chapter 1

인공지능과 자연어 처리 기술 동향

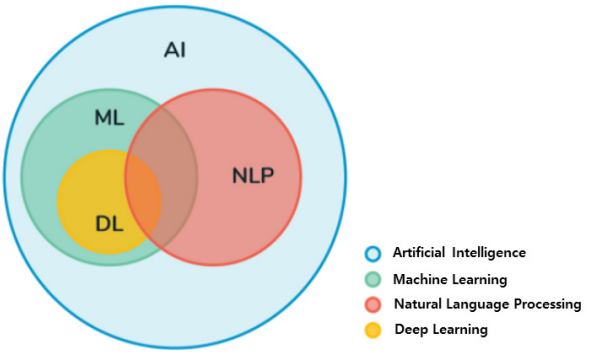
유승의 ‖ 동아대학교 스마트거버넌스연구센터 전임연구원

우리 사회는 최근 몇 년간 어느 때보다 빠른 정보통신 기술의 발전으로 급격한 사회적 변화를 맞이하고 있다. 4차 산업혁명에서 촉발된 인공지능(Artificial Intelligence: AI) 기술이 비약적으로 발전함에 따라 사회적 · 국가적인 차원에서 관심이 증가하고 있으며 관련 연구도 크게 증가하고 있다. 본 고에서는 머신러닝 · 딥러닝과 같은 인공지능 기술을 통해 이루어지는 자연어에 적용되어 실행되고 있는 기술들에 대하여 소개하고, 자연어 처리 단계에서 강조되고 있는 텍스트 임베딩(Embedding) 기술들을 소개함으로써 인공지능을 활용한 자연어(텍스트) 처리에 대한 이해를 돕고자 한다.

I. 인공지능과 언어처리 기술

인공지능 기술이 발전하고 있는 가운데, 이를 적극적으로 활용 및 상용화하고 있는 분야는 “시각이해 기술”과 “언어이해 기술”이 대표적이다[1]. 또한, 이 두 가지 기술은 빅데이터(Big Data)와 결합되어 빠르게 발전되고 있다. 시각이해 기술은 인공지능을 활용하여 이미지나 영상 분석, 객체 인식, 속성 분석을 통해 이미지에 포함된 정보를 추출하는 기술이다. 언어이해 기술은 사람들이 표현하는 방대한 텍스트로부터 의미를 이해하고 텍스트에 포함된 정보를 추출 및 분류하며, 더 나아가 직접 텍스트를 생성하는 기술을 포함한

* 본 내용은 유승의 전임연구원(☎ 051-200-4795, juim0928@donga.ac.kr)에게 문의하시기 바랍니다.
** 본 내용은 필자의 주관적인 의견이며 IITP의 공식적인 입장이 아님을 밝힙니다.
***이 논문은 2018년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임(NRF-2018S1A3A2075240)



〈자료〉 구글 이미지

[그림 1] 인공지능 활용의 기술적 영역

자연어 처리(Natural Language Processing: NLP)로 대표된다.

인공지능이라는 큰 틀 안에 기계학습(Machine Learning: ML)이 있고, 기계학습 안에 딥러닝(Deep Learning: DL)이 있다. 그리고 NLP라고 불리는 자연어 처리는 기계학습과 딥러닝을 교집합으로 가지고 있는 영역이 있다.

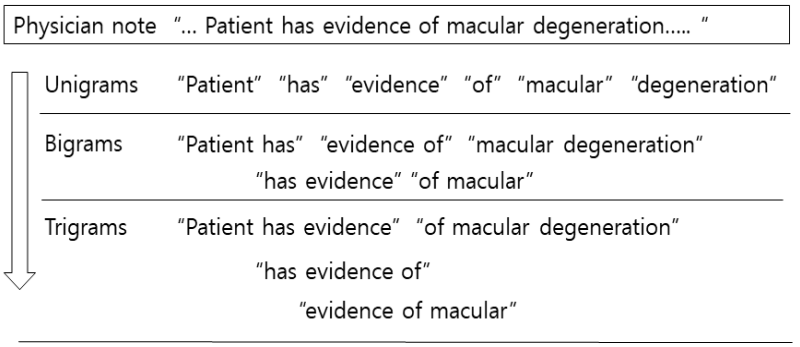
NLP는 기계가 사람의 언어에 대해 처리하는 계산적 기술(Computational Technicques)의 집합이라고 할 수 있다[2]. 이러한 NLP의 세부 분야로는 감성분석 또는 감정분석, 의미분석, 구문분석, 음성인식(질의응답) 등이 있다. 이러한 계산적 기술 분류를 구분해 보면 아래와 같이 구분할 수 있다.

1. 워드 클라우드

워드 클라우드(Word Cloud)는 텍스트를 분석하여 사람들의 관심사, 키워드, 개념 등을 파악할 수 있도록 빈도수를 단순히 카운트하여 시각화시킨 것이다. 즉, 자연어를 컴퓨터 또는 워드 클라우드 생성기를 통해 처리하는 시각화 툴(tool)이다[3].

2. N-gram Model

N-gram은 카운트 기반의 통계적 방법을 사용하고 있다. 이러한 N-gram에는 Uni-gram, Bi-gram, Tri-gram 등 하나의 단어에 이어 어떤 단어가 출현하는가?라는 것에



〈자료〉 동아대학교 스마트거버넌스연구센터 자체 작성

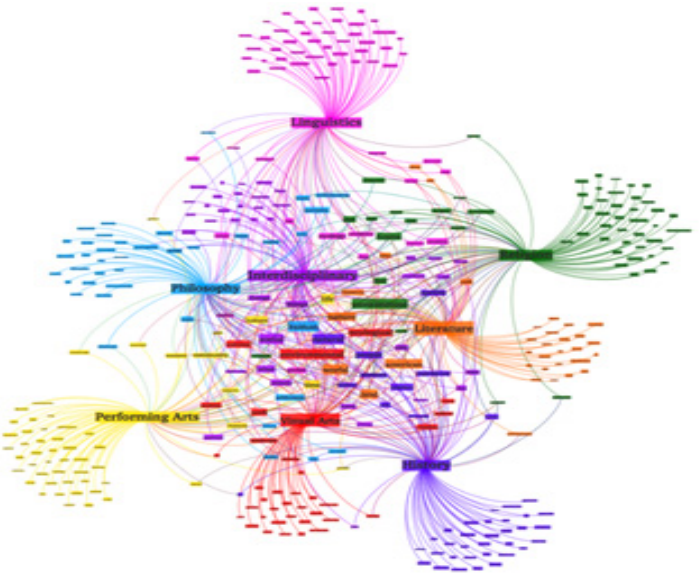
[그림 2] N-gram model의 학습 단계

착안한 확률적 계산의 언어 모델이다.

Bigram은 동시에 연속적인 2개의 단어를 분석하고, Trigram은 3개의 연속적인 단어를 보는 모델이다. 따라서 사용자가 분석하고자 하는 단어를 몇 개 선택할 것인가에 따라 N-gram의 N이 가지는 의미는 달라진다. 하지만 NLP가 계산적 기술의 집합이라고 불리는 것에서 알 수 있듯이, 컴퓨터는 인간의 언어를 단순한 수학적 계산을 통해 처리하는 것이다. 예를 들어, 기존의 텍스트 데이터들을 통해 p("Today is Monday")라는 확률이 0.001이라고 한다면 p("Today Monday is")는 0.000000001이 될 것이다. 즉, 단어의 출현이 적거나 표현되는 순서가 희박할 경우에 계산 결과의 확률은 낮아진다. 이러한 과정은 마치 기계가 인간의 언어를 이해하고 단어를 예측하는 것처럼 보이는 것이다.

3. 토픽모델링

토픽모델링(Topic Modeling)이란 단어 또는 말뭉치(corpus)로부터 숨겨진 주제를 찾고 키워드별로 주제를 묶어 주는 비지도 학습 및 확률 알고리즘이다. 이러한 토픽모델링의 접근법은 다양하다. 대표적인 기법으로 Latent Dirichlet Allocation(LDA)이 있으며, 사람들의 관심사와 관련된 '토픽'이 무엇인지 찾아낼 수 있도록 하는 접근법이다. LDA는 문서에 대한 확률 분포를 가정한다는 점에서는 "나이브 베이즈 분류"(Naive Bayes Classification)와 비슷하며 이론적 기초를 "베이즈 추론"에서 발전시켰다.



〈자료〉 구글 이미지

[그림 3] 토픽모델링 단어 추출과 시각화

II. 자연어 처리 임베딩 기술

자연어 처리를 위해서는 텍스트를 컴퓨터가 이해할 수 있도록 숫자로 바꾸는 작업이 필요하다. 사람은 문장에서 단어가 쓰인 의미를 문맥을 통해 구분할 수 있지만, 기계가 이해할 수 있도록 단어를 0과 1의 수치로 표현하는 방법을 “벡터화(Vectorization) 또는 임베딩(Embedding)”이라고 한다[4]. 임베딩은 전체 단어들 간의 관계에 맞춰 해당 단어의 특성을 갖는 벡터로 바꿔주므로 단어들 사이의 유사도를 계산하는 기법이다. 이러한 유사도 계산을 통해 단어 간의 의미적 · 문법적 관계를 파악해낼 수 있다. 예를 들어, “아들-딸” 사이의 관계와 “소년-소녀” 사이의 의미 차이가 임베딩에 함축되어 있으면 좋은 임베딩이라 할 수 있다.

임베딩 기법의 발전 흐름과 종류는 통계적 기반과 뉴럴 네트워크 기반으로 나눌 수 있고 단어수준과 문장수준의 임베딩 기법으로 구분할 수 있다.

1. 통계적 기반

임베딩 초기 기법은 통계적 기반을 중심으로 말뭉치라 불리는 코퍼스(Corpus)의 통계량을 직접적으로 활용하였다. 대표적인 잠재 의미 분석(Latent Semantic Analysis: LSA)은 단어 사용 빈도 등 코퍼스의 통계량 정보가 들어 있는 행렬에 특이값 분해 등 수학적 기법을 적용해 행렬에 속한 벡터들의 차원을 축소하는 방법이다[3]. 여기서 차원 축소를 통해 얻은 행렬을 기존의 행렬과 비교했을 때 단어를 기준으로 했다면 단어 수준 임베딩, 문서를 기준으로 했다면 문서 임베딩이 된다. 이러한 잠재 의미 분석 수행 대상 행렬에는 Term-Document, TF-IDF, One-hot Encoding 등이 있다.

가. TDM

TDM(Term-Document Matrix)은 단어-문서행렬이라고 부르며 문서에서 등장하는 단어들의 빈도를 행렬로 표현하는 것이다. 이것은 BoW(Bag-of-Word)의 표현을 행렬로 표현하는 것이다. 이러한 방법은 문서로부터 수치화된 단어들을 서로 비교할 수 있다는 장점이 있는 반면 고려해야 할 단어 수가 대량일수록 적용하는데 한계가 있는 단점이 있다.

나. TF-IDF

TF-IDF(Term Frequency-Inverse Document Frequency)는 특정 단어가 문서 내에서 출현하는 빈도(TF)값과 흔한 단어는 문서에서 자주 등장되는 경우가 많아 역빈도(IDF)값을 계산하는 것이다.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF
Term x within document y

$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

〈자료〉 구글 이미지

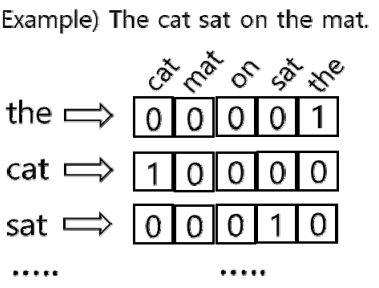
[그림 4] TF-IDF 공식

이는 문서에서 특정 단어가 얼마나 중요한 역할을 하는 것인지를 나타내는 통계적 수치이다[5]. 일반적으로 TF-IDF값이 높은 단어일수록 문서에서 중요도가 높다고 간주한다. 이러한 방법은 문서의 핵심어 추출, 검색 결과의 우선순위 결정 등에 이용된다.

다. One-hot Encoding

원-핫 인코딩은 문자를 숫자로 표현하는 가장 기본적인 방법으로 머신러닝과 딥러닝

학습을 위해서는 반드시 짚고 넘어가야 되는 표현 방법이다. Count Vector와 유사한 개념으로 문서의 단어를 벡터로 표현하는 방식으로 '0'과 '1'로 구분하는 방법이다[6]. 이 방법은 문자(characters) 또는 단어를 기준으로 벡터화할 수 있고 원-핫 인코딩의 차원은 말뭉치 내 단어의 수와 같다. 원-핫 인코딩을 이용한 단어 벡터화의 예는 [그림 5]와 같다.



〈자료〉 동아대학교 스마트거버넌스연구센터

[그림 5] One-hot Encoding을 이용한 단어의 벡터화

이러한 원-핫 인코딩은 단어 간의 관계에서 단어들 간의 유사성과 반대적 의미에 대해서는 전혀 반영하지 못하고 문장의 횟수에만 의존한다는 단점이 존재한다.

[표 1] 원-핫 인코딩의 장단점

구분	내용
장점	- 텍스트를 유의미한 숫자(벡터)로 바꾸는 가장 손쉬운 방법론
단점	- 단어 갯수가 늘어날수록 벡터 저장 공간이 늘어나야 함 - 단어의 문맥정보가 사라짐(단어의 word order와 co-occurrence 사라짐) - 단어 간 유사도를 파악할 수 없고 유사성에서 반대되는 의미를 반영하지 못함

〈자료〉 동아대학교 스마트거버넌스연구센터 자체 작성

2. 뉴럴 네트워크 기반 기법

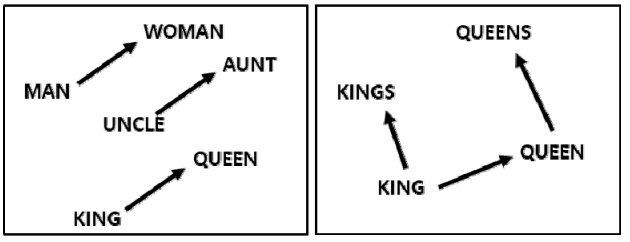
워드 임베딩의 역사는 인공 망을 이용하여 주변 단어의 단어 등장 확률을 예측한 Neural Probabilistic Language Model(NPLM)이 발표된 이후부터 Word2Vec→FastText→ELMO→BERT 기법으로 발전하고 있다. 가장 최신의 언어분석 기법인 BERT는 다른 언어분석 기법들에 비해 임베딩 결과에서 우수한 성능을 보이고 있다. 이는 기존의 임베딩은 문장에서 단어를 순차적으로 입력받고 다음 단어를 예측하는 일방향(uni-directional)이지만 BERT는 문장 전체를 입력받고 단어를 예측하고 양방향(bi- directional) 학습이 가능하기 때문이다[5]. 이러한 Neural Network 구조의 유연성과 풍부한 표현력으로 자연어의 문맥을 상당 부분 학습할 수 있고 높은 정확도를 보이고 있다.

3. 단어 수준의 임베딩 기법

단어 수준의 임베딩은 신경망을 이용하여 텍스트를 변환하는 것이 가장 큰 특징으로 단어가 주어지면 그 단어와 주변 단어가 동시에 일어날 확률을 구하므로 단어의 의미를 수치화할 수 있다. 임베딩 기술은 2017년 이전까지 대부분 단어 수준의 모델로 개발되어 졌다. 단어 수준의 벡터 표현은 텍스트를 수치화한 벡터 형태로 표현하는 것이다. 이는 비슷한 의미를 가진 단어는 크기와 방향에 유사성을 가지는 경향이 있을 것이라는 가정이 핵심이다. 이러한 단어 임베딩은 문장의 유사도를 나타내는데 효율적이라는 사실이 [3]에 의해 증명되어 자연어 처리를 위한 딥러닝 모델적용 시 첫 번째 데이터 처리 레이어에서 자주 활용된다. 단어수준의 임베딩 기법에는 Word2Vec, GloVe, FastText 등이 있다.

가. Word2Vec

앞서 설명에서 원-핫 인코딩은 단어 간 유사도를 파악할 수 없다는 단점을 언급하였다. 이러한 유사도는 단순한 수치뿐만 아니라 의미적 자질을 내포하고 있다. 의미적인 성질이 유사한 단어들은 벡터 공간상에서 “유클리디안(Eucledian) 거리”나 “코사인 유사도(cosine similarity) 거리”가 가까운 벡터들로 표현된다. 이렇게 단어 간 유사도를 반영하고 단어를 벡터화할 수 있는 방법으로 [7]이 제안한 word2vec이 있다.



〈자료〉 구글 이미지

[그림 6] Word2vec Architectures

word2vec의 가장 큰 개념은 “비슷한 분포를 가진 단어이면 가까운 벡터로 표현된다.” 이다. 따라서 이는 학습속도가 빠르며 단어의 맥락을 고려하므로 단어의 의미를 잘 파악한다고 알려져 있다. 이러한 word2vec은 CBOW(Continuous Bag of Words)와 skip-gram 두 가지 모델로 분류된다[7].

CBOW는 특정 단어가 주어졌을 때 앞과 뒤에 붙어있는 단어를 통해 주어진 단어를 유추하는 방법이다. Skip-gram은 CBOW와 반대로 중심단어에서 주변단어를 예측하는 방법으로 중심단어와 연관된 두 가지 이상의 의미론적 벡터를 찾을 수 있다는 장점이 있다. 하지만 문장에서 단어의 출현이 많다고 그 단어가 중요한 의미를 가진다고 볼 수는 없다. 이것은 단어의 빈도수가 높다는 이유로 그 단어의 중요도가 높아진다고 할 수 없기 때문이다.

나. FastText

이 방법은 단어를 개별 단어가 아닌 n-gram의 characters(Bag-Of-Characters)를 적용하여 임베딩하므로 하나의 단어를 여러 개로 잘라서 벡터로 계산하는 방식이다. 예를 들어, where를 Trigram의 characters로 표현하면 〈'wh', 'whe', 'her', 'ere', 're'〉로 FastText는 표현된다. 최종적으로 각 단어는 임베딩된 n-gram의 합으로 표현되고, 빠르고 좋은 성능을 나타내었다.

이러한 FastText는 Word2vec과 동일한 데이터 양을 사용하더라도 더 많은 정보를 학습하기 때문에, Word2vec에 비해 높은 성능을 낼 수 있다[8]. 또한, 기존의 Word2vec의 한계점으로 여겨진 OOV(Out of Vocabulary)에 대한 임베딩까지 가능하게 해 준다. 예를 들어, 데이터 학습 시 'subsequent'라는 단어의 경우, FastText를 사용할 경우 'sub'와 'sequent'라는 n-gram을 학습하였다면, 두 단어의 임베딩 벡터 조합으로 임베딩 벡터를 생성할 수 있기 때문에 문장에서 자주 등장하지 않는 단어를 파악할 수 있고 Word2vec에 비해 보다 우수한 성능을 보이고 있다.

다. ELMo

ELMO(Embeddings from Language Model)는 2018년에 제안된 새로운 워드 임베딩 방법론으로 "언어 모델로 하는 임베딩"이라 해석된다. ELMO의 특징은 사전 훈련된 언어 모델(Pre-trained Language Model)을 사용한다는 점이다. 또한, 다른 특징은 양방향 언어 모델(Bi-directional Language Model: BiLM)을 적용하여 문맥을 반영한 워드 임베딩 기법이다[9]. 예를 들어, Bank라는 단어를 학습할 때. '은행계좌'라는 Bank Account와 '강둑'이라는 River Bank에서 'Bank'는 다른 의미를 가지는데, Word2Vec에서는 이를 제대로 반영하지 못한다는 단점이 있다. Word2Vec은 Bank란 단어를 임베딩하면,

Bank Account와 River Bank에서의 Bank는 전혀 다른 의미이지만 두 가지 상황 모두에서 같은 벡터가 사용된다는 한계점이 있다[10]. 이러한 한계점을 ELMO는 BiLM의 사전훈련으로 극복할 수 있다[11]. 또한, 이 특징은 NLP에서 Transfer Learning이 확산되는 계기가 되어 지금의 BERT가 출현하게 되었다.

[표 2] 단어수준 임베딩 기법의 장/단점

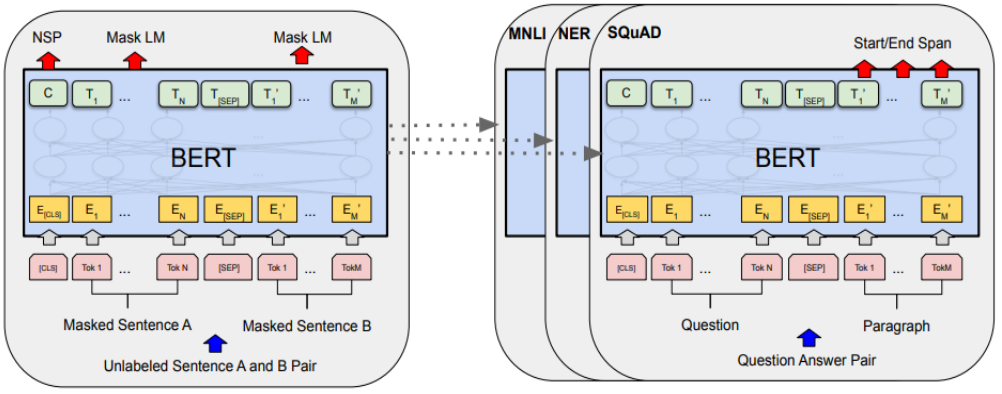
구분	내용
장점	- 단어의 벡터 간에는 사칙연산이 이용되어 단어 간 의미의 합과 차이를 반영 - 단어의 차원을 사용자가 지정한 개수의 차원으로 표현
단점	- 다른 단어를 사용하더라도 단어형태가 같으면 동일한 단어의 벡터로 전달되어 동음 이의어(homonym) 구분이 용이하지 않음

〈자료〉 동아대학교 스마트거버넌스연구센터 자체 작성

4. 문장 수준의 임베딩 기법

문장 수준의 임베딩은 2018년 초에 ELMo(Embedding from Language Models)가 발표된 이후 주목받기 시작했다. 이는 개별 단어가 아닌 단어 Sequence 전체의 문맥적 의미를 함축하기 때문에 단어 임베딩 기법보다 Transfer Learning 효과가 좋은 것으로 알려져 있다. 또한, 단어 수준 임베딩의 단점인 동음이의어도 문장수준 임베딩 기법을 사용하면 분리해서 이해할 수 있다. 문장 수준의 임베딩 기법에는 BERT, GPT 등이 있다.

BERT(Bidirectional Encoder Representations from Transformer)는 2018년 구글의 Jacob Devlin과 그의 동료가 함께 만들었다[10]. 이 모델은 최근까지 딥러닝 모델을 적용한 모든 자연어 처리 분야에서 좋은 성능을 보이고 있는 범용 언어 모델이다. BERT는 사전학습(pre-trained) 모델로서, 특정 과제(task)를 하기 전 사전훈련 임베딩을 실시하므로 기존의 임베딩 기술보다 과제의 성능을 더욱 향상시킬 수 있는 모델로 관심받고 있다. BERT를 적용한 모델링 과정을 살펴보면 Pre-trained는 비지도 학습(Unsupervised Learning) 방식으로 진행되고 대량의 코퍼스를 인코드(Encoder)가 임베딩하고, 이를 트렌스퍼(Transfer)하여 Fine-tuning을 통해 목적에 맞는 학습을 수행하여 과업을 수행하는 것이 특징이다[11]. 또 다른 BERT의 특징은 양방향 모델을 적용하여 문장의 앞과 뒤의 문맥을 고려하는 것으로 이전보다 더 높은 정확도를 나타낸다. BERT의 활용은 대량의 텍스트 데이터와 다양한 언어를 적용할 수 있다는 장점 때문에, 연구자들 사이에서 가장



〈자료〉 구글 이미지

[그림 7] 언어모델링(Pre-training), NLP Task(Fine-tuning)

각광 받는 기술 중 하나이다.

III. 결론

최근 인공지능을 적용한 자연어 처리 기술이 빠르게 발전하고 있다. 하지만 자연어를 이해하고 실생활에 적용하는데 많은 난제들이 존재하고 있다. 자연어 처리를 할 수 있는 알고리즘 생성과 더 높은 수준의 모델링이나 알고리즘 개발에는 여전히 인간의 지식이 필요하며, 인공지능을 탑재한 자연어 처리 기술은 신문기사를 스스로 생성할 수는 있지만, 창조적이고 감명을 받는 문학 또는 컬럼 등은 여전히 사람이 쓰고 있다. 하지만 인공지능을 활용한 텍스트 분석은 빅데이터와 함께 사회적 · 제도적 문제들을 해결할 수 있는 가능성을 제공하고 있다. 예를 들어, 본 저자가 속해 있는 동아대학교 스마트 거버넌스 연구센터에서는 시민들의 다양한 의견 수집 및 인공지능을 활용한 텍스트 분석을 통해, 시민의 요구사항과 불만을 분석하고 사회에서 발생되는 이슈들을 분석하여 시민의 인식을 조사할 수 있을 뿐 아니라 이를 해결할 수 있는 방안 등을 연구하고 있다. 이러한 자연어 처리 기술은 다양한 분야에 적용될 수 있어 향후 우리들의 삶을 보다 풍요롭게 할 것이라 사료된다.

[참고문헌]

[1] 인공지능신문, “자연어처리(NLP) 기술의 상용화와 그에 따르는 과제”, 2020. 12. 1.
[2] 조재신, “4차산업혁명을 선도하는 유럽의 인공지능(AI) 특허기술”, 한국디지털콘텐츠학회 논문지, 19(10), 2018, 1937-1945.
[3] 서상현, 김준태, “딥러닝 기반 감성분석 연구동향”, 한국멀티미디어학회지, 20(3), 2016, 8-22.
[4] Sohrabi, B., Vanani, I. R. & Shineh, M. B.. “Topic Modeling and Classification of Cyberspace Papers Using Text Mining,” Journal of Cyberspace Studies, 2(1), 2018, 103-125
[5] Wikipedia, “Tf-idf“
[6] Harish, B. S. & Rangan, R. K, “A comprehensive survey on Indian regional language processing,” SN Applied Sciences, 2(7), 2020, 1-16.
[7] Mikolov, T., Corrado, G., Chen, K. & Dean, J. “Efficient estimation of word representations in vector space,” Proc. of the International Conference on Learning Representations, ICLR 2013, 1-12.
[8] Kuyumcu, B., Aksakalli, C., DelilL, S., “An automated new approach in fast text classification(fastText) A case study for Turkish text classification without pre-processing,” Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval, 2019, 1-4.
[9] 박광현, 나승훈, 신종훈, 김영길, “BERT를 이용한 한국어 자연어처리: 개체명 인식, 감성분석, 의존 파싱, 의미역 결정”, 한국정보과학회 학술발표논문집, 2019, 584-586.
[10] Wikipedia, “BERT(language model)”
[11] 이제로, 박은환, 이재구, “BERT파생모델의 한국어에 대한 성능 비교”, 한국통신학회 학술대회논문집, 2020, 901-902.