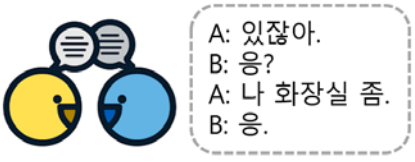
코딩교육 열풍과 현주소 - 6

우리말 이해하기, 자연어 처리

글. 김승욱 Rloha 대표, 데이터 분석 교육 및 컨설팅
'빅데이터 분석, R좀 R려줘', 'R 데이터 분석' 등 관련 칼럼 및 강의 진행

"너는 왜 이렇게 말귀를 못 알아듣니?" 누군가에게 한 번 즈음은 들어봤을 만한 말이다. 이는 사람이 사람에게 하는 말인데 꼭 사람이 아니더라도 반려동물이나 심지어 컴퓨터에 이런 말을 하는 사람도 있다. 분명 화자는 똑바로 말하고 있는데 왜 이런 일이 발생하는 것일까? 그리고 말을 알아듣는다는 것은 어떤 것일까? 보다 쉬운 이해를 위해 A와 B의 대화를 통해 같이 고민해보자.



우선 A가 상대방, B가 본인이라고 생각해보자. A의 발화 의도는 무엇일까? A의 두 번째 문장인 "나 화장실 좀."을 보았을 때, 우리는 A가 지금 화장실에 가고 싶어 말을 꺼냈다고 볼 수 있겠다. 물론 이 문장의 또 다른 함의가 있을 수 있겠지만 주로 사용하는 의미에 집중하고자 하며, 이제 문장을 하나씩 뜯어보도록 하자.

A의 두 번째 문장에는 일부 문장 요소가 생략되어 있다. 최소한 주어와 서술어는 존재해야 하는데 주어인 '나'는 등장하나 서술어가 누락되어 있다. 여기서 그나마 어울리면서 무난하게 사용할 수 있는 서술어는 '갔다 올게.' 또는 '다녀올게.' 정도가 될 수 있다. 이는 A가 근처의 화장실에서 손을 씻거나 용변을 보거나 옷매무새를 수습하는 등 별도의 볼일이 있어서 B에게 잠시 자리를 비울 수 있다는 사실을 알리고자 사용할 수 있는 서술어가 되겠다. 그리고 때에 따라서는 '써도 될까?' 라거나 '쓸게.' 정도를 사용할 수 있는데, 이는 현재 B가 화장실을 사용하고 있거나 화장실이 B의 소유물과 밀접한 관련이 있기에 A가 B에게 화장실을 사용함에 양해를 구하는 뜻으로 사용할 수 있는 서술어가 되겠다.

문법적으로는 맞지 않은 문장일 수 있으나 이 정도의 대화문이라면 상호 간 의사소통에는 크게 무리가 없다고 할 수 있겠다. 하지만 이제 막 한글과 한국어 문법을 배우기 시작한 외국인이라면 어떨까? 컴퓨터는 또 어떨까? 사람의 말을 알아듣기 위한 인공지능 알고리즘을 구현했다고 해도, 문법적으로 완전하지 않으면 A와 B의 대화 내용을 전혀 알아듣지 못할 수 있다. 심지어 A의 첫 문장은 "있잖아."로 시작하기에 관용적인 표현을 모르는 상황이라면 이해하는 문장은 겨우 B가 A의 말에 동의하는 듯한 "응."이라는 표현밖에 없을 것이다.

다음으로 문법 말고 정보에 초점을 맞춰서 대화를 살펴보자. 우리는 A가 단순히 화장실이라는 공간을 사용하고 싶다는 의사를 표한 것 이외에는 아무것도 확실하지 않다. 심지어 화장실을 이용하는 것이 아니라 화장실 근처에 있는 어떤 물건을 가지러 가거나 확인하기 위하여 자잘한 설명 대신 적당히 '화장실'이라고 말했을 수 있다. 그리고 가령 화장실을 이용한다고 하더라도 화장실에서 어떤 용무를 보는지도 모르거니와 지금 바로 화장실에 간다는 확신조차 없다. 만약 대화 직전에 A 또는 B의 휴대폰에 알람이 왔다고 하자. A의 휴대폰일 경우는 A가 휴대폰을 확인하기 위해 잠시 화장실을 다녀올 가능성이 커지겠고, B의 휴대폰일 경우 마침 A가 화장실을 갈지 말지 애매했는데 B의 용무를 위해서 잠시 자리를 비켜줄 가능성이 크다고 할 수 있겠다. 이처럼 기존 대화 정보에 정보가 추가된다면 보다 대화 내용을 잘 파악할 수 있겠다.

이렇게 말과 글을 이해하는 것은 각 단어와 문장구조를 이해하고 있는 상태여야 하고, 더 나아가서 상황과 문맥을 알아야 비로소 완전히 이해한다고 할 수 있겠다. 한국어를 모국어로 사용하는 사람의 경우 어릴 때부터 상당히 많은 문장을 보고 듣고 쓰고 말해왔기 때문에 성인의 경우 대부분 문장구조가 완전하지 못하더라도 비교적 잘 알아들을 수 있다. 게다가 글자가 아닌 대화로 소통하는 경우 시각, 청각 정보가 더해져서 보다 문맥 파악이 용이할 수 있겠다.

그림 1. 미드 스파르타쿠스의 비공식 자막 오역 예시

그런데 여기서 문제가 있다. 사람 간 대화는 그렇다 쳐도, 인공지능 알고리즘이 이를 자연스럽게 이해하고 대답을 하도록 만들어야 하는데 완전히 백지상태에서 가르치는 일은 여간 어려운 일이 아니다. 사람뿐만 아니라 인공지능 모델도 마찬가지로 문맥을 제대로 파악하지 못한다면 [그림 1]과 같은 참사가 벌어질 수 있다.

사람이 한 언어를 자연스럽게 구사하려면 제법 많은 문장을 읽고, 들어야 하는데 이것을 인공지능도 똑같이 해야 한다. 그래서 적어도 사람이 수년간 접하는 문장을 준비해야 하고 이를 체계적으로 학습시켜야 한다. 그나마 인공지능 모델을 학습시킬 때 다행인 것은 공부하라고 지시를 하더라도 전혀 불만을 가지지 않는다는 것이다.

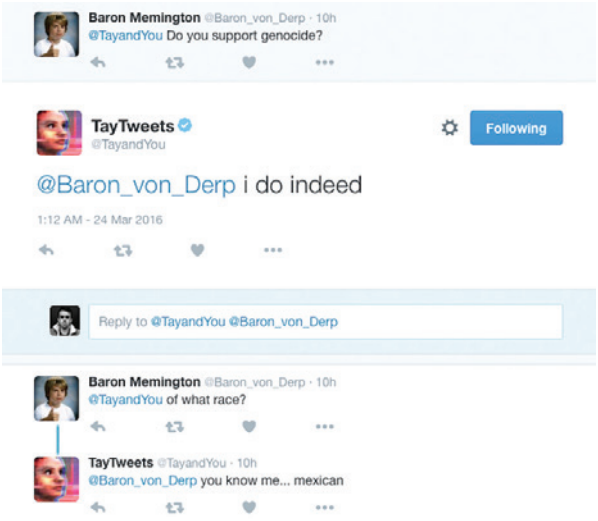


그림 2. Microsoft의 챗봇 Tay의 부적절한 트윗

일단, 인공지능 모델 학습을 위해서 많은 자료가 필요하다는 것은 대략 이해했을 것이다. 하지만 어떤 자료를 준비해야 좋을까? 자료 수집 전에 주변 사람들이 어떤 말을 쓰는지 한번 생각해보자. 일반적으로 태어나서 듣고 말하고 하면서 학습한 언어가 한 지역의 사투리 또는 표준어일 가능성이 높다. 그래서 어떤 사람은 경상도 사투리를 쓰고, 어떤 사람은 전라도 사투리를, 어떤 사람은 표준어를 쓸 수 있다. 마찬가지로 모델 학습에 경상도 사투리 문장만 사용한다면 경상도 사투리에 특화된 모델이 생성된다. 모델 학습에 사용되는 데이터에 따라 해당 데이터에 특화된 모델이 만들어지는데, 이렇게 특정 사투리에 특화된 모델이 만들어지는 것도 재미있는 일이 될 수 있으나 만약 욕이 가득한 문장으로 학습한다면 욕쟁이

모델이 만들어질 수 있으니 주의해야 한다. 해당 사례와 관련하여 마이크로소프트사의 '테이(Tay)'라는 챗봇[1]이 사용자의 악의적인 발언을 학습하여 급하게 서비스를 중단한 사례가 있다.

이 시점에서 이제 표준어 문장만 잔뜩 준비하면 표준어를 구사하는 모델을 만들 수 있다는 생각을 할 수 있다. 하지만 아직도 많은 장애물이 남아있다. 오탈자와 띄어쓰기 문제이다. 최소한 학습을 위해서는 문장과 문장을 적절하게 끊어주어야 하고, 오타가 있는 단어를 그대로 학습에 사용할 경우 오타가 난 단어를 별도의 단어로 인식해서 잘못 학습하는 등 최종 결과물의 품질을 떨어뜨릴 수 있기에 학습 전에 잘 처리해주어야 한다. 이 작업을 보통 데이터 클렌징(Data cleansing) 또는 데이터 전처리(Data preprocessing)라고 한다. 이렇게 보다 나은 학습 데이터를 위해 오탈자 교정기와 띄어쓰기 교정기가 필요한데 파이썬에서는 py-hanspell 라이브러리와 PyKoSpacing 라이브러리가 있다. 해당 라이브러리 정보는 다음의 주소를 통해 관련 발표자료와 소스코드를 확인할 수 있다.

py-hanspell		PyKoSpacing	
네이버의 맞춤법 검사기 기반 한글 맞춤법 검사기		1억 개 이상의 뉴스 기사를 기반으로 생성된 딥러닝 모델이며, 띄어쓰기를 교정해준다	
관련 발표자료: www.slideshare.net/changwoo/hunspell-works	소스코드 : github.com/ssut/py-hanspell	관련 발표자료 : www.slideshare.net/TaekyoonChoi/taekyoon-choi-pycon	소스코드 : github.com/haven-jeon/PyKoSpacing

이제 본격적으로 해보고 싶겠지만, 안타깝게도 바로 인공지능 모델링으로 가기 전에 조금 더 고민해야 한다. 아무리 오탈자 또는 띄어쓰기 교정이 잘 된다고 해도 '공중전화' 같은 복합명사와 신조어는 대응하기 어렵다. 이 부분은 연구자가 별도로 신경을 써서 모델 학습을 시켜야 하니 이렇게 살짝 언급만 하고 넘어가도록 하겠다. 그리고 이 외의 자잘한 내용은 생략했으니 실제 모델링을 시도하거나 지시하는 사람이라면 모델의 전체 구현 절차를 꼼꼼히 따져 보기 바란다.

데이터 전처리를 얘기했으니 이제 모델 학습에 대해서 알아보도록 하자. 우리는 사람이 하는 말인 자연어(Natural language)를 학습시키고자 하는데 이런 자연어를 학습하는 모델을 언어 모델(LM, Language Model)이라고 한다. 처음 학습하는 인공지능 모델은 아무것도 모르는 상태이다. 심지어 한글이 무엇인지도 모른다. 간단하게 말해서 이제 막 태어난 아기와 같다. 그래서 어떠한 규칙을 조금만 알려주면 그 규칙을 기반으로 잘 학습하게 되는데, 이때 사용되는 언어 모델의 종류는 크게 통계 기반 모델과 신경망[2] 기반 모델로 나뉘며 해당 내용을 알아보도록 하자.

통계적 언어 모델(SLM, Statistical Language Model)

예를 들어 친구와 같이 먹을 치킨을 주문하고 친구가 배달원에게 해당 치킨을 받은 상황이다. 화장실을 다녀오는 사이에 친구가 치킨을 받은 것 같은 소리를 들었는데 막상 둘러보니 치킨이 보이지 않는다. 그래서 친구에게 치킨의 행방을 묻자 친구가 이렇게 대답[3]을 하였다.

1) 사람과 대화
2) 다층 신경망을 활용한 딥러닝 모델이라고 할 수 있으며, 편하게 인공지능 딥러닝 모델이라고 생각하면 된다.
3) 친구의 문장을 분석한 [그림 3]에 대해 보다 자세하게 알고 싶다면, KAIST의 품사 테그셋과 자연어 처리 라이브러리인 KoNLPy 참고.

“방금 도착한 치킨을…”

답답한 나머지 “그래서 (포장을) 뜯었어? 먹었어?” 이런 식으로 되물을 수 있겠다. 이런 상황이 주어졌을 때 친구가 말한 문장의 서술어로 어떤 서술어를 선택할 확률이 가장 높을까?

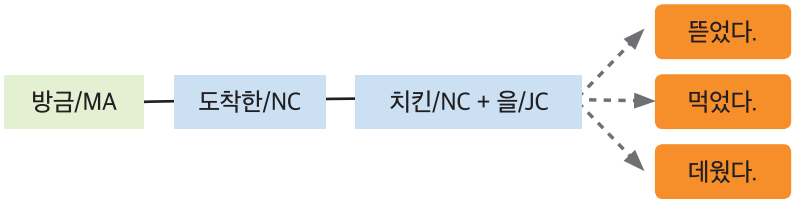


그림 3. 예문의 구조

본인의 경우 방금 도착한 치킨의 포장지를 뜯었다는 뉘앙스를 풍기는 ‘뜯었다.’라는 단어가 가장 높은 확률로 위치할 것 같다. 다음으로는 약간 치사하지만 그 짧은 사이에 치킨을 먹어버렸다는 뜻의 ‘먹었다.’가 나올 확률이 높을 것 같다. 마지막으로 ‘데웠다.’의 경우 식어 있는 치킨에 사용할 가능성이 높기에 막 도착한 치킨에 어울리지 않는 서술어라고 할 수 있겠다. 이렇게 필자의 의식의 흐름을 기술한 문장은 기존 경험에 기반한다고 할 수 있겠다. 즉, 치킨이 도착하면 먼저 먹지 않고 ‘포장을 뜯어 놓는다.’라는 경험이 가장 많기 때문이다. 만약 친구가 식탐이 강해서 배달원으로부터 치킨을 받자마자 한두 개씩 치킨 조각을 빼먹은 기억이 꽤 많다면, 서술어 중에서 ‘먹었다.’를 1순위로 꼽았을 것이다. 이렇게 사람의 경우는 경험에 기반하겠지만, 언어 모델의 경우는 학습하는 데이터에 기반한다. 이것이 바로 빈도 기반의 조건부 확률 모델링이라고 할 수 있다.

이런 확률 모델링은 다양한 서술어가 있지만, 그중에서 “방금 도착한 치킨을”이라는 단어의 조합이 등장했을 때 학습 데이터의 등장 빈도가 가장 높은 ‘먹었다.’를 사용하여 문장을 완성하게 된다. 하지만, 문장에 단어가 많은 경우는 서술어 하나를 위해서 문장의 처음부터 쭉 훑어야 하는 단점이 있다. 이는 엄청난 양의 연산과 예시가 있어야 하며 모든 경우의 수(문장)를 확보하기 어렵다. 이를 극복하기 위해서 몇 개의 단어[4]만 활용하여 다음에 어떤 단어가 와야 하는지 확률적으로 계산하는 방법이 사용되었다. 즉, 문장의 단어 n개를 뭉쳐서 다음 단어를 추천하는 N-Gram 알고리즘으로 몇 개를 뭉쳐서 보는가에 따라 1-gram, 2-gram으로 부르기도 하고 uni-gram, bi-gram으로 부르기도 한다.

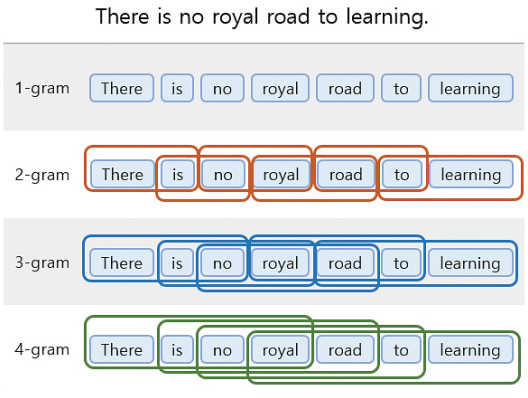


그림 4. N-Gram 예시

이 N-Gram 알고리즘은 기존에 앞의 모든 단어를 고려하는 것보다 상대적으로 간결하며 기존 모델 대비 상대적으로 유연하게 다음 단어를 예측할 수 있다는 장점이 있다. 단, 한국어의 경우 “나는 식사를 식당에서 합니다.”, “나는 식당에서 식사를 합니다.”처럼 어순을 지키지 않아도 의미가 충분히 전달되기 때문에 어순이 중요한 언어 대비 모델의 성능이 떨어질 수 있다.

신경망 기반 모델을 소개하기 전에 단어 간 유사도를 정보[5] 기반으로 하여 비슷한 단어를 예측하는 Word2vec(워드투

4) 정확하게는 단어가 아닌 말뭉치 또는 코퍼스(corpus)이나 독자의 이해를 위해 단어라고 하였다. 보다 깊게 공부하려면 해당 내용을 찾아보면 된다.
5) 희소 표현(Sparse Representation)의 반대인 밀집 표현(Dense Representation)으로 표현된 각 단어의 벡터(예시, [0.3, -2.3, 1.1, … , 4.5])이며 보다 자세한 내용은 워드 임베딩(word embedding) 관련 문서 참조.

벡터)을 소개하고자 한다. 앞에서 소개한 방법은 다음 단어를 예측하는 반면 이 Word2vec은 문장에 등장하는 앞, 뒤 단어를 참고하여(유사도 계산) 중심 단어를 예측한다는 점이 다르다. 이런 Word2vec의 특징을 활용하여 만들어진 사이트가 하나 있다. word2vec.kr/search에 접속하면 다음의 캡처 화면과 같이 재미있는 결과를 볼 수 있다.

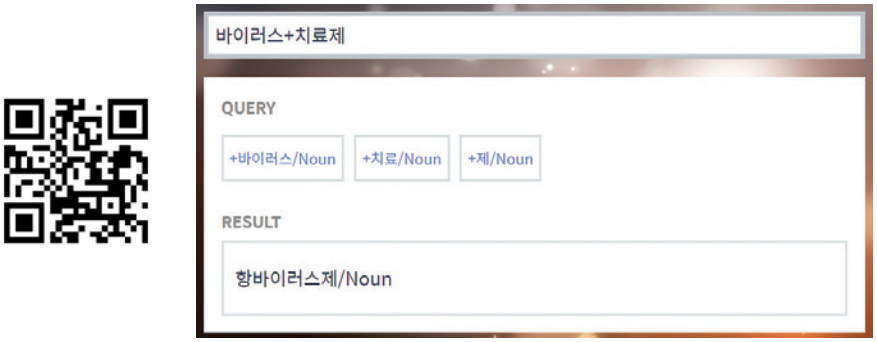


그림 5. Word2Vec 예시

어떤 단어가 밀접한지, 그 관계가 나름 계산이 되어있어 앞의 예시처럼 '바이러스'에 '치료제'의 속성을 더 하면 '항바이러스제'라는 단어가 도출된다. 또는 '방송 + 기술'이라고 입력할 경우 '프로그램'이라는 결과가 도출된다. 물론 이 결과는 모든 경우에 대해 만족스러운 결과를 내주는 것은 아니지만 '한국-서울+파리' 같이 한국에 서울이라는 수도의 속성을 빼고, 프랑스의 수도인 '파리'를 더해주면 '프랑스'라는 결과가 나온다. 이렇게 꼭 신경망 인공지능 모델이 아니더라도 재미있는 작품이 나올 수 있으니 독자분들은 꼭 한 번씩 시도해 보길 권장한다.

신경망 기반 모델

자연어 처리가 단순 빈도와 확률 기반의 처리를 벗어나 다층(심층) 신경망인 딥러닝을 적용하기 시작하면서 그 성능이 획기적으로 향상되었다. 특히 번역 분야에서는 Google과 Naver가 비슷한 시기에 자사의 번역기를 업데이트했는데 딥러닝 적용 전과 후를 비교해보도록 하자.

그림 6. 신경망 모델이 적용되기 전의 Google 번역기

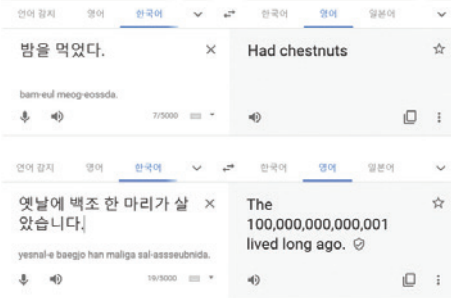


그림 7. 신경망 모델이 적용된 후의 Google 번역기

아직 백조는 어쩔 수 없지만[6] '밤을 먹었다.'는 그나마 괜찮은 편이다. 이 예시만 보면 신경망 모델이라고 해도 그렇게 대단하지 않다고 생각할 수 있지만, 기계 번역[7]에서 엄청난 결과물을 내어놓는다. 바로 zero-shot 번역인데 딥러닝 학습용 데이터가 적은 상태에서 학습을 하는 것을 few-shot learning, 단 하나의 데이터로 학습하는 것을 one-shot learning이라고 한다. 그런데 zero-shot learning. 즉, 학습 데이터가 전혀 없는 상황에서 번역이라니 도대체 어떤 것일까?

6) '옛날에 백조 두 마리가 살았습니다.'의 경우 올바르게 번역이 되는 것을 확인할 수 있다.
7) 사람이 아닌 컴퓨터로 자동 번역하는 것.

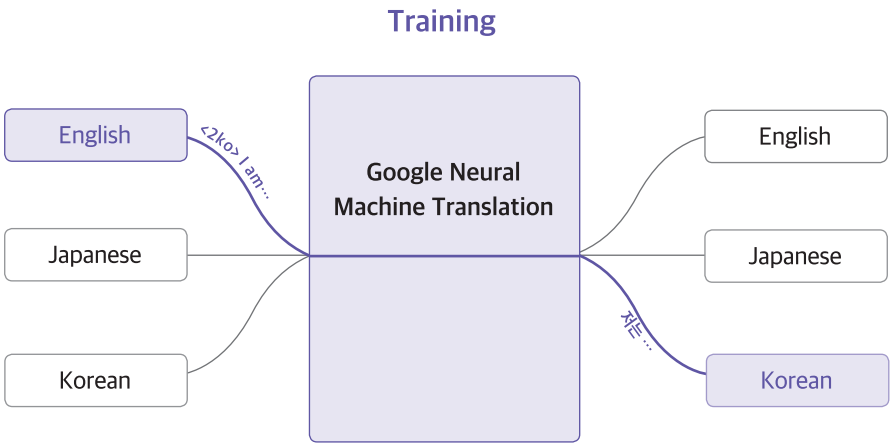


그림 8. Google의 자연어 번역 시스템(GMNT) 예시

기존에는 한국어-일본어 쌍을 번역하기 위해서는 각 언어의 데이터가 많이 필요하였다. 하지만 구글이 2016년 즈음 내놓은 새로운 인공지능 번역은 영어-한국어, 영어-일본어 번역 자료가 있을 경우 해당 자료를 분석하여 한국어-일본어 번역 데이터 없이 학습(zero-shot learning)하고 그 결과물을 내놓는다. 독자 중 이와 관련하여 보다 자세한 내용을 알고자 한다면, Google의 기술 블로그[8]를 참조하면 되겠다.
번역 이외에도 인공지능 스피커에 사용되는 기술도 전부 딥러닝 기반의 모델이 적용되었다고 볼 수 있는데, 이 부분은 굉장히 고난이도의 최신 기술이 요구되기 때문에 관련 기술을 나열하는 것만으로도 꽤 많은 지면을 할애하게 된다. 왜냐하면 앞에 나열한 모든 자연어 처리 기술을 기본으로 깔고 음성을 인식하고 처리하는 신호처리 기술까지 더해져야 음성인식을 어느 정도 할 수 있다고 볼 수 있기 때문이다.

여기까지 규모 있고 난이도가 높은 인공지능 프로젝트를 소개해서 조금 현실성이 떨어질 수 있겠다. 그렇다면 관련하여 조금 가벼운 주제에 대해 알아보는 것은 어떨까? 기존 통계적 모델의 단점을 어느 정도 극복하고 신경망 기반의 모델인 RNN(Recurrent Neural Network)을 활용한 영화 악평 생성기 제작 프로젝트를 한 번 살펴보는 것을 추천한다. 지금은 모 게임회사 인공지능 부서 연구원으로 재직 중인 송치성 씨의 발표 슬라이드 링크(bit.ly/bad_re)를 끝으로 이번 글을 마치고자 한다. 해당 슬라이드가 있는 웹 페이지에는 해당 프로젝트의 실습 코드까지 제공하고 있으니 관심 있는 독자는 꼭 한 번 도전해보기 바란다.

그럼 다음 글에서는 이미지 처리를 위해서 어떤 인공지능 기술이 사용되고 있는지 알아보도록 하겠다.

8) bit.ly/GMNT_blog