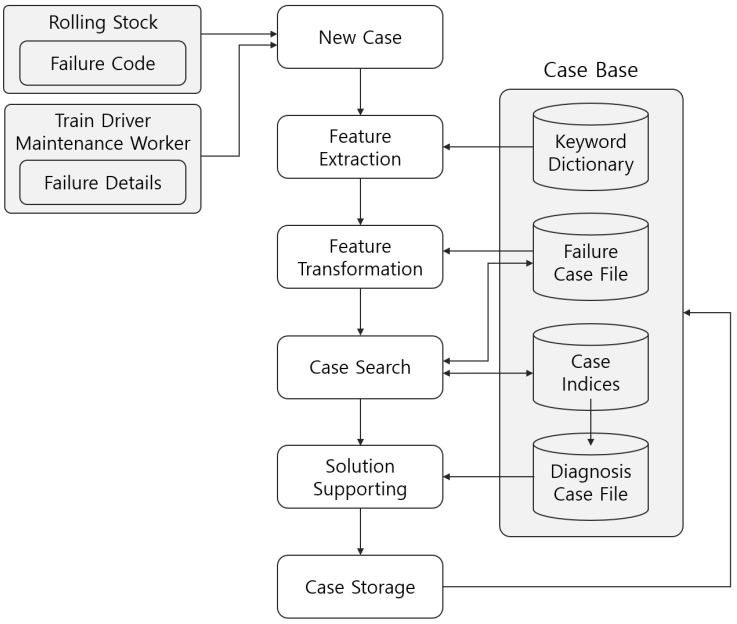
본 절에서는 사례기반추론을 기반으로 하여, 고장내역 간의 유사도를 계산하고 가장 유사한 3가지 사례의 조치내역을 추천하는 방안을 제시한

다. 본 연구에서 제안하는 모델은 다음의 <Figure 3>에서 제시된 것처럼 구현된다.

<Figure 3>에서 보는 바와 같이 지능형 조치지원 모형 프로세스는 기본적으로 사례기반의 문제해결 절차를 따르며 다섯 가지 단계를 거쳐 수행이 된다. 첫째, 새로운 고장이 발생한 경우 이 새로운 고장내역의 주요한 키워드를 통합사전 기반으로 추출한다. 둘째, 추출된 키워드를 사례베이스와 동일한 형태로 변환한다. 셋째, 사례베이스에서 가장 유사한 고장 사례를 찾기 위해 유사도를 계산한다. 넷째, 유사도가 가장 높은 사례 3가지를 선택하고, 이 사례들의 조치내역을 새로운 문제 해결을 위한 해답으로 추천한다. 다섯째, 새로운 고장을 기존 사례베이스와 비교하여 완전히 동일한 고장이 아닌 경우 사례베이스에 추가하고, 새로운 키워드도 통합사전에 없는 경우 추가하여 갱신한다.

3.2 분석 데이터 수집

본 연구에 사용된 데이터는 철도공사 전사적 자원관리 시스템(KOVIS)의 차량 유지보수이력 데이터를 사용하였다. 수집된 데이터는 2015년부터 2017년까지 3년동안 발생한 KTX 고장코드 테이블로, 변수는 총 28개이며, 82,430건의 레코드로 구성되어 있다. 이 중 사례베이스와 사전 구축에 필요한 I/S NO, 컴퓨터명, 계통, 발생구분, 장애코드, 고장내역, 조치내역을 주요변수로 사용하였고, 나머지 21개의 속성은 편성, 차량번호, 발생일자, 불량내역, 점검내역, 조치여부 등으로 고장내역과 조치내역에 내용이 일부 포함되거나 사례 비교에 의미가 없어 이 변수들은 제외하였다. I/S NO는 고장발생에 대한 일종의 조치 문서번호로써 고장사례를 구분할 수 있는 Key 속성이므로 중복 사례를 제거하는데 사용되었으



〈Figure 3〉 Procedure of the Proposed Model

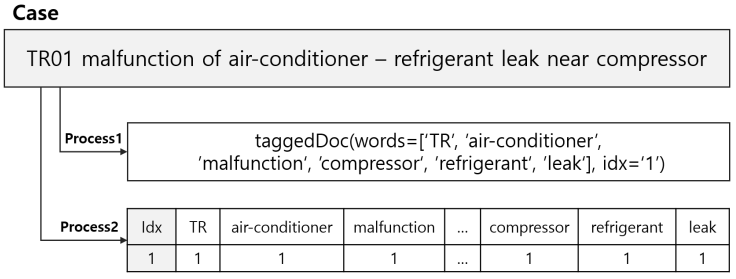
며, 본 연구의 목적이 유사한 고장을 판별하고 이에 대한 조치내역을 추천해주는 것이기 때문에 고장내역과 조치내역이 없는 경우의 데이터는 제거하여 총 82,405건의 데이터를 사용하였다.

3.3 통합사전 구축

본 연구에서는 철도차량 분야라는 특수성으로 인해 전문용어가 사용되는 경우가 많고, 장애 발생 시 차량에 코드형태로 현시되고 저장되므로 고장내역의 특징을 추출하는데 별도의 사전 구축이 필요했다. 또한 조치지원시스템이 제안한 조치내역이 실제 조치에 얼마나 도움이 되었는지에 대한 성능을 평가하기 위해 조치내역의 주요 키워드도 추출하여 사전으로 구축하였고, 키워드 관리를 용이하게 하기위해 하나의 통합사전으로 구축하였다. 고장코드 테이블의 모든 속성들은 텍스트 기반으로 작성되어 있어 사전구축을 세 단계로 진행하였다. 첫째, 파이썬 한국어 형태소 분석기 Konlpy[1] 패키지를 사용하여 고장내역과 조치내역의 명사만 추출한다. 둘째, 장애코드 변수에서 WW-XX-YY 형태의 6자리 장애코드를 추출하여 사전에 추가한다. 셋째, 컴퓨터명, 계통, 발생구분을 포함해서 차량부품 및 발생 위치를 나타내는 영문표기는 별도로 추출하여 사전에 추가한다. 예를 들어 MB라는 단어가 포함된 키워드는 차량의 모터블럭을 의미하며, TR(숫자)는 몇 번째 객차인지를 의미한다. 장애코드와 발생 위치는 고장을 특정하는 중요 키워드로 사전에 포함하였다. 본 연구에서는 사례베이스 82,405건에서 총 9,819개의 키워드로 구성된 통합사전을 구축하였다.

3.4 사례베이스 구축

본 연구에서는 문장 기반의 고장내역과 조치내역을 두가지 형태로 변환하여 사례베이스로 구축하였다. 첫번째는 사례에서 추출된 키워드에 인덱스 번호를 부여하여 튜플 형태로 변환하는 방법이다. 이 형태는 문서의 벡터 임베딩을 위한 학습 데이터의 입력형태로 사용되며, 저장된 모델을 사용하여 새로운 고장을 벡터 임베딩할 때도 사용된다. 또한 조치내역에 대한 추천모형의 성능평가 시 스코어링에도 유용하게 사용할 수 있다. 두번째는 고장내역을 단어-문서 행렬로 변환하는 방법이다. 단어-문서 행렬이란 문서별로 나타난 단어의 빈도를 표(행렬) 형태로 나타낸 것으로, 문서 간 유사도를 계산할 때 가



〈Figure 4〉 Example of Case Processed

1) https://konlpy-ko.readthedocs.io/ko/v0.4.3/

장 많이 사용되는 방법 중 하나이다. <Figure 4>는 고장내역에 대한 사례베이스를 출력한 결과이다. 이 두가지 형태의 사례베이스는 향후 새로운 고장에 대한 유사 사례 검색을 위해 유사도 계산이 용이하도록 구축하였으며, 자세한 방법은 3.5절에서 설명한다.

3.5 고장내역 간 유사도 측정방안

본 연구에서는 사례베이스로부터 유사 사례를 검색하기 위해 사례들을 벡터화하고 고장의 특성을 추출하여 코사인 거리를 기반으로 유사도를 측정한다. 이를 위한 새로운 고장 사례와 사례베이스를 변환하는 알고리즘에 대하여 기술하고자 한다. 유사도 측정 방안은 5가지로 구분되며, 각각의 알고리즘과 차원 축소 알고리즘의 경우 차원 수의 변화가 성능에 통계적으로 유의한 차이를 보이는지에 대하여 검증하고자 한다. 본 연구의 알고리즘을 구현하기 위하여 Python 3.7 버전과 Sklearn 0.20.3[2] 버전, Gensim 3.4.0[3] 버전의 패키지를 활용하였다.

3.5.1 랜덤 제안

고장코드는 3바이트를 사용하여 WW-XX-YY 형태의 6자리형식으로 표현되고, 각각의 바이트를 통해 고장이 속하는 유형 및 열차 기능의 주요 고장을 나타낼 수 있다. 예를 들어, WW 바이트에 C로 시작하며 C1~C8의 코드범위의 코드가 부여되었다면 이 열차는 MF라는 고장의 유형을 나타낸다. MF는 주요고장으로 열차가 더 이상 운행을 할 수 없을 때 기록이 되며, 열차는 상업운전을 끝내고 유지보수를 해야 한다는 것을 의미한다. 따라서 장애가 발생 시 차상 컴퓨터 시스템에 나타나는 고장코드는 차량 고장을 특정하는 진단의 중요한 시작점이라 할 수 있다. 하나의 고장코드에 대해 여러가지 진단과 원인들이 나타날 수 있기 때문에 하나의 고장과 하나의 진단을 매칭할 수는 없지만 본 연구에서는 가장 기본적인 방법으로써 동일한 장애코드를 가지는 사례를 검색하여 그 중 임의로 3가지를 가이드로 제시하는 알고리즘을 구축하였다. 다음은 랜덤 제안 방식의 알고리즘 절차를 나타낸 것이며, <Figure 5>는 랜덤 방식의 예를 도식화하여 나타낸 것이다.

랜덤 제안 방식 알고리즘 절차:

(Step 1) 새로운 사례의 장애코드를 장애코드 변수에서 검색한다.

(Step 2) 동일한 장애코드를 가지는 사례를 추출하고 계통과 발생 구분 변수를 통해 동일한 계통과 발생 위치를 필터링한다.

(Step 3) 필터링된 유사한 고장 사례들 중 임의로 3가지를 선택하여 사례에 대한 인덱스를 추출한다.

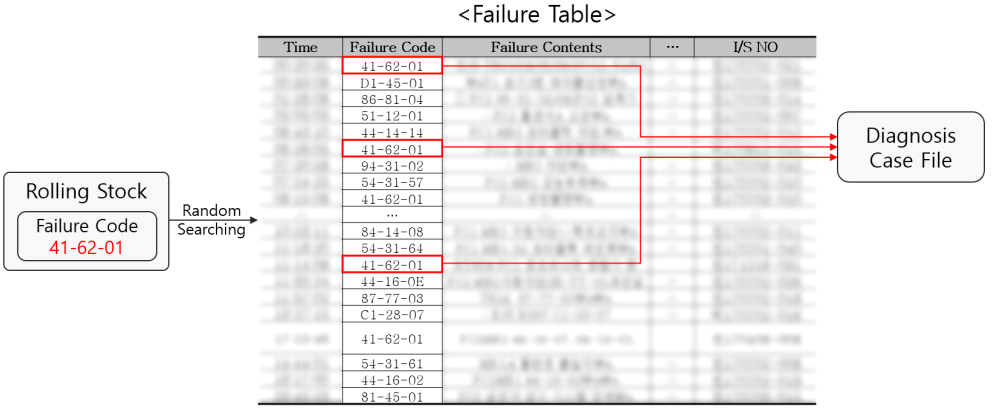
(Step 4) 사례인덱스에 해당하는 조치내역을 검색하여 가이드로 제시한다.

3.5.2 코사인 유사도

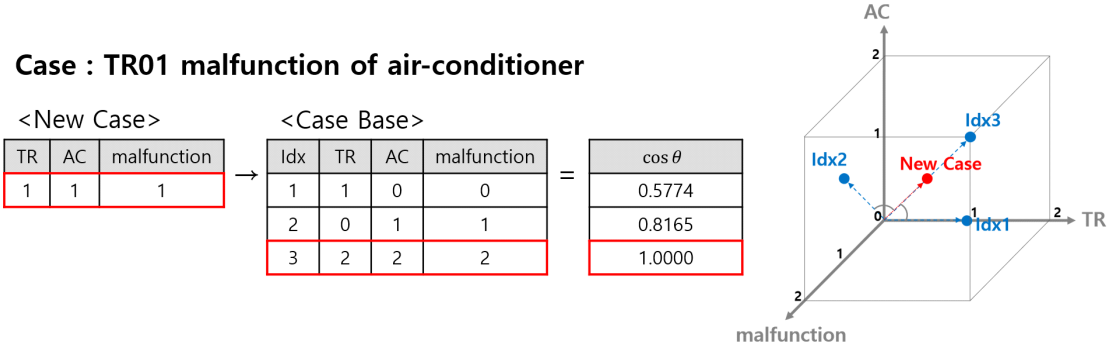
본 연구에서는 새로운 고장 사례와 사례베이스 간의 유사도를 코사인 거리로 계산하여 가장 유사도 값이 큰 3가지를 가이드로 제시하는 알고리즘을 구축하였다. 다음은 코사인 유사도 방식의 알고리즘 절차를 나타낸 것이며, <Figure 6>은 코사인 유사도 방식을 구체화한 예이다.

2) https://scikit-learn.org/

3) https://radimrehurek.com/gensim/



〈Figure 5〉 Example of Random Method



〈Figure 6〉 Example of Cosine Similarity Method

코사인 유사도 방식 알고리즘 절차:

(Step 1) 새로운 사례의 고장내역을 통합사전 기반의 단어-문서 행렬로 변환한다.

(Step 2) 다음 식 (4)를 이용하여 변환된 사례 행렬과 전체 사례베이스 행렬의 코사인 유사도를 각각 계산하여 유사도 값과 사례인덱스를 추출한다.

$$\text{Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (4)$$

(Step 3) 코사인 유사도 값이 가장 높은 3가지의 사례인덱스를 추출하여 해당 사례의 유사도 값과 조치내역을 가이드로 제시한다.

3.5.3 차원 축소

고장내역을 통합 사전 기반의 단어-문서 행렬로 변환하는 경우 차원의 수가 사전의 단어 수인 9,819차원으로 만들어지며 대부분의 차원이 0인 희소(sparse) 행렬이 되기 때문에 데이터의 특성을 파악하기 어려워진다. 따라서 본 연구에서는

고장의 특징을 유지하면서 성능을 향상시킬 수 있는 방안으로 차원 축소 방법을 적용하여 유사 사례를 검색하는 알고리즘을 구축하였다. 차원 축소 방법 중 피처 추출(Feature Extraction) 방식을 적용한 다음의 세가지 알고리즘을 적용하였고 상세한 결과는 4절에서 설명한다.

(1) 비음수 행렬 인수분해(Non-negative Matrix Factorization, NMF)

본 연구에서는 역문서빈도(Term Frequency-Inverse Document Frequency, TF-IDF)와 비음수 행렬 인수분해 기법을 통해 사례베이스를 가중치 행렬과 특성 행렬로 분해하여 사례를 새로운 특성들의 차원으로 축소시킨 후 유사도를 계산한다. 다음은 비음수 행렬 인수분해 방식의 알고리즘 절차를 나타낸 것이며, <Figure 7>은 역문서빈도 행렬을 가중치 행렬과 특성행렬로 분해한 예를 단순화하여 나타낸 것이다.

비음수 행렬 인수분해 방식 알고리즘 절차:

(Step 1) 단어-문서 행렬로 변환된 사례베이스를 이용하여 역문서빈도를 계산한다.

(Step 2) 사례베이스의 역문서빈도 행렬을 가중치 행렬(W)과 특성 행렬(H)로 분해한다.

(Step 3) 사례별로 특징의 변화가 클 수 있으므로 특징값을 정규화 한다.

(Step 4) Step1~3의 절차를 파이프라인으로 만들어 새로운 사례를 동일한 형태로 변환한다.

(Step 5) 새로운 사례와 사례베이스의 특성행렬을 곱하여 코사인 유사도 값을 구한다. (두 사례의 특성행렬은 정규화를 통해 행렬의 크기가 1이므로 코사인 유사도는 두 특성행렬의 곱으로 구할 수 있다.)

(Step 6) 유사도 값이 가장 높은 3가지의 사례 인덱스를 추출하여 해당 사례의 유사도 값과 조치내역을 가이드로 제시한다.

(2) 잠재의미분석(Latent Sematic Analysis, LSA)

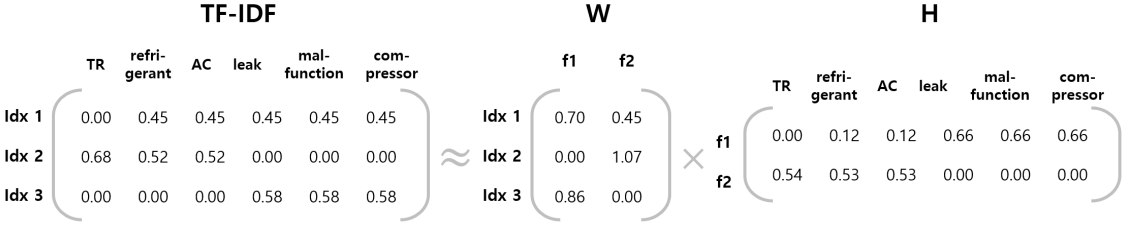
본 연구에서는 특이값분해를 수행해 차원을 축소하고 주요 키워드와 고장 간의 내재적 의미를 보존하는 방식인 잠재의미분석을 적용해 유사도를 계산한다. 다음은 잠재의미분석 방식의 알고리즘 절차를 나타낸 것이다.

잠재의미분석 방식 알고리즘 절차:

(Step 1) 단어-문서 행렬로 변환된 사례베이스를 이용하여 특이값분해의 변형 중 하나인 TruncatedSVD를 적용하여 차원을 축소한다.

(Step 2) 잠재의미분석으로 만들어진 좌표는 문서의 길이에 영향을 받기 때문에 이 영향을 제거하기 위해 문서의 원점에서 거리를 0~1사이 값으로 정규화 한다.

(Step 3) Step1~2의 절차를 파이프라인으로 만들어 새로운 사례를 동일한 형태로 변환한다.



〈Figure 7〉 Example of NMF Method

(Step 4) 새로운 사례와 사례베이스의 행렬을 곱하여 코사인 유사도 값을 구한다.
(Step 5) 유사도 값이 가장 높은 3가지의 사례 인덱스를 추출하여 해당 사례의 유사도 값과 조치내역을 가이드로 제시한다.

(3) Doc2Vec

본 연구에서는 고장에서 발생하는 키워드의 의미를 고려하여 이 키워드들을 벡터로 표현하는 단어 임베딩을 문서 단위로 확장하는 Doc2Vec 기법을 적용하여 유사도를 계산한다. 하나의 고장내역을 하나의 단어처럼 학습시켜 원하는 차원으로 하나의 고장을 벡터 공간에 나타낼 수 있어 위의 두 알고리즘과 동일한 조건으로 성능을 비교하였다. 다음은 Doc2Vec 방식의 알고리즘 절차를 나타낸 것이며, <Figure 8>은 학습된 Doc2Vec모델을 문서에 적용한 예이다.

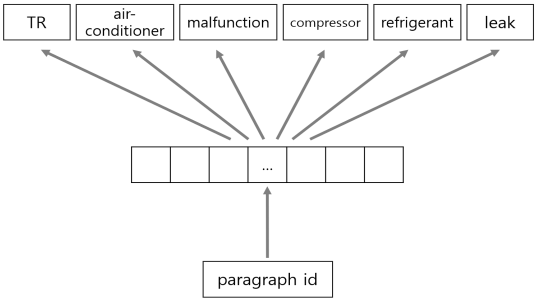
Doc2Vec 방식 알고리즘 절차:
(Step 1) 튜플 형태로 변환된 사례베이스를 이용하여 Doc2Vec을 학습시킨다. (벡터화를 위한 신경망 학습과정으로 Gensim패키지 내 Doc2Vec의 특정 파라미터는 다음과 같이 고정하였고 나머지 주요 파라미터는 최적화가 필요하다.)

$dm = 0(\text{PM} - \text{DBOW 훈련 알고리즘})$,
$vector\_size = (40, 80, 120)$
$min\_count = 1$(통합 사전에 내의 최대한 많은 단어가 포함되도록 설정)

(Step 2) 학습된 모델을 저장한다. (새로운 사례를 사례베이스에 추가 시 기존에 저장된 학습 모델을 사용할 수 있다.)
(Step 3) 새로운 사례를 사례베이스와 동일하게 튜플 형태로 변환한다.
(Step 4) 새로운 사례와 사례베이스의 코사인 유사도 값을 구한다. (본 논문에서는 성능 비교를 위해 모든 알고리즘에 코사인 거리를 적용하였다.)
(Step 5) 유사도 값이 가장 높은 3가지의 사례 인덱스를 추출하여 해당 사례의 유사도 값과 조치내역을 가이드로 제시한다.

3.6 조치내역 활용 및 성능 평가 방법

조치지원시스템은 사용자에게 총 3가지의 조치가이드 목록을 전달한다. 가장 유사도가 높은 상위 3개의 고장 사례들을 뽑아 이 고장들을 실제로 어떻게 조치하였는지에 대한 내역을 제안



Input: taggedDoc(words=[‘TR’, ‘air-conditioner’, ‘malfunction’, ‘compressor’, ‘refrigerant’,‘leak’], idx=‘1’)

Output:

[-0.7396, 0.0298, -0.1149, -0.2411, -0.0544, 0.2425, -0.0961, 0.1621, 0.0177, 0.1293, 0.2505, -0.1925, -0.2204, 0.3159, 0.3414, -0.3155, -0.1018, -0.0048, 0.2643, -0.3555, -0.5033, 0.1050, 0.3093, -0.2113, 0.2757, 0.0218, 0.1775, -0.1413, -0.1056, 0.5001, -0.2668, 0.4704, -0.7359, 0.0253, -0.2362, 0.0082, -0.2635, 0.4616, 0.0092, 0.3093]

〈Figure 8〉 Example of Doc2Vec Method

한다. 작업자가 3가지 조치내역들을 통해 기존의 진단과 조치를 참고하여 새로운 고장에 대한 조치 방향과 계획을 세우도록 지원한다.

본 연구에서는 제안된 조치가 얼마나 실제 조치에 가까운지를 평가하기 위하여 정확도와 재현도, F-measure 방식을 사용하였다. 재현도와 정확도는 할당된 코드나 키워드의 적절성을 결정하는데 유용한 척도이며, 시스템의 전체적인 성능을 평가하기위해 F-measure도 같이 측정하였다. 정확도는 CBR을 통해 할당한 키워드 중 몇 개가 정확했는가에 대한 비율이며, 재현도는 실제 키워드 중 CBR이 정확히 할당한 키워드가 몇 개인지에 대한 비율이다. 성능 평가용 데이터 집합의 실제 조치내역과 사례베이스에서 제안된 조치내역 3개의 키워드를 각각 추출하여 다음의 식 (5), (6), (7)을 통해 정확도, 재현도, F-measure를 측정하고 산술평균 하였다. 성능을 분석한 결과는 4절에서 설명한다.

$$\text{Precision} = \frac{the\ correct\ keywords\ assigned\ by\ CBR}{the\ keywords\ assigned\ by\ CBR} \quad (5)$$

$$\text{Recall} = \frac{the\ correct\ keywords\ assigned\ by\ CBR}{the\ correct\ keywords} \quad (6)$$

$$\text{F} - \text{measure} = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (7)$$

4. 실험 결과 및 검증

본 연구에서는 제안된 방법들의 성능을 비교하기 위하여 수집된 고장데이터 82,405건을 훈련집합 65,924건(80%)과 성능 평가용 데이터 집합 16,481건(20%)으로 나누어 활용하였고, 차원 축소 방식을 적용하는 알고리즘의 경우 차원의 수를 모두 동일하게 40개, 80개, 120개로 선정하였다. 알고리즘의 통계적 유의성을 검증하기 위해 전체 과정을 20번 반복수행 하였으며, 각 과정에서의 데이터 분할과 차원축소에 필요한 모든 Seed 값은 1-100까지 숫자 중 20개를 임의 추출하여 그 값을 부여함으로써 동일한 조건 하에 실험을 수행하였다.

본 실험의 첫 번째 목적은 제안한 5개 알고리즘 간 성능의 차이가 유의미하게 존재하는지를 검증하는 것이다. 그리고 두 번째 목적은 제안한 알고리즘들을 활용할 때 텍스트의 벡터 변환이 필요한데 차원 축소 방식을 적용한 벡터 변환 시 차원 수에 따른 성능 변화를 실증적으로 확인하는 것이다.

4.1 실험결과

4.1.1 조치지원시스템 제안 성능 분석 결과

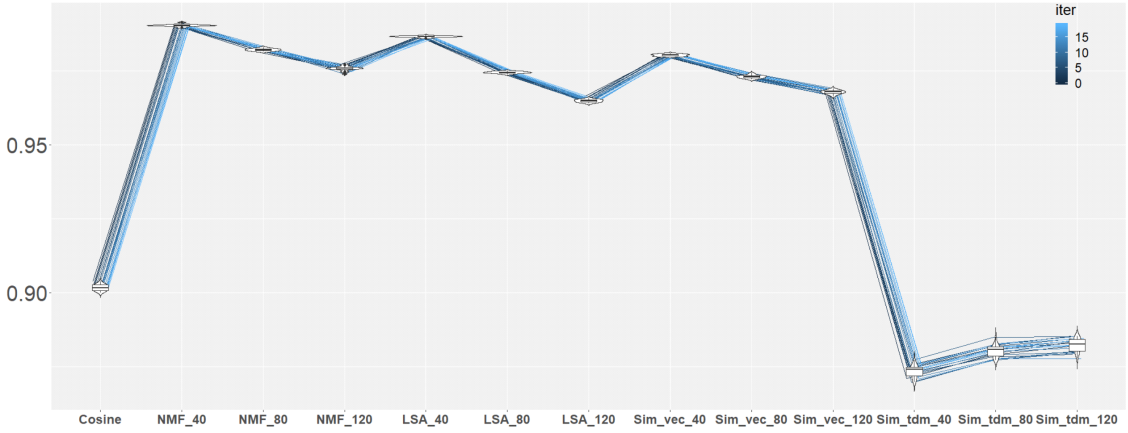
성능 평가용 데이터 집합 내 16,481건을 새롭게 발생한 고장 사례로 하여 사례베이스 간의 유사도를 계산해 가장 유사도 값이 큰 3가지를 추출하고, 이 3가지 각각의 조치내역에 대해 정확도, 재현도, F-measure를 측정한 뒤 이 값들을 산술평균한 값을 1건의 평균 유사도, 평균 정확도, 평균 재현도, 평균 F-measure로 산출하였다. 이렇게 산출된 성능 평가용 데이터 집합 전체 값을 다시 산술평균한 값을 1회(iteration)의 평균 유사도, 평균 정확도, 평균 재현도, 평균 F-measure로 산출하였고, 이 과정을 20번 반복 수행한 결과는 다음의 <Table 2>와 같다. 랜덤 알고리즘의 경우 유사도를 계산하지 않기 때문에 <Table 2>에서 제외하였고, 추가적으로 Doc2Vec의 경우 사례

〈Table 2〉 Experiment Result for Model

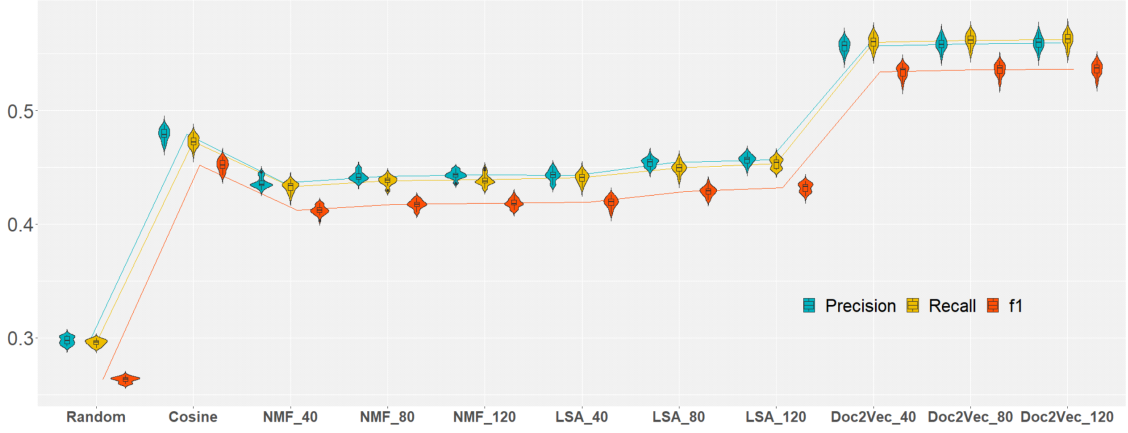
Test Result	Model			Mean	S.D.	Min	Max
Similarity	Cosine			0.9018	0.0011	0.9000	0.9038
	NMF		40	0.9904	0.0004	0.9896	0.9915
			80	0.9822	0.0004	0.9815	0.9830
			120	0.9758	0.0008	0.9740	0.9771
	LSA		40	0.9866	0.0003	0.9861	0.9871
			80	0.9745	0.0003	0.9738	0.9750
			120	0.9649	0.0005	0.9640	0.9658
	Doc2Vec	Sim_vec	40	0.9804	0.0004	0.9799	0.9809
			80	0.9731	0.0005	0.9722	0.9742
			120	0.9678	0.0007	0.9664	0.9690
		Sim_tdm	40	0.8735	0.0021	0.8698	0.8772
			80	0.8805	0.0021	0.8772	0.8851
			120	0.8825	0.0023	0.8777	0.8854
Precision	Random			0.2976	0.0034	0.2921	0.3022
	Cosine			0.4791	0.0053	0.4685	0.4873
	NMF		40	0.4365	0.0045	0.4297	0.4459
			80	0.4421	0.0039	0.4360	0.4500
			120	0.4433	0.0035	0.4359	0.4492
	LSA		40	0.4428	0.0048	0.4340	0.4507
			80	0.4541	0.0037	0.4470	0.4615
			120	0.4568	0.0039	0.4481	0.4634
	Doc2Vec		40	0.5566	0.0052	0.5455	0.5651
			80	0.5586	0.0055	0.5469	0.5686
			120	0.5594	0.0056	0.5481	0.5695
Recall	Random			0.2957	0.0027	0.2907	0.2996
	Cosine			0.4727	0.0049	0.4620	0.4810
	NMF		40	0.4328	0.0042	0.4228	0.4395
			80	0.4381	0.0039	0.4297	0.4447
			120	0.4389	0.0049	0.4311	0.4495
	LSA		40	0.4408	0.0045	0.4311	0.4486
			80	0.4494	0.0048	0.4386	0.4580
			120	0.4534	0.0044	0.4476	0.4600
	Doc2Vec		40	0.5602	0.0056	0.5495	0.5697
			80	0.5620	0.0056	0.5505	0.5707
			120	0.5623	0.0058	0.5506	0.5721
F-measure	Random			0.2631	0.0024	0.2592	0.2668
	Cosine			0.4519	0.0050	0.4431	0.4595
	NMF		40	0.4122	0.0039	0.4034	0.4185
			80	0.4174	0.0036	0.4104	0.4231
			120	0.4184	0.0038	0.4120	0.4259
	LSA		40	0.4192	0.0045	0.4091	0.4257
			80	0.4288	0.0041	0.4210	0.4373
			120	0.4320	0.0040	0.4245	0.4378
	Doc2Vec		40	0.5340	0.0052	0.5221	0.5415
			80	0.5357	0.0053	0.5241	0.5435
			120	0.5362	0.0055	0.5246	0.5445

문장의 의미가 보존된 벡터 값으로 유사도를 계산한 코사인 거리(Sim_vec)와 Doc2Vec으로 추출된 유사 사례를 단어의 의미를 고려하지 않은 단어-문서 행렬 상태로 유사도를 계산하였을 때의 코사인 거리(Sim_tdm)를 비교하여 Doc2Vec의 효과성을 살펴보았다.

5가지의 알고리즘을 적용해 유사 사례를 검색하고 조치를 제안하는 성능을 분석한 결과는 다음과 같다. 차원 축소의 관점에서 본다면, <Figure 9>에서 보는 것처럼 NMF, LSA, Doc2Vec 모두에서 차원이 커질수록(40 → 120) 유사도 값은 작아지는 경향을 보였지만 정확도, 재현도, F-measure의 값은 오히려 커지는 경향이 있었다. 즉, 알고리즘 별로 차원이 40개로 축소되었을 때 고장사례 간의 유사도 값은 가장 크게 측정이 되지만, 차원이 120개로 축소되었을 때가 가장 실제 조치에 유용한 결과를 제공해주었다. 특히, 코사인 유사도를 그대로 적용하는 경우에서 이 현상은



(a) Trend of Similarity



(b) Trend and Box-Plot of Precision, Recall and F-measure

〈Figure 9〉 Trend and Box-Plot for Comparison to Experiment Results

두드러지게 나타났다. 전체적으로 NMF를 40개의 특성으로 축소한 경우의 유사도가 0.9904로 코사인 유사도 알고리즘보다 0.0886만큼 크게 측정되었으나, 오히려 조치를 제안해주는 측면에서의 성능은 코사인 유사도를 바로 적용한 알고리즘이 F-measure를 기준으로 0.0397만큼 높았다.

요약하자면, 첫번째로 새로운 고장을 사례베이스에서 검색할 때는 고장 사례 간의 코사인 거리로 측정하였는데 이 값은 차원의 수를 작게 할수록 높게 나타나는 경향이 있었고, 코사인 유사도 알고리즘을 직접 적용하는 것보다는 차원 축소 방법을 적용하는 경우가 더 높게 나타났다. 두번째로 유사한 고장으로 검색된 사례의 조치가 실제로 유용한지는 정확도, 재현도, F-measure로 측정하였는데 이 값은 반대로 차원의 수를 크게 할수록 높게 나타나는 경향이 있었고, 차원을 축소하는 NMF, LSA 방식보다 직접 코사인 유사도를 적용하는 알고리즘이 더 좋은 성능을 보였다. 이는 차원 축소의 장단점이 모두 나타난 것으로 보인다. 차원의 수를 너무 작게 설정하게 되면 특성이 너무 크게 분류되어 특성의 의미가 사라져 다른 고장임에도 유사한 것으로 판별하기 때문에 첫번째 경우처럼 차원의 수가 적을수록 코사인 거리값이 크게 나온 것으로 보이며, 차원을 적절하게 설정하는 경우 특성을 잘 보존하기 때문에 차원 축소 방법이 일반적으로 직접 코사인 유사도 알고리즘을 적용하는 것보다는 좋다. 두번째 경우처럼 차원을 축소하는 두 가지 방식이 코사인 유사도 알고리즘보다 성능이 낮은 원인 중 하나는 고장 내역에서 추출되는 키워드 수와 용어의 특수성에서 기인하는 것으로 판단된다. 고장내역의 경우 고장의 상태를 장애코드-키워드, 키워드-키워드의 간단한 조합으로 표현하기 때문에 추출되는 키워드의 수가 5개 이하인 사례가 전체의 58%에 달하는데다 특수한 용어들이 많이 포함되어 있어 하나의 단어가 중복적인 의미로 사용되는 경우가 많지 않아 단어-문서 행렬에 코사인 유사도를 직접 적용하는 것만으로도 충분한 효과가 있었던 것으로 보인다. 그럼에도 문장 전체의 의미를 고려하면서도 좀 더 조밀한 차원의 벡터로 표현하는 Doc2Vec의 경우 유사 문서 검색과 조치 가이드 제공 성능 두가지 모두에서 매우 우수한 성능을 보여주었다. 이를 검증하기 위해 Doc2Vec에서 검색된 유사 문서의 인덱스를 단어-문서 행렬에서 찾아 코사인 거리를 측정해본 결과 코사인 거리값이 F-measure를 기준으로 0.0853만큼 높은 것을 확인할 수 있었다. 다시 말해서, Sim_tdm으로 측정된 값은 코사인 유사도 알고리즘의 계산법과 동일한데도 불구하고 성능 측정에서 차이가 났다는 것은 Doc2Vec이 유사 사례로 검색한 인덱스와 코사인 유사도 알고리즘이 서로 다른 인덱스를 추출했다는 의미이며, 고장의 특성을 잘 보존하여 실제 조치에 더 유용한 결과를 검색해 냈다는 것을 의미한다. 결론적으로 알고리즘 선정 시 차원의 수를 적정하게 선정하는 것과 문서의 특징을 고려하는 것이 성능에 큰 영향을 미친다는 것을 알 수 있었다.

4.2 성능 차이에 대한 통계적 검정

결과값의 편향을 제거하기 위하여 유사 고장을 검색하고 조치의 유용성을 검증하는 과정을 20번 반복 수행하고 통계적 유의성을 검정하였다. 결과는 다음의 <Table 3>, <Table 4>, <Table 5>와 같다. <Table 3>은 각 알고리즘의 성능을 요약한 자료이며, <Table 4>는 알고리즘별로 20회 반복수행을 통해 도출된 평균값의 차이가 있

〈Table 3〉 Summary for Experiment Results

Model		Dimension	Similarity	Precision	Recall	F-measure
Random			-	0.2976	0.2957	0.2631
Cosine			0.9018	0.4791	0.4727	0.4519
NMF		40	0.9904	0.4365	0.4328	0.4122
		80	0.9822	0.4421	0.4381	0.4174
		120	0.9758	0.4433	0.4389	0.4184
LSA		40	0.9866	0.4428	0.4408	0.4192
		80	0.9745	0.4541	0.4494	0.4288
		120	0.9649	0.4568	0.4534	0.4320
Doc2Vec	Sim_vec	40	0.9804	0.5566	0.5602	0.5340
		80	0.9731	0.5586	0.5620	0.5357
		120	0.9678	0.5594	0.5623	0.5362
	Sim_tdm	40	0.8735	-	-	-
		80	0.8805			
		120	0.8825			

〈Table 4〉 ANOVA Test for Performance Difference Analysis

		Similarity	Precision	Recall	F-measure
Model	F	29827.003	5434.380	5434.380	6429.956
	p-value	<2e-16	<2e-16	<2e-16	<2e-16
iter	F	0.135	0.660	0.660	1.253
	p-value	0.713	0.417	0.417	0.264
Model:iter	F	0.677	0.838	0.838	1.221
	p-value	0.773	0.593	0.593	0.279

는지를 검정하기 위하여 분산분석을 수행한 결과이다. 분석결과, 모델 간 평균의 차이가 있는 것으로 나타났다(평균이 모두 같지 않고 적어도 하나는 차이가 발생하였다). 수행 횟수(iter)에 따라서는 평균이 차이를 보이고 있지 않으며, 모델과 수행 횟수 사이의 교호작용도 없는 것으로 분석되었다. <Table 5>에서는 Tukey's HSD test를 통해 모델 간 평균 차이의 유의성을 추가로 검정하였다. 유사도의 경우 모델 간 평균의 차이가 모두 유의하게 나타났고, 정확도, 재현도, F-measure의 경우 모델 내에서 차원의 수에 따른 평균 차이가 유의하지 않은 경우는 NMF의 (NMF80, NMF120), LSA의 (LSA80, LSA120), Doc2Vec의 (Doc2Vec 40, Doc2Vec80), (Doc2Vec40, Doc2Vec120), (Doc2Vec40, Doc2Vec120) 다섯가지 경우이고, 모델 간 차원의 수에 따른 평균 차

〈Table 5〉 Paired T-Test Result for The Models

Groups being compared				Similarity		Precision		Recall		F-measure	
Model A		Model B		A-B	p-value	A-B	p-value	A-B	p-value	A-B	p-value
Random		Cosine		-		-0.1815	0.0000	-0.1770	0.0000	-0.1888	0.0000
		NMF	40	-		-0.1389	0.0000	-0.1371	0.0000	-0.1491	0.0000
			80	-		-0.1445	0.0000	-0.1424	0.0000	-0.1543	0.0000
			120	-		-0.1457	0.0000	-0.1433	0.0000	-0.1553	0.0000
		LSA	40	-		-0.1453	0.0000	-0.1451	0.0000	-0.1561	0.0000
			80	-		-0.1566	0.0000	-0.1537	0.0000	-0.1658	0.0000
			120	-		-0.1592	0.0000	-0.1578	0.0000	-0.1689	0.0000
		Doc2Vec	40	-		-0.2590	0.0000	-0.2646	0.0000	-0.2709	0.0000
			80	-		-0.2610	0.0000	-0.2663	0.0000	-0.2727	0.0000
			120	-		-0.2619	0.0000	-0.2667	0.0000	-0.2731	0.0000
Cosine		NMF	40	0.0886	0.0000	-0.0426	0.0000	-0.0399	0.0000	-0.0396	0.0000
			80	0.0804	0.0000	-0.0370	0.0000	-0.0346	0.0000	-0.0345	0.0000
			120	0.0740	0.0000	-0.0358	0.0000	-0.0338	0.0000	-0.0335	0.0000
		LSA	40	0.0848	0.0000	-0.0362	0.0000	-0.0319	0.0000	-0.0327	0.0000
			80	0.0727	0.0000	-0.0249	0.0000	-0.0233	0.0000	-0.0230	0.0000
			120	0.0631	0.0000	-0.0223	0.0000	-0.0192	0.0000	-0.0199	0.0000
		Doc2Vec	40	0.0786	0.0000	0.0775	0.0000	0.0875	0.0000	0.0821	0.0000
			80	0.0713	0.0000	0.0795	0.0000	0.0893	0.0000	0.0839	0.0000
			120	0.0660	0.0000	0.0804	0.0000	0.0897	0.0000	0.0844	0.0000
NMF	40	NMF	80	-0.0082	0.0000	0.0056	0.0063	0.0053	0.0202	0.0052	0.0102
			120	0.0146	0.0000	-0.0068	0.0002	-0.0061	0.0028	-0.0061	0.0007
		LSA	40	0.0038	0.0000	-0.0064	0.0008	-0.0080	0.0000	-0.0069	0.0001
			80	0.0159	0.0000	-0.0177	0.0000	-0.0166	0.0000	-0.0166	0.0000
			120	0.0255	0.0000	-0.0204	0.0000	-0.0206	0.0000	-0.0197	0.0000
		Doc2Vec	40	0.0100	0.0000	-0.1202	0.0000	-0.1274	0.0000	-0.1217	0.0000
			80	0.0173	0.0000	-0.1222	0.0000	-0.1292	0.0000	-0.1235	0.0000
			120	0.0225	0.0000	-0.1230	0.0000	-0.1295	0.0000	-0.1240	0.0000
NMF	80	NMF	120	0.0063	0.0000	-0.0012	0.9990*	-0.0008	1.0000*	-0.0010	0.9998*
		LSA	40	-0.0045	0.0000	-0.0008	1.0000*	-0.0027	0.7639*	-0.0017	0.9731*
			80	0.0077	0.0000	-0.0121	0.0000	-0.0113	0.0000	-0.0114	0.0000
			120	0.0173	0.0000	-0.0147	0.0000	-0.0153	0.0000	-0.0146	0.0000
		Doc2Vec	40	0.0018	0.0002	-0.1145	0.0000	-0.1221	0.0000	-0.1166	0.0000
			80	0.0091	0.0000	-0.1165	0.0000	-0.1239	0.0000	-0.1183	0.0000
			120	0.0143	0.0000	-0.1174	0.0000	-0.1243	0.0000	-0.1188	0.0000
NMF	120	LSA	40	-0.0108	0.0000	0.0004	1.0000*	-0.0019	0.9732*	-0.0008	1.0000*
			80	0.0013	0.0272	-0.0109	0.0000	-0.0105	0.0000	-0.0105	0.0000
			120	0.0109	0.0000	-0.0135	0.0000	-0.0145	0.0000	-0.0136	0.0000
		Doc2Vec	40	-0.0045	0.0000	-0.1133	0.0000	-0.1213	0.0000	-0.1156	0.0000
			80	0.0028	0.0000	-0.1153	0.0000	-0.1231	0.0000	-0.1174	0.0000
			120	0.0080	0.0000	-0.1162	0.0000	-0.1234	0.0000	-0.1179	0.0000
LSA	40	LSA	80	-0.0121	0.0000	0.0113	0.0000	0.0086	0.0000	0.0097	0.0000
			120	0.0217	0.0000	-0.0140	0.0000	-0.0126	0.0000	-0.0128	0.0000
		Doc2Vec	40	0.0063	0.0000	-0.1138	0.0000	-0.1194	0.0000	-0.1148	0.0000
			80	0.0136	0.0000	-0.1158	0.0000	-0.1212	0.0000	-0.1166	0.0000
			120	0.0188	0.0000	-0.1166	0.0000	-0.1215	0.0000	-0.1171	0.0000
LSA	80	LSA	120	0.0096	0.0000	-0.0027	0.7530*	-0.0040	0.2089*	-0.0031	0.4649*
		Doc2Vec	40	-0.0059	0.0000	-0.1025	0.0000	-0.1108	0.0000	-0.1051	0.0000
			80	0.0015	0.0081	-0.1045	0.0000	-0.1126	0.0000	-0.1069	0.0000
			120	0.0067	0.0000	-0.1053	0.0000	-0.1129	0.0000	-0.1074	0.0000
LSA	120	Doc2Vec	40	-0.0155	0.0000	-0.0998	0.0000	-0.1068	0.0000	-0.1020	0.0000
			80	-0.0081	0.0000	-0.1018	0.0000	-0.1086	0.0000	-0.1038	0.0000
			120	-0.0029	0.0000	-0.1026	0.0000	-0.1089	0.0000	-0.1043	0.0000
Doc2Vec	40	Doc2Vec	80	-0.0073	0.0000	0.0020	0.9509*	0.0018	0.9821*	0.0018	0.9689*
			120	0.0125	0.0000	-0.0028	0.6781*	-0.0021	0.9430*	-0.0023	0.8635*
Doc2Vec	80	Doc2Vec	120	0.0052	0.0000	-0.0008	1.0000*	-0.0003	1.0000*	-0.0005	1.0000*

* A p-value higher than 0.05 (> 0.05) is not statistically significant and indicates weak evidence against the null hypothesis.

이가 유의하지 않은 경우는 다음 (NMF80, LSA40), (NMF120, LSA80)의 두 경우로 총 일곱 가지 경우를 제외하고는 평균의 차이가 유의하게 나타났다.

이를 통해 고장 유사 사례의 검색은 고장 코드만 활용하는 것보다 고장 내역의 키워드와 결합하여 사용할 때 더 성능이 우수하였으며, 키워드를 단순히 고차원으로 활용하는 방법보다는 고장의 특징을 보존할 수 있도록 차원을 축소하는 방법을 적용하는 것이 더 우수한 성능을 낸다는 것을 확인할 수 있었다. 또한 차원 축소 기법을 적용할 때 차원의 수는 고장의 특징을 너무 포괄적으로 보존하지 않도록 적절히 크게 하는 것이 성능에 유의미한 차이를 보여준다는 것도 확인할 수 있었다. 기존의 사례기반추론을 활용한 차량고장 연구들의 경우 키워드 매칭 방식을 사용하였으며 이에 대한 유사 사례의 검색과 조치의 제안과 관련한 성능 지표를 확인할 수 없어 본 연구의 방법론들의 성능이 우수하다고 판단하기 어렵다. 하지만 고장 코드를 단순 매칭하여 조치 내용을 검색하는 방안과 비교하였을 때 Doc2Vec의 경우 2배 이상의 성능을 보이고 있으며 기업 내부 연구원에서 자체적으로 진행하였던 인공신경망을 활용한 진단시스템에 관한 연구에서 보였던 진단 성능보다도 최대 20% 향상된 결과를 보여주었다. 현재의 결과를 실무에 바로 적용해서 사용하는 것은 어려울 수 있으나 통계적 유의성 검증을 통해 보고자 하는 바는 적용했던 모델 간 유의한 차이가 있음을 통해 조치를 제안하는 알고리즘 모델 중 성능이 더 좋은 모델을 찾고자 함이며, 실제 시스템에 접목한다면 성능이 가장 좋은 Doc2vec을 기반으로 차원의 수를 최적화하고 의미상 유사 단어의 키워드 표준화를 통해 사례 검색과 조치 내역을 제안할 수 있도록 성능을 개선해야 할 것이다. 키워드 표준화는 실제 적용 시 유사 사례 검색의 성능을 높이는 효과 외에 조치의 제안 측면에서도 성능과 밀접한 관련이 있다. 고장진단시스템에서 조치의 제안과 관련해서는 재현도를 높이는 것이 더 중요할 것으로 판단된다. 실제 조치가 필요한 키워드가 제안에서 도출되지 않는 경우 고장 진단에서는 더 위험할 수 있기 때문이다. 다만, 모든 키워드를 조치로 제안하게 되면 재현도는 최대로 높일 수 있겠으나 현실적인 투입 인원이나 조치 소요시간이 제한적이므로 자원 범위 내에서 최소한의 예측에 대한 정확도를 확보한 후 재현도를 높이도록 해야 할 것이고 향후 사례가 더 축적되고 유사어나 동의어들에 대한 키워드가 표준화된다면 정확도와 재현도를 동시에 높일 수 있을 것으로 보인다.

5. 시사점

5.1 학술적 시사점

본 연구를 통해 비정형 기반의 사례기반추론을 적용할 때 데이터의 효과적인 특징 추출과 비정형 데이터 변환 방안들의 유용성을 실험 결과로 확인할 수 있었다. 차량분야의 경우 업무의 복잡성으로 인해 전문가의 지식과 경험들을 모두 규칙으로 만들기 어렵고, 하나의 고장에 대한 다양한 진단을 일반화하는데도 한계가 있기 때문에 규칙기반 시스템을 구축하는데 어려운 환경임은 분명하다. 실제로 인공 신경망을 활용한 진단시스템과 같은 규칙기반의 연구가 진행된 적이 있었다. KTX 모터블록에 대한 고장과 부품 교환 이력을 학습하여 고장 발생 시 예상되는 부

품을 미리 진단하도록 예측시스템을 개발하였으나, 실제 교환부품과의 비교를 통해 성능을 검증한 결과 진단의 정확도는 34~37%였다. 주요 원인은 하나의 고장에 대한 하나의 조치를 매핑할 수 없어 학습이 잘 되지 않는 문제가 있었고, 다른 하나는 학습을 위한 입력데이터를 만드는 전처리 과정에서 텍스트 구조의 정형화가 어렵다는 것이었다. 반면, 사례기반추론을 통한 고장진단의 경우 학습 대상을 부품교환 이력을 포함한 고장 전체로 확장한 상태에서도 Doc2Vec의 경우 정확도를 54%까지 올릴 수 있었다. 사용 데이터와 대상의 차이 등을 감안할 때 직접적인 비교는 어려우나 사례기반추론의 모형 중 가장 단순한 결합함수와 거리함수를 적용한 코사인 유사도 모형도 F-measure를 기준으로 45% 정도를 낼 수 있었다. 또한 테이블의 모든 속성이 비정형이면서 전문적인 용어의 결합으로 구성되어 있는 데이터에 적용이 가능한 용어사전 구축 방법과 문장의 특징을 추출하는 기법들에 대한 자동화된 방법을 연구하여 새로운 사례를 사례베이스의 학습데이터로 구축하는데 이 방법론을 적용하였다. 용어사전 구축의 경우 일반적인 문장의 형태가 아닌 고장코드-발생위치-단어의 조합으로 구성된 형태가 많아 형태소 분석기를 직접 적용하면 고장코드 추출이 어려워 두 번의 전처리 과정과 한 번의 후처리 과정을 거쳐 고장내역에서 고장코드만을 추출하는 방법을 연구하여 구축하였다. 이러한 자동화된 방법론의 적용을 통해 기존 연구들에서 단순히 고장의 키워드 일치여부로만 고장 내용을 검색하였을 때 나타날 수 있는 한계점을 보완할 수 있었다. 동일한 고장 코드를 검색하는 알고리즘의 경우 정확한 조치 방안을 제안하는 성능이 26%정도였던 것을 감안하면 동일 고장이라도 조치 방법은 코드-발생위치-단어의 조합에 따라 달라질 수 있을 것이고 해당 키워드와의 일치 여부로만 고장 내역을 검색하게 되면 의미적 관계가 무시되어 잘못된 조치 방안이 확인될 가능성이 높다고 볼 수 있다. 이에 본 연구에서는 이를 보완하기 위한 방법으로 고장 내역의 텍스트와 함께 주변 정보(코드, 발생위치)를 함께 결합하였고, 이러한 결합정보에서 특징을 추출하는 방안과 유사도를 산출하는 다양한 방안들의 유용성을 실험을 통하여 확인하였다. 전통적인 텍스트 마이닝 기법에서 활용되었던 단어기반의 Bag-of-words 방식보다 의미의 유사성을 파악할 수 있는 최신의 Doc2Vec을 적용함으로써 차원 축소 방법의 유용성을 사례적으로 보였다는데 의의가 있다. 현재의 결과로는 사례기반 시스템이 현실에 더 유용할 것으로 보이나 진단시스템에서 연구되고 있는 규칙기반 시스템의 유용성을 하이브리드 방식으로 접목하는 방안도 고려해볼 필요가 있을 것이다.

5.2 실무적 시사점

본 연구는 실제 고장데이터를 기반으로 하여 실용화 관점에서 접근하였기 때문에 현재 유지보수 프로그램의 단계적 개선에 유의미한 결과를 제공할 수 있을 것이다. 현재는 새로운 고장을 과거 데이터에서 성능 평가용 데이터 집합으로 대체하여 실험하였지만, 개통이후부터 축적된 방대한 데이터를 사례베이스로 구축하고 고장 내역을 입력할 수 있도록 시스템을 개선한다면 좀 더 정확한 진단 가이드를 제공할 수 있을 것으로 사료된다. 성능 검증을 위해 3가지 가이드만 제시하였으나, 유사도 값을 기준으로 특정 임계치 이상의 점검 내역과 조치 내역 전체를 제

공한다면 숙련자의 경우 고장을 진단하기 위한 점검 방향을 자신만의 방식과 비교하여 검증하는 용도로 사용할 수 있을 것이고, 비 숙련자의 경우에는 진단 방향을 만들기 위한 학습 가이드로 활용할 수 있을 것이다. 다시 말해서 실제 조치 기반의 best practice를 제공해줌으로써 유지보수 시 진단의 신뢰성을 향상시켜준다. 또한 운행 중 장애 발생 시 유사 고장에 대한 사례 검색을 통해 조치에 필요한 부품을 사전에 준비할 수 있도록 한다면 신속한 유지보수가 가능해질 것이다. 이러한 유지보수 시스템의 개선과 더불어 주기적으로 유지보수 전문가들이 모여 유사한 고장 사례에 대한 적절한 조치 내역들을 선별하여 작업 매뉴얼을 개선하거나 사례집을 만들어 교육에 활용하는 방안도 지식의 전수 측면에서 필요해 보인다.

6. 결론

KTX는 20량의 차량을 1개 편성으로 운행하며, 피크 시간에는 한 번에 1,000명 이상의 이용객을 실어 나르는 대량수송 시스템이다. 운행 중 장애는 지연을 발생시키고 후속열차까지 연쇄적으로 영향을 끼치기 때문에 회사는 물론 고객에게 막대한 금전적, 시간적 손실을 발생시킨다. 따라서 차량 고장발생에 대한 점검대상 및 조치에 대한 정보제공을 통해 신속한 의사결정을 내릴 수 있도록 해주는 것은 매우 중요하다. 본 연구는 이러한 관점에서 현재 고속차량 유지보수 체계와 기존의 전문가시스템들이 가지고 있는 기능을 보완할 수 있는 지능형 유지보수 지원시스템을 제안하였고, 이 시스템의 성능을 검증하기 위해 실험을 수행하였다. 코사인 유사도를 기본으로 차원 축소 방법들을 적용하여 유사한 사례의 진단 내역을 가이드로 제시하였다. 20번의 반복적인 실험을 통해 분산분석을 수행하고 평가한 결과, 전체적으로 Doc2Vec이 유사한 고장을 잘 검색하여 그에 대한 조치 가이드를 가장 잘 제안해주는 것으로 나타났다. 제시된 방안은 사례기반추론과 텍스트 마이닝 기법을 적용하여 고장 키워드의 의미까지 고려한 가장 유사한 사례를 검색할 수 있도록 하였고, 유지보수 전문가들이 보유한 경험과 노하우로 해결한 고장진단 이력을 조치가이드로 제시해줌으로써 고장매뉴얼의 불완전성을 보완해 유지보수의 효율성과 안전성을 높이는데 기여할 수 있을 것으로 기대한다.

본 연구에서 유사사례 검색이라는 관점에서 두가지 한계점이 있다. 첫째, 고장내역의 문장이 짧아 통합사전에 특징적인 키워드만 넣게 되는 경우 사례베이스의 학습데이터가 적어지게 되기 때문에 불가피하게 통합사전의 키워드를 많이 생성함으로써 차원이 불필요하게 커졌다는 점이다. 이는 연산 량을 늘려 대용량 처리 시에는 속도에 영향을 주며, 데이터 탐색 과정도 어렵게 만든다. 둘째, 현재 시스템은 수기로 작성하는 방식을 전자적인 시스템으로만 옮겨 놓았기 때문에 작성자가 기록하는 고장과 조치내역의 텍스트가 그대로 데이터화 된다. 따라서 오타의 가능성이 많고, 동일한 고장이 여러가지 방식으로 표현될 수 있어 현재의 텍스트 기반 방식으로는 다른 용어를 사용한 동일 고장을 검색하는데 어려움이 있다. 타 연구에서는 데이터 전처리 단계에서 서로 동일한 뜻을 의미하는 키워드에 대한 표준화 처리가 필요하다고 하였으며(Kim and Kim, 2014), 도시철도차량의 전문가시스템에서는 유지보수에 쓰이는 용어를 동의어와 유사어

기반으로 등록하고 검색할 수 있도록 시스템을 구축한 사례가 있다(Park, 2012). 이 두가지 한계점은 향후 연구를 통해 고장 키워드들의 동의어들을 만드는 표준화 과정을 통해 의미상 중복되는 단어를 줄임으로써 일부 개선될 수 있을 것으로 보이며, 본질적으로는 고장별로 특정한 분류체계를 가지고 입력될 수 있도록 코드화 하거나, 용어의 표준화와 고장내용의 항목화를 통해 오입력을 줄여 나가는 방식으로 유지보수시스템을 개선해야 할 것이다. 아울러 차원 축소의 다양한 변형들을 적용해 유사한 고장 사례를 검색하는 개선된 방법이나 가이드한 조치내역이 실제로 조치에 도움이 되었는지에 대한 평가를 피드백하여 사례베이스에 반영할 수 있는 방안에 대한 후속연구가 필요할 것이다.

참고문헌(References)

Aamodt, A. and E. Plaza, "Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AI communications*, Vol.7, No.1(1994), 39-59.

Ahn, T. B. and J. T. Park, "Development of Model for Knowledge of Railway Facility Failure Cases," *Journal of The Korean Society for Railway*, Vol.22, No.2(2019), 169-177.

Ahn, T. K. and K. J. Park, "Case-Based Expert System for EMU," *Proceedings of Conference of The Korean Institute of Electrical Engineers*, (2006), 1085-1086.

Choi, S. J. and M. H. Kim, "Case Study on the KTX High Speed Rolling Stock Maintenance Characteristic by Analyzing Failures Statistics for 10 Years," *Proceedings of Conference of The Korean Society for Railway*, (2014), 1297-1302.

Eom, J. K., "The Text-mining using Railway Accident Data," *Journal of The Korean Society for Urban Railway,* Vol.7, No.3 (2019), 397-405.

Heo, G. E. and Y. G. Jung, "Efficient Text Documents Learning using Non-negative Matrix Factorization," *Proceedings of Conference of The Korean Institute of Information Scientists and Engineers*, Vol.36, No.2C (2009), 276-279.

Jeon, S. M., H. W. Suh, and M. G. Jeong, "Automatic Failure Knowledge Extraction from Failure Analysis Documents," *Proceedings of Conference of Society for Computational Design and Engineering*, (2015), 12-22.

Kim, B. J., S. Y. Lee, Y. D. Ahn, and S. J. Kang, "Wind Turbine Blade Fault Diagnosis System Using Machine Learning," *Proceedings of Conference of The Korean Institute of Electrical Engineers*, (2017), 1498-1499.

Kim, D. S. and J. W. Kim, "Research Trend Analysis Using Bibliographic Information and Citations of Cloud Computing Articles: Application of Social Network Analysis," *Journal of Intelligence and Information Systems*, Vol.20, No.1(2014), 195-211.

Le, Q. and T. Mikolov, "Distributed Representations of Sentences and Documents," *Proceedings of International Conference on Machine Learning*, (2014), 1188-1196.

Lee, D. D. and H. S. Seung, "Algorithms for Non-negative Matrix Factorization," *Advances in Neural Information Processing Systems*, (2001), 556-562.

Lee, G. J., B. Y. An, and M. H. Kim, "The Hybrid

of Artificial Neural Networks and Case-based Reasoning for Diagnosis System," *Proceedings of Conference of Korean Institute of Intelligent Systems*, Vol.16, No.1 (2006), 130-133.

Lee, J. S. and H. S. Myoung, "Development of a Book Recommender System for Internet Bookstore using Case-based Reasoning," *Journal of Society for e-Business Studies*, Vol.13, No.4(2008), 173-191.

Lee, J. S. and Y. K. Kim, "A Hybrid Malfunction Diagnostic System Using Rules and Cases," *Journal of Intelligence and Information Systems*, Vol.4, No.1(1998), 115-131.

Lee, W. Y., "Diagnostic Reasoning," *Journal of Communications of the Korean Institute of Information Scientists and Engineers*, Vol.10, No.4(1992), 50-55.

Lee, W. Y., "A study on Fault Diagnosis Methodology [written in Korean]," *Proceedings of Conference of Korean Institute of Industrial Engineers*, (1998), 763-765.

Linoff, G. S. and M. J. Berry, *Data Mining Techniques: for Marketing, Sales, and Customer Relationship Management*. Third Edition, John Wiley & Sons, New Jersey, 2011.

Mikolov, T., K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", *arXiv preprint arXiv:1301.3781*, (2013).

Park, K. H., K. S. Kim, and J. W. Lee, "Efficient Expert System Establish using Text Data of Crop Disease based on Cosine Similarity," *Proceedings of Conference of The Korean Institute of Communications and Information Sciences*, (2018), 312-313.

Park, K. J., "The Development of Case-Based Fault Diagnosis Expert System of Urban Transit Vehicles," *Proceedings of Conference of Korean Society for Precision Engineering*, (2012), 1249-1250.

Park, S. K. and M. C. Shin, "Implementation of Korean Sentence Similarity using Sent2Vec Sentence Embedding," *Proceedings of Conference of Human and Language Technology*, (2018), 541-545.

Park, Y. K., S. B. Park, N. I. Park, and H. A. Lee, "Web News Classification Using Latent Semantic Analysis," *Proceedings of Conference of The Korean Institute of Information Scientists and Engineers*, (2017), 1828-1830.

Song, G. J. and J. J. Lim, "A Study on the Diagnosis and Prediction System of Vehicle Faults Using Condition Based Maintenance Technique," *Journal of The Korea Institute of Intelligent Transport System*, Vol.18, No.4 (2019), 80-95.

Wang, A., "An Industrial Strength Audio Search Algorithm," *Ismir*, Vol.2003, (2003), 7-13.

Yoon, M. H., J. H. Kim, and H. Jin, "Prediction for Performance of KNN in Diagnosis considering Features of Coronary Artery Disease Dataset," *Proceedings of Conference of The Institute of Electronics and Information Engineers*, (2013), 834-838.

Abstract

An Intelligence Support System Research on KTX Rolling Stock Failure Using Case-based Reasoning and Text Mining

Hyung Il Lee* · Jong Woo Kim**

KTX rolling stocks are a system consisting of several machines, electrical devices, and components. The maintenance of the rolling stocks requires considerable expertise and experience of maintenance workers. In the event of a rolling stock failure, the knowledge and experience of the maintainer will result in a difference in the quality of the time and work to solve the problem. So, the resulting availability of the vehicle will vary. Although problem solving is generally based on fault manuals, experienced and skilled professionals can quickly diagnose and take actions by applying personal know-how. Since this knowledge exists in a tacit form, it is difficult to pass it on completely to a successor, and there have been studies that have developed a case-based rolling stock expert system to turn it into a data-driven one. Nonetheless, research on the most commonly used KTX rolling stock on the main-line or the development of a system that extracts text meanings and searches for similar cases is still lacking. Therefore, this study proposes an intelligence supporting system that provides an action guide for emerging failures by using the know-how of these rolling stocks maintenance experts as an example of problem solving.

For this purpose, the case base was constructed by collecting the rolling stocks failure data generated from 2015 to 2017, and the integrated dictionary was constructed separately through the case base to include the essential terminology and failure codes in consideration of the specialty of the railway rolling stock sector. Based on a deployed case base, a new failure was retrieved from past cases and the top three most similar failure cases were extracted to propose the actual actions of these cases as a diagnostic guide. In this study, various dimensionality reduction measures were applied to calculate similarity by taking into account the meaningful relationship of failure details in order to compensate for the limitations of the method of searching cases by keyword matching in rolling stock failure expert system studies using

* Department of Business Informatics, Graduate School, Hanyang University
** Corresponding Author: Jong Woo Kim
School of Business, Hanyang University
222 Wangshimni-ro, Seongdong-gu, Seoul 04763, Korea
Tel: +82-2-2220-1067, Fax: +82-2-2220-1169, E-mail: kjw@hanyang.ac.kr

case-based reasoning in the precedent case-based expert system studies, and their usefulness was verified through experiments. Among the various dimensionality reduction techniques, similar cases were retrieved by applying three algorithms: Non-negative Matrix Factorization(NMF), Latent Semantic Analysis(LSA), and Doc2Vec to extract the characteristics of the failure and measure the cosine distance between the vectors. The precision, recall, and F-measure methods were used to assess the performance of the proposed actions. To compare the performance of dimensionality reduction techniques, the analysis of variance confirmed that the performance differences of the five algorithms were statistically significant, with a comparison between the algorithm that randomly extracts failure cases with identical failure codes and the algorithm that applies cosine similarity directly based on words. In addition, optimal techniques were derived for practical application by verifying differences in performance depending on the number of dimensions for dimensionality reduction. The analysis showed that the performance of the cosine similarity was higher than that of the dimension using Non-negative Matrix Factorization(NMF) and Latent Semantic Analysis(LSA) and the performance of algorithm using Doc2Vec was the highest. Furthermore, in terms of dimensionality reduction techniques, the larger the number of dimensions at the appropriate level, the better the performance was found.

Through this study, we confirmed the usefulness of effective methods of extracting characteristics of data and converting unstructured data when applying case-based reasoning based on which most of the attributes are texted in the special field of KTX rolling stock. Text mining is a trend where studies are being conducted for use in many areas, but studies using such text data are still lacking in an environment where there are a number of specialized terms and limited access to data, such as the one we want to use in this study. In this regard, it is significant that the study first presented an intelligent diagnostic system that suggested action by searching for a case by applying text mining techniques to extract the characteristics of the failure to complement keyword-based case searches. It is expected that this will provide implications as basic study for developing diagnostic systems that can be used immediately on the site.

Key Words : KTX, Fault Diagnosis System, Case-based Reasoning, Rolling Stock Failure, Text Mining

Received : November 14, 2019 Revised : March 10, 2020 Accepted : March 14, 2020
Publication Type : Regular Paper Corresponding Author : Jong Woo Kim

저 자 소 개

이 형 일

현재 한국철도공사에서 AI·빅데이터 추진 부문에 재직 중이다. 뉴저지주립대학교 경영학과에서 학사를 마쳤으며, 국민대학교 빅데이터경영MBA에서 석사학위를 취득하였다. 주요 연구 관심분야는 데이터마이닝, 기계학습과 딥러닝 기법의 비즈니스 활용, 지능형 정보시스템 등이다.

김 종 우

현재 한양대학교 경영대학 경영학부 교수로 재직 중이다. 서울대학교 수학과에서 학사를 마쳤으며, 한국과학기술원에서 경영과학으로 석사학위를, 산업경영학으로 박사학위를 취득하였다. 주요 연구 관심분야는 데이터마이닝 기법과 응용, 기계학습과 딥러닝, 오피니언 마이닝, 상품추천기술, 지능형 정보시스템, 집단지성, 사회 네트워크 분석, 클라우드 컴퓨팅 서비스 등이다.