Journal of Convergence for Information Technology
Vol. 9. No. 8, pp. 1-8, 2019
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2019.9.8.001

기계학습을 이용한 한국어 대화시스템 도메인 분류

정영섭
순천향대학교 빅데이터공학과 교수

Machine Learning Based Domain Classification for Korean Dialog System

Young-Seob Jeong
Professor, Department of Big Data Engineering, Soonchunhyang University

요 약 대화시스템은 인간과 컴퓨터의 상호작용에 새로운 패러다임이 되고 있다. 자연어로써 상호작용함으로써 인간은 보다 자연스럽고 편리하게 각종 서비스를 누릴 수 있게 되었다. 대화시스템의 구조는 일반적으로 음성 인식, 자연어 이해, 문맥 파악 등의 여러 모듈의 파이프라인으로 이뤄지는데, 본 연구에서는 자연어 이해 모듈의 도메인 분류 문제를 풀기 위해 convolutional neural network, random forest 등의 기계학습 모델을 비교하였다. 사람이 직접 태깅한 총 7개 서비스 도메인 데이터에 대하여 각 문장의 도메인을 분류하는 실험을 수행하였고 random forest 모델이 F1 score 0.97 이상으로 가장 높은 성능을 달성한 것을 보였다. 향후 다른 기계학습 모델들을 추가 실험함으로써 도메인 분류 성능 개선을 지속할 계획이다.

주제어 : 한국어 대화시스템, 자연어이해, 도메인 분류, 기계학습, 랜덤 포레스트

Abstract Dialog system is becoming a new dominant interaction way between human and computer. It allows people to be provided with various services through natural language. The dialog system has a common structure of a pipeline consisting of several modules (e.g., speech recognition, natural language understanding, and dialog management). In this paper, we tackle a task of domain classification for the natural language understanding module by employing machine learning models such as convolutional neural network and random forest. For our dataset of seven service domains, we showed that the random forest model achieved the best performance (F1 score 0.97). As a future work, we will keep finding a better approach for domain classification by investigating other machine learning models.

Key Words : Korean dialog system, Natural language understanding, Domain classification, Machine learning, Random forest

*This work was supported by the Soonchunhyang University Research Fund. This research was financially supported by the Ministry of SMEs and Startups(MSS), Korea, under the "Innovative enterprise technology development Program (Deep learning based industrial / occupational classification automatic coding system development, S2631746)" supervised by the Korea Technology and Information Promotion Agency for SMEs(TIPA).

*Corresponding Author : Young-Seob Jeong(bytecell@sch.ac.kr)

Received July 2, 2019 Revised July 23, 2019
Accepted August 20, 2019 Published August 28, 2019

1. Introduction

컴퓨터가 발명된 이후로 인간은 컴퓨터를 다루기 위한 다양한 상호작용 방법을 고안해왔다. 최초의 키보드, 마우스가 등장하면서 인간은 손가락을 이용한 접촉으로써 컴퓨터와 손쉽게 상호작용할 수 있게 되었으며, 수십 년간 이러한 패러다임이 지속되어왔다. 넘쳐나는 정보 틈에서 원하는 정보를 찾기 위해 각종 검색 서비스가 개발되었고, 효과적인 검색결과를 제공하기 위한 기술과 검색어 추천을 위한 연구들이 등장했다.

대화시스템은 인간이 '자연어' 음성 또는 텍스트로 컴퓨터와 상호작용하도록 돕는다. 수십 년간 인간이 지속해왔던 컴퓨터와의 상호작용 방법에 대한 새로운 패러다임을 제시하고 있으며, 이 기술이 인류 문화에 미치는 파급력은 매우 크다고 볼 수 있다. 대한민국에서도 한국어로 동작하는 대화시스템들이 개발되어 서비스되고 있다.

대화시스템들은 일반적으로 음성 인식, 자연어 이해, 문맥 파악 등의 모듈들의 연속된 구조를 가지는데, 특히 자연어 이해 모듈은 사용자가 입력한 텍스트 문장으로부터 구조화된 정보를 추출 및 정규화하는 역할을 수행한다. 자연어 이해 모듈에서 수행하는 작업 중에 '도메인 분류'는 사용자 입력 문장이 대화시스템에서 제공하는 어떤 서비스(도메인)에 속하는지 분류하는 작업인데, 도메인을 잘못 분류하게 될 경우 대화시스템에서 올바른 응답문장을 생성할 수 없으므로 매우 중요한 작업이라고 볼 수 있다.

본 연구에서는 대화시스템에 입력된 '한국어 문장'에 대한 도메인 분류 문제를 풀고자 하였으며, 직접 구축한 한국어 데이터에 대하여 기계학습 기술들을 적용하여 성능을 비교하였다.

2. Background

2.1 Dialog System

대화시스템은 인간과 컴퓨터가 자연어로 대화할 수 있도록 돕는 시스템이다. 음성 인식 기술, 음성 합성 기술, 자연어 처리 기술 등의 발전에 힘입어 대화시스템은 지난 수년간 산업계의 블루오션 아이템으로 자리매김했으며, Amazon Alexa[1], Naver Clova[2], Samsung Bixby[3] 등을 비롯한 제품들이 시장에서 성공적인 반응을 얻고 있다.

대화시스템은 일반적으로 Fig. 1 에 묘사된 것과 같이 5가지 모듈의 순차적인 실행으로써 동작한다. 사용자가 대화시스템에 음성 신호를 입력하면, 첫 번째 모듈인 Speech Recognition (SR) 모듈에서는 음성 신호를 텍스트로 변환한다. 이 모듈에서 수행하는 작업을 Speech-To-Text (STT) 라고 하며, 음성 신호에 존재하는 순차적인 패턴을 기반으로 유망한 단어 연속을 생성한다. 이 모듈에서 생성된 결과물은 여러 결과 후보들에 대한 정보를 포함하는 격자 형태일 수도 있다.

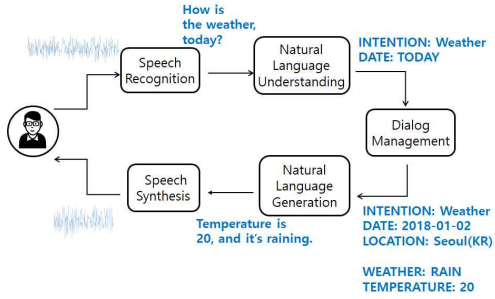

Fig. 1. Pipeline process of dialog system

SR 모듈의 결과로 생성되는 텍스트는 Natural Language Understanding (NLU) 모듈의 입력으로 사용된다. NLU 모듈에게 주어진 텍스트는 사용자의 음성을 텍스트로 변환한 것이므로 '자연어' 문장이며, 이 자연어 문장으로부터 구조화된 정보를 추출한다. 예를 들어, '지금 날씨 어때?'라는 자연어 문장으로부터 {의도: 날씨, 날짜: 오늘} 이라는 정보를 추출할 수 있을 것이다. NLU 모듈이 생성한 구조화된 정보는 Dialog Management (DM) 모듈의 입력으로 사용된다. DM 모듈은 사용자와 대화시스템 간의 대화 문맥을 파악, 유지하는 역할을 수행한다. 예를 들어, 사용자가 '오늘 날씨 어떠니?'라고 질문하여 시스템이 적절한 날씨 정보를 응답문으로 제공했다고 가정하자. 사용자가 '내일은?'이라고 다시 질문했을 때, 내일 날씨가 어떤지 질문했다는 것을 대화 문맥을 고려하여 유추하는 과정이 필요하며, DM 모듈이 이 과정을 수행하게 된다. 만약 DM 모듈이 적절하게 동작하지 않는다면, '내일은?'이라는 문장에 대하여 다양한 오인식, 이를테면 '내일은'이라는 제목의 노래를 틀거나 '내일'이라는 가수에 대한 정보를 찾는 등의 오인식을 유발하게 되며, 이는 결국 서비스 품질 하락의 원인이 된다. 즉, NLU 모듈은 사용자로부터 주어진 한 개의 텍스트를 이해하는 것에 집중하는 반면, DM 모듈은 지난 입력들에 대한 정보, 즉 대화 문맥을 바탕으로 사용자가 가진 진정한 의도가 무엇인지 추론한다. 따라서, DM 모듈은 현재 시간, 현재 장소 등과 같은 상황에 대한 정보를 정규화하는 역할도

수행한다. 가령, '오늘 날씨 어때?'라는 문장으로부터 '오늘'이 몇 년도 몇월 몇일인지를 정규화하고, 현재 위치정보를 '서울'로 정규화하는 등의 역할을 수행한다. 마지막으로, DM 모듈을 거치면서 시스템이 어떤 행동 또는 응답을 해야 하는지에 대한 결정도 이뤄진다. 만약, 사용자 의도를 정확히 이해하지 못했다면 '다시 묻는 행동'이 필요할 것이고, 적절한 응답문을 생성할 준비가 되었다면 '대답하는 행동'이 필요할 것이다. 이를 Speech Act (SA) 라고 부른다.

DM 모듈에서 정규화된 정보는 Natural Language Generation (NLG) 모듈에 전달되며, NLG 모듈은 사용자에게 전달될 응답문을 자연어 형태로 생성한다. 생성된 자연어 형태의 응답문은 Speech Synthesis (SS) 모듈에 전달되며, SS 모듈은 사용자에게 전달될 응답문 음성을 생성한다. 결과적으로 사용자는 자연어 음성을 통해 시스템에 의사를 전달하고, 자연어 음성을 통해 시스템으로부터 응답을 전달받게 된다. 가장 처음과 뒤에 위치한 SR 모듈과 SS 모듈을 제외한 나머지 3개의 모듈로만 구성된 시스템은 사용자와 자연어 텍스트를 통해 소통할 수 있게 된다.

2.2 Domain Classification

대화시스템의 NLU 모듈은 자연어 텍스트로부터 구조화된 정보를 추출하는 역할을 수행한다. 구조화된 정보는 사용자의 '의도', '슬롯' 등을 포함하는데, '의도'는 사용자가 받고자 하는 서비스의 종류, 혹은 사용자가 원하는 동작을 의미한다. 대화시스템들마다 '의도'를 표현하는 방법은 차이가 있을 수 있으나, '의도'를 예측하는 과정은 반드시 포함되어 있다. 왜냐하면, 사용자 '의도'를 예측하지 못하면 시스템이 적절한 응답문을 생성할 수 없게 될 것이 자명하기 때문이다. 가령, '내일 날씨 알려줘'라는 자연어 문장으로부터 '날씨'에 관한 서비스를 받고 싶다는 것을 정확히 인지하지 못하고 '전철' 시간표에 관한 서비스를 받고 싶다는 의도로 오인식을 하게 된다면, 사용자가 의도하지 않은 엉터리 응답문이 전달될 것이다. 이처럼, 사용자 '의도'에는 서비스 종류(예: 날씨, 전철)가 포함되는데, 이 서비스 종류를 정확히 예측하는 것은 대화시스템 성능에 지대한 영향을 미치며, 이 서비스 종류를 '도메인'이라고 부른다.

NLU 모듈은 일반적으로 '도메인'을 예측한 후, 다른 필요 정보들을 추출하는 방식으로 구현되곤 한다. 예를 들어, '내일 날씨 알려줘'라는 문장에 대하여, 도메인 추출 과정을 통해 {도메인: 날씨} 정보를 얻을 수 있고, 슬롯 추출 과정을 통해 {도메인: 날씨, 날짜: 20XX-XX-XX}와 같이 정보가 추가된다. 다시 말해서, NLU 모듈 내에서 도메인 분류 과정은 통상적으로 가장 처음에 이뤄지고, 인식된 도메인을 바탕으로 나머지 정보의 정규화 과정이 이뤄진다. 따라서, 도메인 분류 실패는 다른 모든 정규화 작업에 영향을 미치게 되므로, 매우 중요하다.

한국어에 대한 도메인 분류와 연관된 연구는 거의 수행된 적이 없는데, 그 이유는 도메인 분류를 위한 데이터는 서비스 품질과 직접적인 연관이 있기 때문에 충분한 양의 데이터를 가지고 있는 기업들에서 데이터를 공개하기 어렵기 때문이다[4]. 본 연구에서는 사용자가 대화시스템에 입력값으로 제공한 '한국어 텍스트 문장'에 대한 도메인 분류 문제를 목표로 하였으며, 직접 구축한 데이터를 사용하여 기계학습 모델들의 성능을 비교하였다.

2.3 Machine Learning

2.3.1 Convolutional Neural Network

양질의 많은 데이터가 쌓이고 이 데이터를 활용할 수 있는 다양한 기계학습 기술이 발전하는 가운데, 현재 가장 주목받는 기술은 딥러닝 기술이다. 딥러닝 기술은 인공신경망 모델의 은닉층을 여러 개 쌓음으로써 복잡한 패턴을 효과적으로 잡아낼 수 있는 기술로서[5], 그 중에서도 Convolutional Neural Networks (CNN) 모델은 이미지 데이터 분석에 극적인 성능 향상을 가져다주었다[6-9].

CNN 모델은 일반적으로 1개 이상의 convolutional layer와 1개 이상의 pooling layer들로 구성되는데, convolutional layer는 입력값의 국소적인 패턴을 잡아내는 역할을 수행하며, pooling layer는 주어진 입력값에서 결과값 생성에 가장 중요한 영향을 미치는 값을 선택하는 역할을 수행한다. 여러 개의 convolutional layer와 pooling layer를 적절히 조합하여 다양한 이미지 데이터 관련 task에 적용되었는데, 특히 VGGNet[7], ResNet[8] 등은 층을 효과적으로 깊게 쌓을 수 있는 서로 다른 방법을 제시함으로써 이미지 분석 성능 향상에 크게 기여하였다.

CNN 모델은 이미지 데이터뿐 아니라 텍스트 데이터에도 적극 활용되었다[10-13]. CNN 모델이 텍스트 분류 문제에 적용될 때, convolutional layer는 복합된 임의의 feature (예: n-gram 기반의 feature) 들을 잡아내게 되며, pooling layer는 임의의 threshold 값을 기준으로 결과에 영향을 주는 값들을 골라내는 역할을 수행함으로써 효과적으로 텍스트 분류를 수행한다[14].

2.3.2 Random Forest

본 연구에서는 사용자가 대화시스템에 입력한 텍스트 문장에 대한 도메인 분류 문제를 풀기 위해 CNN 모델을 적용하였다. CNN 모델을 비롯한 딥러닝 기법의 장점 중 한 가지는 데이터에 내재된 임의의 feature들을 모델이 스스로 감지해내기 때문에 기존의 기계학습 모델들에 비해 feature engineering에 들어가는 노력이 줄어든다는 점이다. 이것은 분명한 사실이지만, 대화시스템 도메인 분류 문제를 정확히 이해하고 데이터 특징을 파악했을 때 비로소 최적의 CNN 모델의 구조(층 개수, 층의 크기 등)를 설계할 수 있으며, CNN 모델에 입력될 값들에 대한 정의는 결국 사람이 해야 하고 모델 파라메터 최적화를 위한 engineering에 충분한 노력을 기울였을 때 만족스러운 결과를 얻을 수 있다.

딥러닝 기술이 가장 주목받는 가운데, 본 연구에서 비교 대상으로 적용한 또 다른 기계 학습 모델은 Random Forest (RF) 이다[15]. RF 모델은 여러 개의 의사결정트리 생성을 위한 bagging 기법으로 해석이 가능한데, 각 의사결정트리의 bias를 작게 유지하면서도 의사결정트리의 한 계점이었던 큰 variance를 줄이는 효과를 가지게 된다[16]. 특히, 데이터 샘플을 무작위로 추출하는 과정에서 임의의 feature 조합으로부터 의사결정에 도움이 되는 최적의 규칙을 찾아내게 되며, 각 의사결정트리의 깊이와 노드 분할 조건을 조절함으로써 성능을 최대화하면서도 과적합(overfitting) 문제를 피하도록 조정이 가능해진다. 이렇듯 작은 bias와 variance를 갖는 장점을 통해 텍스트 데이터 분석에도 좋은 성능을 보여왔다[17,18].

특히, RF 모델은 본 연구에서 다루는 대화시스템에 입력되는 짧은 문장 데이터를 다루기에 적절하다고 볼 수 있다. 각 문장은 전체 토큰의 집합 vocabulary V에서의 index의 연속으로 표현하게 되는데, V = ['불이야', '으악', ...]일 경우, '으악 불이야'라는 문장은 [1, 0, PAD, ...]가 될 것이다. 여기서 PAD는 모든 문장을 같은 크기로 맞춰주기 위한 패딩을 의미한다. Vocabulary index가 '값'을 의미하는 것이 아니므로 vocabulary index들을 nominal value로써 취급해야 하며 이를 위해 흔히 이용되는 방법은 one-hot 벡터로써 표현하는 것이다. 즉, 각 토큰을 $|V|$ 크기의 벡터로 표현하되, 해당 토큰의 vocabulary index번째 값만 1이 되고 나머지는 0이 된다. 결국 각 문장은 1 또는 0 값을 가지는 sparse 벡터로서 매우 큰 feature dimension을 가지게 되는데, 이러한 feature들 중에서 의사결정에 중요한 feature들을 선택적으로 고려함으로써 효과적인 분류가 가능해진다. 물론 각 의사결정트리의 bias가 커질 위험이 있다고 볼 수 있으나, 한 문장에 존재하는 토큰 개수가 적기 때문에 각 의사결정트리에서 중요한 feature들을 모두 고려하게 될 가능성이 높다고 볼 수 있다.

3. Machine Learning Based Domain Classification

3.1 Data Investigation

본 연구의 목표는 대화시스템에 입력된 한국어 텍스트 문장에 대한 도메인 분류 문제를 푸는 것이다. 본 연구의 실험에 사용하는 한국어 데이터 샘플을 Table 1에서 볼 수 있는데, 대화시스템에 입력되는 텍스트는 대부분 매우 짧은 문장이라는 점을 알 수 있다. 예를 들어, 사용자가 '날씨'에 대한 서비스를 받고자 하는 경우, '내일 날씨 어때?', '오늘은?', '내일 서울 날씨 어떠니?' 등과 같이 짧은 문장으로 질문하거나, '병원' 관련 서비스를 받고자 할 때에도 '병원 언제 시작해?', '안과는 어느 건물에 있는지 알려줘'와 같이 불과 3~5개의 토큰으로 이루어진 문장들이 많다. 실험 데이터 전체에 대한 평균 토큰 개수는 3.8289 였으며, Fig. 2에서 볼 수 있듯이 대부분의 문장들이 3~5개의 토큰들로 이루어져있는 것을 관찰할 수 있었다. 물론, 한국어 데이터에 대한 전처리 과정, 이를테면 형태소 분석 및 Part-Of-Speech (POS) 태깅 등의 전처리 과정을 거치게 되면 문장을 구성하는 요소의 개수가 더 많아지게 되겠으나, 본 연구에서는 문장의 구성요소를 토큰이라고 가정하였다. 토큰 단위로 문장 분류를 수행할 경우, 특정 형태소 분석기에 대한 의존성이 사라지고, 형태소 분석 오류에 의한 성능 하락 문제로부터 자유롭다는 장점을 가지고 있다.

Table 1. Data samples used for experiments

Domain	Sentence (data instance)
hospital	병원 언제 업무 시작해? (When does the hospital begin?)
hospital	충남대병원 오픈시간 (When does the Chungnam university hispital begin?)
hospital	안과는 어딨냐 (Where is ophthalmic clinic?)
emergency	불이야! (Fire!)
emergency	살려줘 여기 강도야 (Help, thief)
emergency	지진이다! (Earthquake!)
transport	전철 언제 와? (When will train come?)
transport	반석역 지하철 마지막 언제야? (When the last train arrives at the Banseok station?)
transport	판암역 지하철 마지막차 시간 궁금해 (When the last train arrives at the Panam station?)

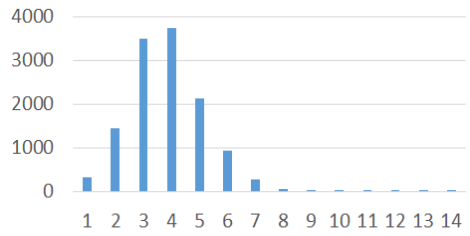

Fig. 2. Distribution of the sentence length (i.e., the number of tokens in sentences)

3.2 Convolutional Neural Network

본 연구에서는 도메인 분류를 위해 딥러닝 기법에 속하는 CNN 모델을 적용하였는데, 이론적으로는 모델의 층 개수가 많아질수록 더 좋은 성능을 얻을 수 있다고 알려져 있으나 실제로는 성격이 다른 데이터에 대한 최적의 모델 구조는 달라지기 때문에 데이터 성격을 분석하여 모델 구조를 설계해야 할 필요가 있다.

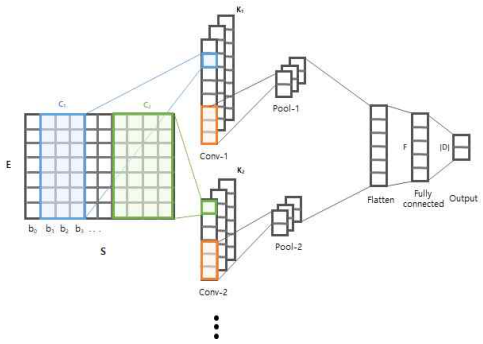

Fig. 3. CNN structure for domain classification

대화시스템에 입력되는 짧은 텍스트 분류를 위해 본 연구에서 설계한 CNN 구조는 Fig. 3 와 같다. 입력되는 텍스트 문장 $S = [b_0, b_1, ..., b_{|S|}]$ 에 대하여, embedding layer (Emb)를 통해 각 토큰 b_i $(0 \le i \le |S|)$이 E-dimensional vector 로 변환된다. 이렇게 단어 임베딩을 통해 변환된 $E \times S$ 행렬은 각 토큰의 의미적 특징을 담게 된다. 고려하고자 하는 convolution size (width)들을 $C = \{C_1, C_2, C_j, ..., C_{|C|}\}$ 라고 하였을 때, convolution width C_j 만큼 인접한 $S \times C_j$ 행렬이 convolutional filter를 거치면서 C_j 크기만큼의 위치 혹은 구조적 특징을 담은 값을 생성한다. 이 convolutional filter가 어떠한 특징을 잡아낼지는 데이터를 통해 학습되며, 임의의 특징 조합을 잡아낼 수 있기 때문에 convolution width C_j 에 대해 K_j 개의 convolutional filter를 거치도록 함으로써 보다 다양한 특징 조합을 잡아낼 수 있게 된다. 이렇게 추출된 결과를 feature map 이라고 부르며, 각 feature map 에 대하여 max-pooling layer를 적용함으로써 가장 유망한 값들을 추려낸다. 이 유망한 값들은 결과값에 긍정적인 영향을 미치는 값들에 대하여 임의의 threshold 값 (물론, 데이터로부터 학습되는 값)을 통해 추려진 것으로 해석할 수 있다.

Max-pooling layer로부터 생성된 유망한 값들을 1차원 벡터 형태로 이어붙이는 flattening을 수행한 후, F-dimensional Fully-Connected (FC) layer에 전달된다. Flattenning 결과물과 FC layer 사이는 bi-partite connection으로써 연결되므로, max-pooling layer로부터 생성된 유망한 값들에 대하여 고차원 패턴을 잡아주는 역할을 수행한다고 볼 수 있다. 이렇게 생성된 고차원 패턴 값들은 output layer에 bi-partite connection으로써 연결된다. Output layer의 각 노드 o_k는 도메인 모음 $D = \{D_1, D_2, D_k, ..., D_{|D|}\}$의 각 도메인 D_k에 해당한다.

본 연구에서 설계한 CNN 모델은 대화시스템에 입력되는 짧은 문장으로부터 embedding layer, convolutional layer를 통해 의미적 특징, 구조적 특징을 추출하고, max-pooling layer, FC layer를 통해 가장 유망한 특징에 집중하여 보다 고차원 패턴을 잡아내고자 설계되었다.

3.3 Random Forest

학습 데이터 일부로부터 생성되는 각 의사결정트리는 서로 다른 feature 조합에서 찾을 수 있는 패턴을 모델링하게 되며, 이들의 모음인 RF 모델은 다양한 패턴을 고려하는 robust한 모델이 된다.

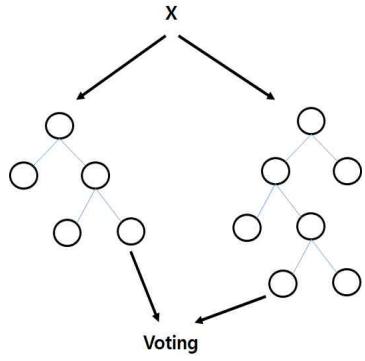

Fig. 4. Random Forest

학습이 완료된 RF 모델에 대하여 새로운 임의의 데이터 X가 입력될 경우, Fig. 4와 같이 각 의사결정트리로부터 결과를 생성하게 되며 이 결과들에 대하여 voting을 통해 최종 결과를 결정하게 된다.

RF 모델의 구조는 본 연구에서 설계한 CNN 모델의 구조와 역할 면에서 유사한 점을 볼 수 있는데, 학습을 통해 생성된 각 의사결정트리는 마치 CNN 모델에서 학습을 통해 생성되는 convolutional filter와 역할이 비슷하다고 볼 수 있으며, 의사결정트리의 결과물로부터 voting하는 과정은 본 연구에서 설계한 CNN 모델의 pooling layer부터 output layer까지의 역할과 비슷하다고 볼 수 있다. 다시 말하면, CNN 모델이 가진 여러 개의 convolutional filter는 임의의 feature 모음을 고려함으로써 서로 다른 패턴을 모델링하는 RF 모델의 각 의사결정트리와 그 역할이 같다고 볼 수 있으며, pooling layer부터 output layer까지의 과정은 여러 의사결정트리의 결과물 중에서 유망한 결과를 결정하는 voting 과정과 역할이 비슷하다고 볼 수 있는 것이다. 하지만, RF 모델의 각 의사결정트리는 데이터로부터 샘플링된 일부를 사용하도록 설계된 반면, CNN 모델의 모든 convolutional filter는 모두 같은 데이터로부터 학습된다는 점에서 차이가 있다.

4. Experiment

4.1 Data and Parameter Setting

실험에 사용된 데이터는 총 7개의 도메인에 대한 문장들이며, 각 도메인 별 문장 개수는 Table 2와 같다. Alarm, transport, weather 도메인에 비하여 bible, campus_admission 도메인은 데이터 개수가 매우 작은 data imbalance 문제가 존재함을 알 수 있다.

Table 2. Data statistics

Domain	Number of sentences
alarm	2729
bible	197
campus_admission	258
emergency	1014
hospital	832
transport	2633
weather	4792

본 연구에서 다루는 대화시스템에 입력되는 문장은 대부분 3~5개의 토큰으로 이루어져있다는 점에 착안하여 CNN 모델의 convolution size C = {3, 4, 5}로 결정하였으며, grid searching을 통해 각 convolution size별로 256개의 convolutional filter가 존재하는 구조를 취하였다. 문장의 토큰 최대길이를 감안하여 |S| = 15로 결정하였으며, 길이가 15 미만인 문장의 뒷부분은 패딩하였다. FC layer의 dimension은 100으로 하였고 모델 regularization 효과가 있다고 알려진 drop-out 기법을 keep probability 0.1로 적용하였다[19]. Embedding dimension E는 1024로 설정하였다. Convolutional layer의 weight 값은 normal(0, 0.05)으로 초기화하고 bias는 0으로 초기화하였으며, embedding layer는 uniform(-0.05~0.05)로 초기화하였다. FC layer는 Xavier uniform initialization으로 초기화하고 bias는 0으로 초기화하였다[20]. 학습을 위해 Adam 알고리즘(초기학습률 0.0001)을 사용하였으며, data imbalance 문제 해결을 위해 cost-sensitive learning을 적용하였다[21]. 학습 배치 크기는 100, 총 epoch 횟수는 20으로 설정하였다.

RF 모델을 구성하는 의사결정트리 개수는 100개로 설정하였으며, 각 의사결정트리 생성을 위해 샘플링되는 데이터의 비율은 100%로 하였다. 각 의사결정트리 생성에 고려되는 feature 개수는 log(트리개수)+1로 설정하였으며, 각 의사결정트리의 깊이 제한은 두지 않았다.

4.2 Result

CNN 모델, RF 모델 외에 baseline으로 Decision Tree (DT)도 실험하였으며, 결과는 Fig. 5, Fig. 6, Fig. 7에 요약되어있다. 각 성능 수치는 precision, recall, F1 score이며 각 도메인의 데이터를 고르게 분포시킨 10-fold cross validation을 통해 측정되었다.

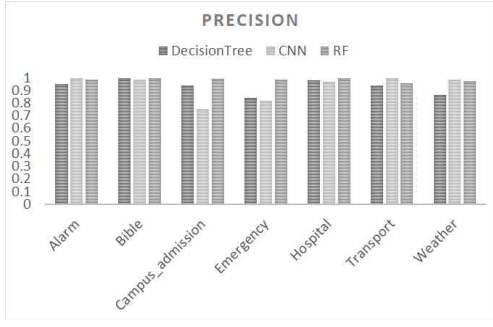

Fig. 5. Precision of comparable models

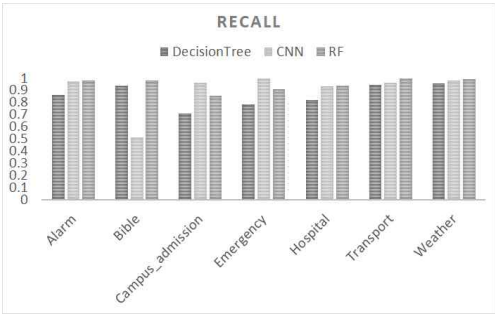

Fig. 6. Recall of comparable models

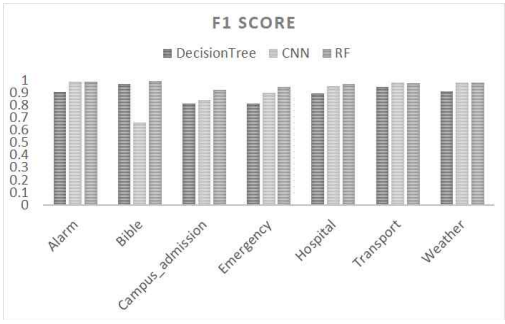

Fig. 7. F1 score of comparable models

DT 모델과 CNN 모델은 도메인이 따른 성능이 0.2 ~ 0.3 정도 차이가 났는데, cost-sensitive learning을 적용하였음에도 불구하고 비교적 데이터 개수가 적은 도메인에서 성능이 낮게 나온 것을 볼 수 있다. 반면 RF 모델은 도메인에 따른 성능 차이가 0.1 안쪽에 그쳤다. RF 모델은 임의 feature를 고려하는 의사결정트리 생성 과정에서 데이터 불균형 문제가 발생하는 도메인의 데이터에 대해서도 효과적으로 학습되었다고 해석할 수 있다. 각 도메인의 데이터 개수에 따라서 weighted score를 계산한 결과는 Table 3과 같다.

Table 3. Weighted performance of comparable models

Model	Weighted precision	Weighted recall	Weighted F1 score
Decision Tree	0.909	0.906	0.905
Convolutional neural network	0.970	0.965	0.965
Random Forest	0.977	0.976	0.976

CNN 모델의 성능이 DT 모델에 비교했을 때 매우 높은 성능인 것은 사실이지만, RF 모델이 전반적으로 CNN 모델 이상의 높은 성능을 달성하는 것을 관측할 수 있었다. 뿐만 아니라, 데이터 불균형 문제가 발생하는 도메인인 bible, campus_admission 등의 도메인들에 대한 성능도 타 도메인과 큰 차이 없이 높은 성능을 달성한 것을 확인할 수 있었다.

5. Conclusion

대화시스템의 자연어 이해 모듈에서 도메인 분류는 매우 중요하다. 대화시스템에 입력되는 문장은 대부분 3~5개의 토큰으로 이루어진 짧은 문장인데, 이러한 짧은 문장에 대한 도메인 분류를 위해 기계학습 모델을 적용하였다. 직접 구축한 데이터에 딥러닝 기법인 CNN 모델, Random Forest (RF) 모델 등을 적용하여 실험하였으며, RF 모델이 가장 좋은 성능을 보였다. 추후, 본 연구에서 설계한 CNN 모델과 다른 구조의 CNN 모델을 추가하여 실험하여 더 높은 성능을 가진 도메인 분류 모델 개발을 지속할 계획이다.

REFERENCES

[1] Amazon Alexa. https://developer.amazon.com/alexa
[2] Naver Clova. https://clova.ai/ko
[3] Samsung Bixby, https://www.samsung.com/sec/apps/bixby/
[4] Y. S. Jeong. (2018). Out-Of-Domain Detection Using Hierarchical Dirichlet Process. *Journal of The Korea Society of Computer and Information, 23(1),* 17-24.
[5] W. S. McCulloch & W. Pitts. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *The bulletin of mathematical biophysic, 5(4),* 115-133.
[6] A. Krizhevsky., I. Sutskever. & G. E. Hinton. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems.* (pp. 1097-1105).
[7] K. Simonyan. & Andrew Zisserman. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *Proceedings of 3rd International Conference on Learning Representations.* (pp. 1-14).
[8] K. He., X. Zhang., S. Ren. & J. Sun. (2016). Deep Residual Learning for Image Recognition. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* (pp. 770-778).
[9] C. Szegedy., W. Liu., Y. Jia., P. Sermanet., S. Reed., D. Anguelov., D. Erhan., V. Vanhoucke. & A. Rabinovich.

(2015). Going Deeper with Convolutions. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* (pp. 1-9).

[10] Y. Kim. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.* (pp. 1746-1751).

[11] H. Kim. & Y. S. Jeong. (2019). Sentiment Classification Using Convolutional Neural Networks. *Applied Science*, 9(11), 1-14.

[12] S. Baker., A. Korhonen. & S. Pyysalo. (2016). Cancer Hallmark Text Classification Using Convolutional Neural Networks. *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining.* (pp. 1-9).

[13] S. Lai., L. Xu., K. Liu. & J. Zhao. (2015). Recurrent Convolutional Neural Networks for Text Classification. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence.* (pp. 2267-2273).

[14] A. Jacovi., O. S. Shalom. & Y. Goldberg. (2018). Understanding Convolutional Neural Networks for Text Classification, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP.* (pp. 56-65).

[15] L. Breiman. (2001). Random Forests. *Machine Learning, 45(1),* 5-32.

[16] J. R. Quilan. (1986). Induction of Decision Trees. *Machine Learning*, 1(1), 81-106.

[17] B. Xu., X. Guo., Y. Ye. & J. Cheng. (2012). An Improved Random Forest Classifier for Text Categorization. *Journal of Computers*, 7(12), 2913-2920.

[18] A. Bouaziz., C. Dartigues-Pallez., C. da C. Pereira., F. Precioso. & P. Lloret. (2014) Short Text Classification Using Semantic Random Forest. *Proceedings of International Conference on Data Warehousing and Knowledge Discovery.* (pp. 288-299).

[19] N. Srivastava., G. Hinton., A. Krizhevsky., I. Sutskever. & R. Salakhutdinov. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929-1958.

[20] X. Glorot. & Y. Bengio. (2010). Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics.* (pp. 249-256).

[21] D. P. Kingma. & J. L. Ba. (2015). Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations.* (pp. 1-15).

정 영 섭(Young-Seob Jeong) **[정회원]**

· 2016년 2월 : 한국과학기술원 전산학과 (공학박사)
· 2016년 2월 ~ 2016년 12월 : Naver
· 2017년 1월 ~ 현재 : 순천향대학교 빅데이터공학과 교수

· 관심분야 : 인공지능, 빅데이터
· E-Mail : bytecell@sch.ac.kr