Journal of The Korea Society of Computer and Information
Vol. 23 No. 7, pp. 19-26, July 2018

www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.07.019

Structuring of Unstructured SNS Messages on Rail Services using Deep Learning Techniques

JinGyu Park*, HwaYeon Kim**, Hyoung-Geun Kim***, Tae-Ki Ahn****, Hyunbean Yi*****

Abstract

This paper presents a structuring process of unstructured social network service (SNS) messages on rail services. We crawl messages about rail services posted on SNS and extract keywords indicating date and time, rail operating company, station name, direction, and rail service types from each message. Among them, the rail service types are classified by machine learning according to predefined rail service types, and the rest are extracted by regular expressions. Words are converted into vector representations using Word2Vec and a conventional Convolutional Neural Network (CNN) is used for training and classification. For performance measurement, our experimental results show a comparison with a TF-IDF and Support Vector Machine (SVM) approach. This structured information in the database and can be easily used for services for railway users.

▸Keyword: Data Structuring, Deep neural network, Information of Rail services, Social network service, Text mining

I. Introduction

철도 이용객은 'Fig. 1'과 같이 연평균 2.61%씩 증가하고 있으며, 각 도시 철도 또한 해마다 증가하고 있다[1]. 최근 교통비용 감축과 저탄소 체계 구축을 위한 철도 시설이 증설되고 있어, 향후 철도 이용객은 계속하여 증가할 것으로 보인다. 시설의 증설로 이용객이 증가하고 있는 반면에 'Fig. 2' 교통안전공단 철도 사고 현황[2]에 따르면 해마다 철도 사고는 감소하고 있다. 2012년에 발생된 사고 발생 수와 2016년을 비교해보면 반 이하로 감소했다. 그 이유로는 철도 환경의 전반적 개선과 철도 종사자에 대한 사고예방 교육 훈련 강화, 대국민 안전 홍보, 안전 시설물 확충 그리고 전반적인 철도 안전 관리의 강화에 기인한다. 여러 해결 방안으로 사고가 줄어들고 있지만 현재까지도 사고가 발생하고 있으며, 사상 사고를 제외한 열차 사고나 건널목 사고는 적게 발생하고 있지만 줄어들지

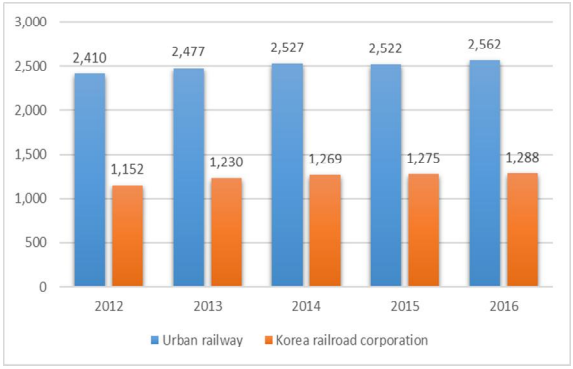

Fig. 1. Number of rail services passengers 2012-2016 (unit : million)

• First Author: JinGyu Park, Corresponding Author: Hyunbean Yi
*JinGyu Park (jgow87@naver.com), Dept. of Computer Engineering, Hanbat National University
**HwaYeon Kim (hwayeon9465@gmail.com), Dept. of Computer Engineering, Hanbat National University
***Hyoung-Geun Kim (khg9792@hanmail.net), Smart R&D Center, U-CORE System Co.
****Tae-Ki Ahn (tkahn@krri.re.kr), Smart Station Research Team, Korea Railroad Research Institute.
*****Hyunbean Yi (bean@hanbat.ac.kr), Dept. of Computer Engineering, Hanbat National University
• Received: 2018. 06. 22, Revised: 2018. 07. 02, Accepted: 2018. 07. 12.
• This research was supported by a grant (17RTRP-B086931-04) from "Information Providing Technology Development based on ICT for Rail-load Passenger through the Research of Rail-load Technology" Program funded by Ministry of Land, Infrastructure and Transport of Korean government.

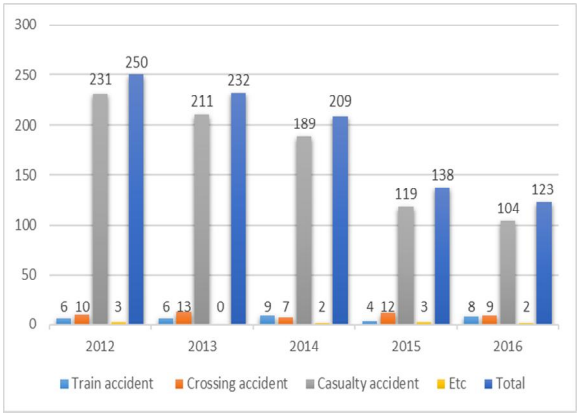

Fig. 2. Number of rail services accident 2012-2016

는 않고 있다.

철도 사고가 발생하거나, 출퇴근 시간에 많은 사람들이 몰려 열차가 제시간에 당 역을 빠져나가지 못하게 되면, 다음 역부터는 열차의 연쇄 지연이 발생하게 된다. 철도 운영자나 열차 이용객들이 이 문제 발생 사실을 신속하게 파악하지 못하면, 사고처리나 열차의 운행시간 조정에 늦어지며, 그렇게 되면 열차 이용객들의 시간을 허비하여 불편을 겪게 될 것이다. 그러므로 본 연구에서는 실시간 열차 운행 상태에 대한 정보를 신속하게 파악하고 활용하는데 도움을 주기 위해 Social Network Service(SNS)를 이용한 철도 운행 정보 정형화 방법을 제안한다. SNS는 누구나 사용할 수 있고, 자신이 가지고 있는 정보들을 인터넷에 올려 공유할 수 있다. 누구든 열차 운행에 문제를 발견하고, 그 정보를 올리면, 다른 사람들이 공유된 정보를 보고 돌발 상황을 인지할 수 있다. SNS에서 수집된 텍스트들의 필요 정보들을 추출 및 활용하기 위해서는, 비정형 텍스트를 정형화된 데이터로 가공하여 필요 정보들을 추출해야 한다. 본 논문에서는 한국철도공사와 서울교통공사의 트위터(Twitter)에서 공식적으로 올리는 메시지와 댓글을 크롤링(Crawling)하여, 총 여섯 가지 항목(일시, 철도 운영기관, 호선, 역, 방향, 열차 운행상태)과 관련된 키워드를 추출하여 정형화하는 과정을 소개한다. 여섯 가지 항목 중, 일시, 철도 운영기관, 호선, 역, 방향은 미리 추출된 키워드로 예측이 가능하나, 열차 운행상태는 몇 가지 세부 상태로 분류하여야 하고 키워드 매칭 만으로는 정확한 분류에 한계가 있어 딥러닝을 이용한 문서 분류 기법을 적용한다.

문서 분류에 많이 사용되는 기법인 Term Frequency-Inverse Document Frequency(TF-IDF)은 단어의 빈도수와 단어가 여러 문서에 등장하는 빈도수를 계산하여 단어의 특성을 추출한다[3,4]. 추출된 단어의 특성은 벡터로 표현되는데, 이 벡터를 가지고 Support Vector Machine(SVM)과 같은 분류기들을 사용하여 문서를 분류한다. 다른 방법으로 D. W. Kim 등[5]은 영상인식 분야에서 이미지의 특징을 찾는 데 많이 사용되고 있는 Convolutional Neural Network(CNN)[6]이 텍스트의 특징을 찾는 데에도 좋은 성능을 보인다는 점을 착안하여[7], 한국어 신문 기사를 특정 범주로 분류를 시도하였다. Word2Vec[8]과 Doc2Vec[9]을 사용하여 텍스트를 벡터화하고 CNN을 적용한 방법을 비교한 결과, 각각 86.89%와 89.88%의 정확도를 보여주었다. 본 논문에서는 위에서 설명한 TF-IDF와 Word2Vec을 사용하여 비교한다.

본 논문은 크게 서론, 본론, 결론으로 구성된다. 본론의 전반부에서는 데이터 정형화 전체 구조, 데이터 수집, 데이터 셋 생성, 텍스트 수치화 방법 중 하나인 워드 임베딩(word embedding) 방법, 텍스트 분류를 위한 CNN 적용 과정을 소개하고, 본론의 후반부에서는 이러한 과정에 따라 얻은 열차 운행 상태 분류 결과와 그 밖의 열차 정보 추출 결과를 제시한다. 마지막으로 결론에서는 논문의 내용을 간략히 요약하고 결과에 대한 분석과 향후 연구 방향에 대해서 논한다.

II. The Proposed Scheme

1. Data Structuring Process

'Fig. 3'은 이 논문에서 제안하는 데이터 정형화 흐름도이다.

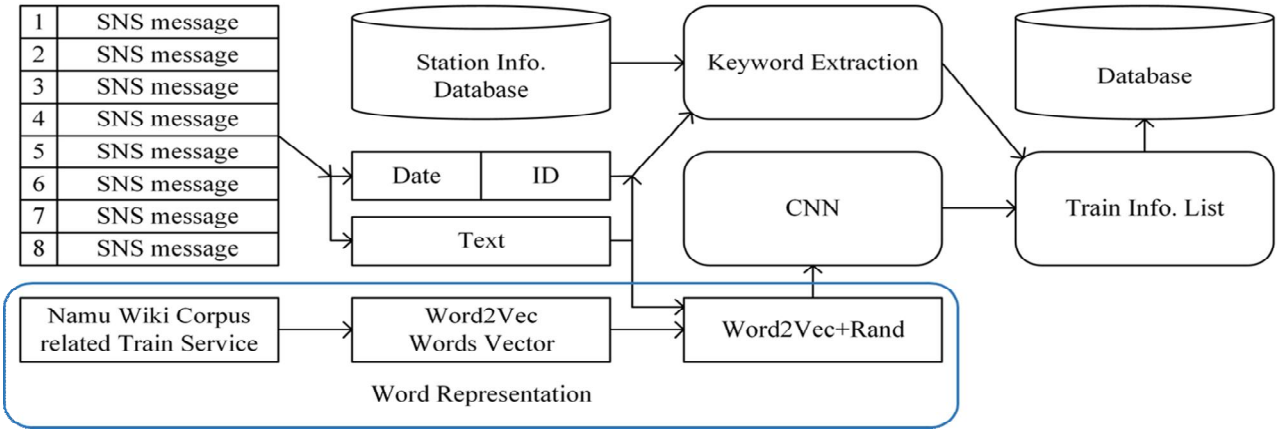

Fig. 3. Data Structuring Process

Table 1. Types of rail service

Index	Types of rail service	Description
1	Normal Operation	The train is running normally at the present time.
2	Train Delay	Condition in which a train is unable to run at a fixed time for various reasons.
3	Operating Obstruction	Condition in which a train is not operating normally due to mechanical defect or unknown cause.
4	Train Accident	Condition in which a train is not operating normally due to natural and human disasters.
5	Extra Train	Additional train operation notifications.

우선 SNS에 올라오는 글들은 시간 순으로 저장되어 SNS timeline을 생성한다. 각각의 글들은 시간 순으로 자기만의 고유 번호 값을 지니고 있어서, 이 고유 번호를 이용하여 최신의 글부터 오래된 글까지 SNS message를 수집할 수 있다. 수집된 SNS message에는 여러 태그 정보들이 붙어있어, 글쓴이의 ID와 글을 쓴 날짜의 추가 정보를 얻을 수 있다. 그 중 생성 날짜, 작성자 ID, 본문(Date, ID, Text)으로 만들어진 데이터는 키워드들을 정규식으로 추출(Keyword Extraction)하는데 사용된다. 키워드를 추출할 때 정확한 역 이름, 방향등으로 추출하기 위하여 역 정보(Station Info.) DB를 참조한다. 다음으로 CNN에는 본문 텍스트만 입력하여 열차 운행 상태 키워드를 출력하는데, 여기서 사용되는 입력 텍스트는 Word2Vec으로 단어를 표현하되 Word2Vec으로 표현하지 못 한 단어에 대해서는 랜덤 값으로 할당하였다. Word2Vec 표현은 열차에 대한 키워드가 들어간 나무 위키 페이지에서 말뭉치 데이터를 사용했으며, Skip-gram으로 학습하였다. 그 후 추출된 다른 키워드들과 종합하여 열차 정보 리스트(Train Info. List)를 생성한다. 마지막으로 이 리스트는 데이터베이스 서버(Database)에 저장된다.

2. Data Crawling

본 연구에 사용한 데이터는 파이썬(Python)의 트위터 크롤링 모듈 Tweepy를 사용하여 크롤링하고, 키워드 추출에 사용될 정보인 트윗 생성날짜, 유저 ID, 본문 텍스트 세 가지를 수집하였다. 최근 5년간(2013년~2017년) 트위터에 게시된 메시지 중 열차 운행 상태와 관련된 782개의 메시지를 선별하였으며, 'Table 1'과 같이 정상운행(1), 열차지연(2), 장애발생(3), 사고발생(4), 임시열차(5)의 다섯 가지 유형으로 나누어 데이터셋을 구축하였다.

3. Word Embedding

워드 임베딩(Word Embedding)은 자연어 처리에서(Natural Language Processing) 어떠한 단어를 실수의 벡터로 매핑하는 방법들을 말한다[4,8,9,10]. 각 단어나 연속된 단어들의 그룹에 대하여 필요에 따라 설정된 파라미터들의 의해 얻어진 확률 분포를 언어 모델링이라 한다. 이로 인하여 문장의 생성이나 분류를 할 수 있게 된다. 이 과정에서 언어 모델링을 위하여 사용된 파라미터들이 실수의 벡터로 표현될 수 있다. 이러한 벡터들을 생성하기 위하여 기계 학습(Machine Learning) 방법들이 사용된다. 본 연구에서는 우선, 어휘 사전을 구축하고, 각 어휘를 벡터로 수치화하는 Word Representation 방법에 대하여 설명한다.

3.1 Vocabulary Dictionary Build

CNN에서 사용할 텍스트들을 실수의 벡터로 치환하기 위해서, 본 연구에서는 텐서플로우(TensorFlow)[11]의 워드 임베딩 모듈의 룩업테이블(Lookup Table)을 사용하였는데, 이 룩업테이블은 정수로 표현된 단어나 단어의 그룹을 실수 벡터로 치환해준다. 단어나 단어의 그룹을 정수로 표현하기 위하여, 본 논문에서는 SNS message의 텍스트들을 형태소 분석을 통하여 일정 크기의 단어들로 나누고, 중복된 단어들을 제거한 뒤 등장 순으로 정수 값으로 인덱싱하여 어휘 사전을 구축하였다. 형태소 분석은 KoNLPy의 Twitter tag를 사용하였고, 품사 중 명사와 동사, 조동사만을 단어로 추출하여 총 1544개의 단어로 어휘 사전을 만들었으며, 결과적으로 'Table 2'와 같은 어휘 사전 및 룩업테이블이 만들어지게 된다. 각 텍스트들을 분류하기 위하여 CNN을 사용하는데, CNN을 사용하기 위해서는 입력되는 텍스트의 길이(텍스트를 구성하는 단어의 개수)가 일정해야 한다. 하지만 텍스트의 길이는 서로 다르기 때문에, 수집된 텍스트 중에서 단어 개수가 가장 많은 것을 기준으로 고정 크기를 정하고, 단어 개수가 부족한 텍스트는 패딩 값으로 채워 고정 크기로 만든다. 패딩의 값은 0으로 설정하였다.

Table 2. Words and Lookup Table

Index	Words	Distributed representation (128 dimension vector)
0	Pad	{0.0, 0.0, 0.0 … 0.0, 0.0}
1	Normally	{0.57, -0.25, 0.03 … 0.20, 0.24}
2	Running	{-0.04, -0.18, -0.01 … 0.13, 0.37}
3	It's	{-0.23, -0.44, -0.32 … 0.27, -0.03}
⋮	⋮	⋮
1543	again	{-0.40, -0.18, -0.33 … 0.38, 0.44}

3.2 TF-IDF vs. Word2Vec

TF-IDF는 정의된 수식을 사용하여 생성하는데, 문서 내 단어의 빈도수와 단어가 들어간 문서 개수를 측정하는 방식으로, t는 용어, d는 문서, D는 문서 집합일 때 수식(1, 2, 3)을 적용

$$tf(t,d) = f(t,d) \quad \cdots(1)$$

$$idf(t,D) = \log\left(\frac{|D|}{d \in D : t \in d}\right) \quad \cdots(2)$$

$$tf-idf(t,d,D) = tf(t,d) \times idf(t,D) \quad \cdots(3)$$

하여 어휘를 벡터로 변환한다[4]. Word2Vec의 경우에는 유사한 의미를 가지는 단어들을 벡터에서 거리상 가깝도록 표현한다. 거리상 가까운 단어들은 유사한 방향 벡터와, 크기를 가지게 되며, 이는 유사 단어들이 비슷한 벡터 값을 지니게 해준다. Word2Vec을 사용하게 되면, 같은 의미를 가지면 거리상 가깝게 다른 의미를 가지면 거리상 멀어지게 때문에 의미에 따라서 단어 표현이 확실하게 구분되기에 문장 분류에 효과적이다. 또한 Word2Vec은 비지도로 학습되기에 많은 말뭉치 데이터가 있다면, CNN에 학습되지 않은 단어들을 학습된 유사한 단어들로 인해 학습을 한 것으로 연산하는 효과도 기대할 수 있다. 단점 또한 존재 하는데 Word2Vec은 문장에서 단어가 존재하는 위치에 따라 벡터로 표현되는데, 단어들이 다른 의미를 지니더라도 문장에서 같은 위치에 존재하는 단어들은 유사 단어들처럼 표현된다는 것이다. 본 연구에서는 말뭉치 데이터를 나무위키[12]에서 제공하는 백업 데이터에서 철도와 관련된 키워드가 들어간 2,508개의 페이지를 분류하여, Word2Vec 학습에 사용하였다. Word2Vec 학습 방식에는 Continuous-Bag-Of-Words(CBOW)와 Skip-gram 두 가지 방식이 있는데, CBOW는 여러 단어와 공통적으로 연관된 한 단어를 예측하는 방향으로 학습하고, Skip-gram은 한 단어를 중심으로 이 단어와 연관성이 높은 여러 단어를 예측하는 방향으로 학습한다. T. Mikolov 등[8]에서 Word2Vec을 소개하고, 의미 분석에 있어서 Skip-gram이 기존의 신경망을 이용한 언어 모델을 사용한 방식과 CBOW 방식에 비해 더 높은 정확도를 나타내는 결과를 보여주었다. 그러므로 본 연구에서도 Skip-gram 방식을 사용하여 학습하였다. Word2Vec는 학습할 때 문장에서 하나의 단어를 선택하여 해당 단어 좌우에 등장한 단어들을 사용하여 학습을 한다. 해당 단어의 주변 단어를 몇 개까지 학습에 사용할 것인가를 설정하는데, 이것을 윈도우 사이즈(Window Size)라 한다. 본 연구에서는 윈도우 사이즈를 3으로 설정하여 해당 단어와 좌우의 단어 세 개씩 사용하여 학습하였다. Word2Vec의 학습이 완료되면 학습에 사용된 단어들이 고정 크기 차원의 벡터로 표현된다.

초창기에 딥러닝에서 단어의 벡터 표현은 One-hot encoding 방식이 사용되었다. One-hot encoding은 어휘 사전의 총 단어 개수만큼의 차원 수를 가지며, 딱 하나의 요소에만 '1'을 주고, 나머지 요소들은 '0'을 가진다. '1'의 위치는 단어마다 다르게 가지게 되어, 서로 독립되어 구분이 가능하다. 이런 방식으로 서로 구분이 용이하지만 이 방식은 단어의 개수가 증가할 때마다 벡터의 요소 개수가 증가하고, 다른 단어들과 연관성을 표현하지 못 한다는 한계를 가지고 있었다. 이를 극복하기 위해 Distributed representation 방식으로 벡터를 표현하는 방법이 연구되었다[10]. 이 방식은 기존의 One-hot encoding에서처럼 벡터의 하나의 요소에만 값을 주지 않고, 모든 요소에 값을 주어 벡터 표현이 분산 되도록 하였다. 그렇게 함으로써 벡터의 차원 수를 일정하게 만들 수 있었고, 더 많은 단어의 정보들을 표현할 수 있게 되었다. 본 논문에서는 CNN에서 사용될 벡터의 표현을 두 가지 방식으로 하였다. 하나는 Word2Vec, 다른 하나는 어휘 사전 단어들의 벡터를 '-1'에서 '1'사이의 랜덤 값으로 초기화한 후 CNN의 학습 중에 갱신하여 표현하는 랜덤 방식(Rand)이다. 본 연구에서 어휘 사전은 SNS message를 기반으로 구축하였고 Word2Vec 학습에 사용한 말뭉치 데이터는 나무 위키를 사용하였기 때문에 어휘 사전의 1544개의 단어 중 49개의 단어가 Word2Vec에 학습에 사용되지 않았다. 따라서, 본 연구에서는 1494개의 Word2Vec 단어와 48개의 Rand 단어를 혼합하여 Distributed representation 방식으로 'Table 2'와 같이 128차원의 고정 크기로 나타냈다. 나머지 한 개는 패딩으로 쓰였다.

4. Convolutional Neural Network

Convolution layer는 Convolution filter들을 사용하여, 텍스트 단어들 중에 중요한 단어들의 영향력을 강화하는 역할을 한다. 본 논문에서 사용한 CNN Parameter 값들은 'Table 3'과 같다.

Table 3. CNN Parameters

Parameter	Filter size	Filter number	Dropout	L2_norm
Value	2,3,4	50,60	0.5	0.1

'Fig. 4'는 Convolution layer에서 한 문장에 Filter size 2의 Convolution filter를 적용시킨 예시이다. Convolution layer의 Convolution filter는 단어의 벡터와 같은 차원 크기의 벡터를 가지고 있고, 초기 값들은 랜덤으로 초기화되며, 학습을 통하여 문장 분류 정확도가 높아지는 방향으로 필터의 값들을 갱신한다. Filter size 2의 필터를 한 개 적용시킨 다음 출력되는 값의 개수는 총 단어 수 - (Filter size - 1)과 같다. 출력 값은 Filter number만큼 생성되며, 총 단어 수 - (Filter size - 1) × Filter number의 2차원 매트릭스로 나타낼 수 있다. 다음으로 Pooling의 기본적인 방식으로는 Mean pooling과 Max pooling이 있는데, 일반적으로 주요 특징을 더욱 부각시키는 Max pooling의 성능이 더 좋은 결과가 나온다. 따라서, 본 논문에서도 Max pooling을 사용하여 'Fig. 4'의 Pooling layer의 결과와 같이 일곱 개의 값들 중 가장 큰 값만을 남긴다. CNN에는 오버피팅(Overfitting)을 방지하기 위한 Dropout과 Euclidean norm(L2_norm)을 각각 0.5와 0.1로 설정하면, Pooling layer에서 출력된 결과 개수 중 50%만 Fully Connected(FC) layer의 입력 값으로 사용되고 학습 중 FC layer의 가중치 값들이 고르게 분포하게 된다. 그림에서는 Filter size를 하나만 적용했으므로, Pooling 결과는 Filter size number × Filter number가 출력된다. 그중 50%인 Filter number * 0.5개의 값이 FC layer에 입력되어 'Table 1'과 같이 다섯 개의 유형으로 분류하는데 사용된다.

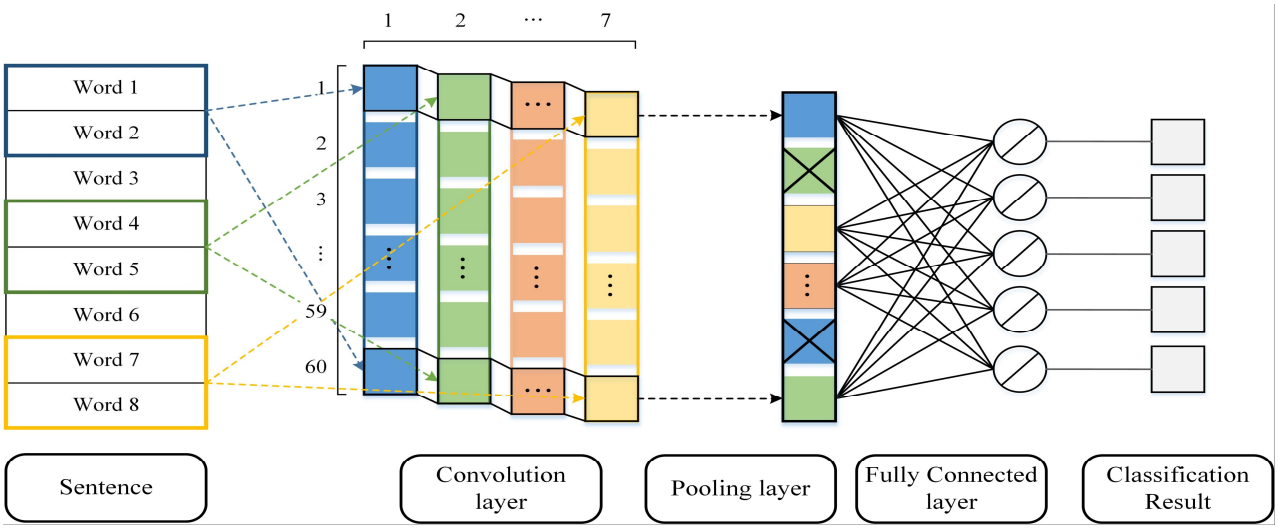

Fig. 4. An Example of CNN for sentence classification

Table 4. Epoch when the highest accuracy is obtained in each iteration

Word Representation	Filter size	Filter number	Test Count									
			1		2		3		4		5	
			Epoch	Acc	Epoch	Acc	Epoch	Acc	Epoch	Acc	Epoch	Acc
Rand	2	50	135	82	149	82	116	84	107	84	287	84
		60	155	85	118	80	256	82	150	82	105	85
	3	50	276	82	623	82	222	85	85	82	901	84
		60	128	82	437	87	501	84	569	85	487	83
	4	50	885	83	478	82	107	84	795	82	198	84
		60	518	84	277	83	630	83	212	85	521	84
Word2Vec+Rand	2	50	989	86	862	85	481	86	705	89	644	84
		60	678	86	485	89	352	85	492	87	273	86
	3	50	494	87	439	88	812	87	615	85	496	89
		60	774	88	674	89	361	88	964	89	199	86
	4	50	181	87	884	87	811	87	961	87	876	87
		60	135	87	352	88	95	89	679	86	443	87

III. Experimental results

1. Rail Service Status Classification

CNN의 학습은 1000 Epoch 동안 하였고, 1 Epoch 당 19 step으로 되어있으며, 1 step은 64 batch size의 학습 데이터를 가진다. 그리고 Learning rate는 0.001로 설정하였다. 'Table 3'의 L2_norm Regularization을 사용하면 학습 중에 정확도가 일정한 값으로 수렴되지 않고 계속 변화하는데 본 연구에서는 step마다 CNN의 분류 정확도가 가장 높을 때를 측정하였다. 학습이 끝나면 CNN 모델을 초기화하여 다시 학습 후 측정하였으며 이 작업을 5번 반복하였다. 'Table 4'는 각 학습 중 정확도가 가장 높은 시점이 몇 Epoch 인지를 나타낸다. Word2Vec+Rand는 총 단어 1544개 중 1494개를 Word2Vec으로 48개를 Rand로 합하여 사용한 방식이다. 학습 중에 CNN의 FC layer 가중치들은 L2_norm Regularization을 통하여 지속적으로 고르게 작아졌고, 반대로 Word2Vec+Rand의 경우 Convolution filter의 값이, Rand의 경우 Convolution filter의 값과 단어 벡터의 값들이 조금씩 커지는 모습을 보였다. 두 방식의 차이가 발생하는 이유는 Word2Vec의 단어들은 상수로서 설정되어 고정된 값이기 때문이며, 단어 벡터 대신에 Convolution filter의 값이 더 크게 변화했다. 테스트 결과 정확도는 Rand 방식이 87%, Word2Vec+Rand가 89%로 가장 높게 나타났다. 아래의 'Table 5'는 Dropout과 L2_norm을 사용하지 않았을 때의 결과 값을 정리한 것이다. 두 개를 모두 사용하지 않을 경우의 최고 정확도는 4%가 낮게 측정되었다. 'Table 4, 5'의 테스트에서 사용된 CPU는 intel® Core™ i5-3570 3.40GHz 4코어이며, 한 번의 테스트를 하는데 대략 8분의 시간이 걸렸다.

Table 5. Accuracy without Dropout and L2_norm (Word2Vec+Rand+CNN, Filter size 3, Filter number 60)

Parameter	Test Count				
	1	2	3	4	5
without Dropout	87	85	84	84	87
without L2_norm	84	86	86	84	85
without Dropout+L2_norm	83	84	85	84	83

2. Accuracy Evaluation

'Table 6'은 Word2Vec과 TF-IDF의 SNS message 열차운행 상태 유형 분류 정확도를 보여준다. 성능 비교를 위하여, TF-IDF와

Table 6. Comparison of classification accuracy

Word Representation	Model	Parameter	Accuracy(%)
TF-IDF	SVM/SVC	Kernel=linear, C=1.0	79
	BNB	Alpha=1.0	73
	LR	C=10.0	80
Word2Vec	SVM/SVC	Kernel=linear, C=8.0	77
	BNB	Alpha=0.1	57
	LR	C=7000.0	80
Rand	CNN	Filter size=3, Filter number=60	87
Word2Vec+Rand	CNN	Filter size=3, Filter number=60	89

Table 8. Examples of information list

Date and Time	Organization	Train Line	Station	Direction	Type of rail service
201709131408	Korea Railroad	중앙선	양평 원덕	-	정상운행
201707161147	Korea Railroad	중앙선	-	-	장애발생
201708040954	Korea Railroad	충북선	조치원 봉양	-	사고발생
201708040954	Seoul Metro	4	사당	상선	열차지연
201701261935	Seoul Metro	1~4	-	-	임시열차

Word2Vec은 일반적으로 많이 사용되는 SVM, Bernoulli Naive Bayes(BNB), MNB, Logistic Regression(LR)을 사용하여 분류하였다. 여기서 TF-IDF는 하나의 SNS message를 한 문서로 지정하여 표현했으며, Word2Vec은 유사도 행렬[13]에서 한 텍스트에 등장하는 단어 벡터들의 평균값을 구하여 1544개의 차원으로 나타냈다. 모델들의 파라미터를 간단히 설명하면 SVM의 Kernel은 어떤 방식으로 데이터 샘플들을 분류하는지에 대한 함수이다. linear로 설정하면 최적의 선형 결정 경계를 찾아준다. C값은 어느 클래스 데이터 샘플이 다른 클래스의 경계를 넘어갈 수 있도록 허용하는데 작을수록 많이 허용한다. BNB, MNB의 Alpha 값은 전에 학습되지 않은 데이터들이 등장하더라도 일정 값을 보정해 주는 것이며 작을수록 적게 영향을 미친다. LR의 C값은 SVM의 C와 같은 효과를 가진다. 하이퍼 파라미터의 값은 10000에서 0.01 사이로 가장 높은 정확도가 나오도록 설정하였다. 정확도를 보면 두 표현 모두 LR로 분류하였을 때 80%로 가장 높게 측정되었다. Word2Vec+Rand+CNN과 Rand+CNN 두 방식의 정확도는 'Table 4'의 5번의 측정 중 가장 높은 값을 표시하였다. 결과적으로 Word2Vec+Rand+CNN이 TF-IDF+LR, Word2Vec+LR에 비해서는 9%, Rand+CNN에 비해서는 2%가 높은 89%로 측정됐다.

3. Rail Service Information List

키워드 추출은 보통 통계 또는 확률을 기반으로 추출된다. 하지만 본 연구에서는 데이터가 많지 않고 역의 경우 각 역의 빈도수가 적어, 정규식을 통하여 키워드들을 추출하였다. 정규식은 문자열 패턴을 정하면 문장에서 패턴에 맞는 문자열을 검색하거나 치환하여 준다. 본 연구에서는 총 여섯 가지 항목(일시, 철도 운영기관, 호선, 역, 방향, 열차 운행 상태) 중 네 가지 항목(철도 운영기관, 호선, 역, 방향)을 정규식을 통하여 추출하였다. 네 가지 키워드들 중 역 정보를 우선적으로 추출하였고, 추출된 역 정보를 기반으로 운영 회사와 열차 호선 정보를 추가적으로 'Table 7'의 데이터베이스에서 읽어왔다. 읽어 온 철도 운영 회사 정보는 SNS message의 Id 정보와 함께 철도 운영 회사를 알아내는데 쓰였으며, 열차 호선 정보는 SNS message의 텍스트에서 정규식을 사용하여 추출된 호선 정보와 함께 열차 호선을 알아내는데 사용되었다. 일시 정보는 트위터에 트윗이 생성된 시간을, 열차 운행 상태는 CNN에서 출력된 열차 운행 상태 정보를 키워드로 추출하였다. 이상으로 최종 결과인 'Table 8'의 열차 운행 정보 리스트가 생성되며, 생성된 리스트는 데이터베이스 서버에 저장된다. 'Table 8'의 열차 운행 정보 리스트는 각 열차 운행 유형의 최신 결과를 하나씩 샘플로 뽑은 것이다. SNS message의 텍스트에는 짧은 댓글의 간단한 운행상태만을 나타내는 경우도 많아 추가적인 정보가 없을 경우에는 빈칸으로 두었다. Direction은 구간 정보만을 표현하고 생략한 경우가 종종 있었다. Train line, Station, Direction 세 개의 항목이 빈칸이면, 모든 열차에 해당되는 경우이거나 SNS message의 글쓴이가 정보를 부족하게 올린 경우이다. 임시 열차 운행의 경우 철도 운영 측에서 올리는 글이기 때문에, 링크로 추가 정보가 있거나, 모든 열차를 뜻한다.

Table 7. Station information in the Database

Organization	Train Line	Station
Seoul Metro	1	서울
Seoul Metro	2	강남
Korea Railroad	1	천안

IV. Conclusions

본 연구는 SNS을 통하여 수집한 비정형 데이터를 열차 운행 정보 리스트로 정형화하였다. 일시와 열차 운행 상태를 제외한 다른 키워드들은 정규식을 통하여 추출하였고, 오추출(Error Extraction) 되었을 경우에는 불용어로 취급하여 제거되었다. 일시는 SNS message가 SNS에 업데이트된 시간을, 열차 운행 상태는 SNS message의 텍스트를 TF-IDF와 Word2Vec을 사용하여 단어를 벡터로 표현하고 SVM과 CNN 등으로 분류하여 키워드로 추출하였다. 그 중 Word2Vec과 Rand 방식을 결합하여 CNN으로 열차 운행 상태를 추출한 결과는 89%의 정확도를 보여 주었다. CNN은 다른 방식들과 비교하여 10%정도 높은 정확도를 나타냈으며, 특히 설정 파라미터인 CNN의 Convolution filter와 Dropout 그리고 L2_norm으로 성능이 향상되는 것을 확인할 수 있었다. 반면 학습 시간에서는 다른 분류 모델들의 학습 시간이 거의 없는데 비해 CNN은 8분 정도가 소요되며 더 많은 시간이 소요되었다. 또 다른 단점으로는 학습을 하는 중에 측정된 정확도가 매번 다른 모습을 보였는데, 이는 CNN의 파라미터들의 초기 값들이 랜덤으로 설정되고 적은 학습 데이터로 학습되기 때문인 것으로 분석된다. 하지만 SNS message의 단문 텍스트를 여러 분류 모델들로 분류하는데 있어 Word2Vec과 TF-IDF의 표현 방식들은 적은 데이터로도 높은 정확도를 보여주었으며, 특히 CNN을 통한 분류 성능은 뛰어난 모습을 보였다. 이런 방법들을 통해 최종적으로 얻어진 정형화된 열차 운행 정보 리스트는 철도 운영자와 열차 이용객들에게 제공되어 보다 나은 서비스를 제공하는데 활용될 수 있을 것이다.

REFERENCES

[1] K. indicator, *http://www.index.go.kr*
[2] Rail Safety Information System, *http://www.railsafety.or.kr*
[3] N.G. Kim, D.H. Lee, H.C. Choi and W.X.S. Wong, "Investigations on Techniques and Applications of Text Analytics," The Journal of Korean Institute of Communications and Information Sciences, Vol. 42, No. 2, pp. 471-492, February 2017.
[4] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing and Management, Vol. 24, No. 5, pp. 513-523, January 1988.
[5] D.W. Kim and M.W. Koo, "Categorization of Korean News Articles Based on Convolutional Neural Network Using Doc2Vec and Word2Vec," Journal of Computing Science and Engineering, Vol. 44, No. 7, pp. 742-747, July 2017.
[6] Y. LeCun and Y. Bengio, "Convolutional Network for Images, Speech, and Time-Series," The handbook of brain theory and neural networks, pp. 255-258, 1998.
[7] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing, pp. 1746-1751, Doha, Qatar, October 2014.
[8] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781v3, September 2013.
[9] Q. Le and T. Mikolov, "Distributed Representation of Sentences and Documents," Proceedings of the 31st International Conference on International Conference on Machine Learning, Vol. 32, No. 2, pp 1188-1196, Beijing, China, June 2014.
[10] Y. Bengio, R. Ducharme, P. Vincent and C. Jauvin, "A Neural Probabilistic Language Model," Journal of Machine Learning Research, Vol. 3, No. 6, PP. 1137-1155, February 2003.
[11] Machine Learning Framework, *www.tensorflow.org*
[12] Wiki Web Site, *https://namu.wiki*
[13] W.J. Kim, D.H. Kim and H.W. Jang, "Semantic Extension Search for Documents Using the Word2vec," Journal of the Korea Contents Association, Vol. 16, No. 10, PP. 687-692, October 2016.

Authors

Jin Gyu Park received the B.S. degree in Computer Engineering from Hanbat National University, Korea, in 2016. he is currently in the M.S. course in the Department of Computer Engineering, Hanbat National University. He is interested in artificial intelligence and machine learning.

Hwa Yeon Kim received the B.S. degree in Computer Engineering from Hanbat National University, Korea, in 2017. She is currently in the M.S. course in the Department of ICT, University of Science and Technology in Korea.She is interested in natural language processing, dialogue system, deep learning.

Hyoung Geun Kim received the B.S. degrees in Computer Engineering in 2005. Hyoung Geun Kim is currently a Senior Researcher in the Smart R&D Center U-Core System. He interested in artificial intelligence, deep learning.

Tae Ki AN received the M.S. degree in Electronic Engineering from Kyungpook National University, Daegu, Korea, in 1996, and the Ph.D. degree in Department of Electrical and Computer Engineering from Sungkyunkwan University, Seoul, Korea in 2011. He is a Chief Researcher in Korea Railroad Research Institute. His research interests are artificial intelligence, pattern recognition, and video analysis.

Hyunbean Yi received the B.S., M.S. and Ph. D. degrees in Computer Science and Engineering from Hanyang University, Korea, in 2001, 2003, and 2007, respectively. He had been a Postdoctoral Researcher at University of Massachusetts, USA from 2007 to 2009 and at Nara Institute of Science and Technology (NAIST), Japan from 2009 to 2011. He is currently a professor in the Department of Computer Engineering at Hanbat National University. He is interested in machine Learning and circuit/flash memory aging monitoring.