한국어 사전학습 모델을 활용한 자연어 처리 모델 자동 산출 시스템 설계

장지형[O,1], 최호윤[1], 이건우[2], 최명석[2], 홍참길[1]

한동대학교 전산전자공학부[1], 한국과학기술정보연구원 기계학습데이터연구단[2]
{jihyoung, 21600744}@handong.ac.kr, {gwlee, mschoi}@kisti.re.kr, charmgil@handong.edu

An Automated Production System Design for Natural Language Processing Models Using Korean Pre-trained Model

Jihyoung Jang[O,1], Hoyoon Choi[1], Gun-woo Lee[2], Myung-seok Choi[2], Charmgil Hong[1]
School of Computer Science and Electrical Engineering, Handong Global University[1]
Department of Machine Learning Data Research, Korea Institute of Science and Technology Information[2]

요 약

효과적인 자연어 처리를 위해 제안된 Transformer 구조의 등장 이후, 이를 활용한 대규모 언어 모델이자 사전학습 모델인 BERT, GPT, OPT 등이 공개되었고, 이들을 한국어에 보다 특화한 KoBERT, KoGPT 등의 사전학습 모델이 공개되었다. 자연어 처리 모델의 확보를 위한 학습 자원이 늘어나고 있지만, 사전학습 모델을 각종 응용작업에 적용하기 위해서는 데이터 준비, 코드 작성, 파인 튜닝 및 저장과 같은 복잡한 절차를 수행해야 하며, 이는 다수의 응용 사용자에게 여전히 도전적인 과정으로, 올바른 결과를 도출하는 것은 쉽지 않다. 이러한 어려움을 완화시키고, 다양한 기계 학습 모델을 사용자 데이터에 보다 쉽게 적용할 수 있도록 AutoML으로 통칭되는 자동 하이퍼파라미터 탐색, 모델 구조 탐색 등의 기법이 고안되고 있다. 본 연구에서는 한국어 사전학습 모델과 한국어 텍스트 데이터를 사용한 자연어 처리 모델 산출 과정을 정형화 및 절차화하여, 궁극적으로 목표로 하는 예측 모델을 자동으로 산출하는 시스템의 설계를 소개한다.

주제어: 한국어 사전학습 모델, 자연어 처리 도구, 모델 자동 산출

1. 서론

최근 신경망에 의한 자연어 처리(natural language processing) 연구는 Transformer[1]의 제안을 기점으로 해당 구조를 활용하여 대규모 언어 모델(language model)을 구성하고 이를 다양한 자연어 처리 문제에 활용하는 방향으로 발전해 오고 있다. 이러한 대규모 언어 모델에는 대표적으로 Google의 BERT(Bidirectional Encoder Representations from Transformers)[2], OpenAI의 GPT(Generative Pre-trained Transformer)[3] 그리고 Meta의 OPT(Open Pre-trained Transformer Language Models)[4] 등이 있으며, 이들은 각 개발사의 자원을 활용하여 대량의 언어 데이터로부터 미리 학습된 사전학습(pre-training) 모델의 형태로 GitHub이나 HuggingFace, 혹은 유사한 서비스 플랫폼을 통해 배포되고 있다. 다수의 주요 사전학습 모델들은 기본적으로 다국어 데이터를 사용해 학습되지만, 해당 학습 데이터를 이루는 한국어 데이터는 상대적으로 소수이기 때문에, 이들로부터 높은 한국어 처리 성능을 기대하기에는 한계가 존재한다. 때문에 국내 연구 그룹에서는 BERT[2], GPT[3] 등과 동일한 구조의 모델에 대규모 한국어 말뭉치를 학습시켜, KoBERT[5], KoGPT[6] 등과 같은 한국어 특화 사전학습 모델을 공개하고 있다.

배포된 사전학습 모델을 각종 응용작업에 적용하기 위해서는 데이터 준비, 코드 작성, 모델 파인 튜닝(fine-tuning) 및 저장과 같은 통상적인 과정을 수행해야 한다. 각 절차에 대한 구체적인 방법과 세부 과정은 논문, 온라인 게시물 등 공개되어 있는 자료를 참고할 수 있는데, 관련하여 실무 경험이 부족한 다수의 도메인 전문가 혹은 응용 사용자가 이를 활용하는 것은 쉽지 않다. 이는 참고하는 자료에 오류가 존재할 가능성이 여전히 존재하고, 사용자의 데이터에서도 공개된 성능을 달성한다는 것은 보장할 수 없기 때문이기도 하며, 절차를 수행하는 중에 실수를 할 수 있기 때문이다. 나아가 이러한 문제는 비단 대규모 사전학습 모델을 적용하는 중에만 생기는 문제는 아니며, 전통적인 기계 학습 모델이나 소규모의 신경망 모델을 활용하는 중에도 발생할 수 있다.

이러한 이유로 응용 현장에서는 AutoML (Automated Machine Learning)[7] 기법을 도입하여, 다양한 기계 학습 모델을 목적하는 문제에 보다 쉽게 적용할 수 있도록 시도하고 있다. 여기서 AutoML은 사용자가 제공하는 데이터에 대해 적용할 수 있는 기계 학습 모델을 탐색하고, 필요한 파라미터를 결정하는 과정을 정형화하여, 궁극적으로 의사결정 모델의 산출의 모든 과정을 자동화할 수 있도록 하는 접근이다. 즉, 사용자는 AutoML 기법을 통해 목적하는 데이터에 적합한 모델을 선택하고, 각 모델에 대해 최적의 성능을 달성하기 위해 여러 파라미터(parameters)를 조정하는 과정을 자동으로 진행할 수 있다. 하지만 AutoML의 루틴화된 절차가 진행되는 동안 사

용자는 학습

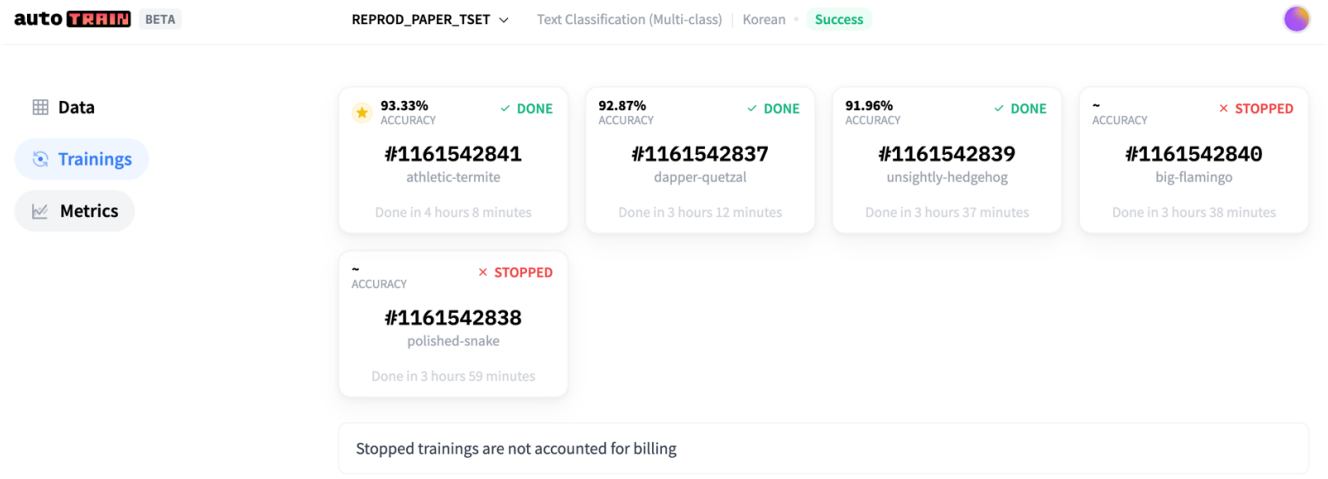

그림 1. AutoTrain 학습 결과 – AutoTrain에서 캡처한 화면

중간 과정을 구체적으로 파악하기 어려울 수 있고, 지원하는 기계 학습 문제의 종류(분류, 군집화, 번역, 생성 등)도 아직 제한적이라는 한계점이 있어, 개별 응용 사용자가 손쉽게 AutoML 기법을 취입하기에도 아직은 어려움이 따른다.

본 연구팀은 사용자가 제공하는 한국어 데이터로부터 자연어 처리 모델을 자동으로 산출하는 시스템의 설계 및 구현을 수행하고 있다. 본 논문은 해당 개발 과정에서 도출된 시스템 설계를 소개하고, 특히 사전학습 된 한국어 대규모 언어 모델을 활용하여 고도화된 예측 모델을 자동으로 산출할 수 있는 웹 기반 플랫폼의 구현 상세를 보고한다. 소개하는 시스템은 사용자 데이터와 사전학습 모델의 선택만을 입력으로 필요로 하며, 별도의 코드를 작성할 필요없이 자동으로 높은 성능의 모델을 산출할 수 있도록 설계되었다. 플랫폼에 탑재되는 사전학습 모델은 KoBERT[5], KoGPT[6], KorSciBERT[8], KorSciElectra[9] 등 한국어에 특화된 모델들로, 해당 모델을 활용한 자동 모델 산출 과정의 실시간 모니터링을 지원한다. 개별 학습 결과물에 대한 API(application programming interface)를 제공하여 사용자가 제3의 서비스에 학습된 모델을 호출하고 적용할 수 있도록 하여, 다양한 응용 현장에서 자동 산출된 모델을 활용할 수 있다. 또한, 학습에 사용된 코드와 라이브러리 정보도 함께 제공하여 사용자가 동일한 학습을 직접 재현해 볼 수 있도록 하였다.

2. 관련 사례

다음은 자동화된 한국어 자연어 처리 시스템의 설계를 소개하기 앞서 현재 공개적으로 이용가능한 AutoML 관련 자원들 중 대표적인 서비스 플랫폼 AutoTrain과 대중적으로 널리 사용되는 SW개발 라이브러리 AutoGluon을 살펴보고, 이들의 특징과 한계점을 논의한다.

2.1. AutoTrain

HuggingFace사에서 제공하는 AutoTrain[1]은 코드를 작성하지 않고도 사전학습 모델을 활용하여 자연어 처리 모델을 생성할 수 있는 AutoML 플랫폼이다. AutoTrain은 웹 서비스 형태로 제공되고 있으며, 텍스트 분류, 질의응답, 번역, 요약 등을 포함한 다양한 자연어 처리 문제를 수행할 수 있고, 한국어를 포함해 10여 개 이상의 다양한 언어로 표현된 데이터의 사용을 지원하고 있다. 사용자는 비용을 고려하여 모델 학습의 규모(몇 개의 독립적인 학습 스레드를 실행할지)를 결정하며, 다수의 유력한 사전학습 모델과 AutoML 기법을 동원하여 최종 모델을 산출한다. 사용자는 필요에 따라 HuggingFace Hub에 공개된 특정 모델을 지정하여 학습을 수행할 수도 있으며, 학습이 완료되었을 때엔 그림 1과 같이 학습 결과를 확인할 수 있다. 하지만 학습에 실패했다고 나타나는 모델에 대해서 그 이유를 제공하지는 않으며, 학습에 성공한 모델에 대해서도 학습과정을 관찰할 수 있는 기능을 제공되지 않는다. 또한, 학습에 영향을 미칠 수 있는 설정들을 사용자가 조정할 수 있도록 제공하고 있지 않아, 학습과정에 대한 보다 세밀한 제어는 수행이 어려울 수 있다.

2.2. AutoGluon

AutoGluon[10]은 텍스트 데이터뿐만 아니라 이미지, 시계열 데이터 등 대부분 유형의 데이터에 적용할 수 있는 범용 Python 라이브러리이다. 사용자는 해결하고

[1] https://huggingface.co/autotrain

자 하는 데이터 문제에 대해 AutoGluon을 활용하여 전통적인 기계 학습 모델과 딥러닝 모델을 포함한 다양한 의사결정 모델들을 산출할 수 있다. 모델 학습이 진행되는 동안에는 학습 상태에 대한 모니터링을 지원하며, 학습이 완료된 뒤에는 결과에 대한 시각화와 모델별 성능을 간단히 요약하는 편의를 제공한다. AutoGluon은 사용자로 하여금 코딩을 요구하는 SW개발을 위한 라이브러리로, 그림 2와 같은 코드 작성 및 실행을 통해 AutoML을 실현하지만, 사용자가 해당 라이브러리의 도움없이 모델 학습의 전 과정을 준비하는 것보다 훨씬 수월한 개발 경험을 제공하며, 코드 수정을 통해 정밀한 학습과정의 제어도 가능하다. 하지만 여전히 Python을 이용한 개발 지식이 부족한 경우는 사용이 매우 어려울 수 있으며, 구동을 위한 사용자의 연산환경이 필요하기 때문에, 사용성면에서 한계가 있다.

```
from autogluon.text import TextPredictor

df = pd.read_csv('DATA_PATH')
target = df[['TARGET']]
train_data, test_data = train_test_split(df, test_size=0.2,
        random_state=777, shuffle=True, stratify=target)

subsample_size = 1000
train_data = train_data.sample(n=subsample_size, random_state=0)
print(train_data.head(10))

predictor = TextPredictor(label='TARGET', eval_metric='acc', path='./ag_sst')
predictor.fit(train_data, time_limit=60)

test_score = predictor.evaluate(test_data)
print(test_score)
```

그림 2. AutoGluon 예시 코드

3. 시스템 설계

소개하는 시스템은 AutoML 기법을 채용하여 사용자가 데이터를 시스템에 업로드하고 원하는 옵션을 선택하는 것 만으로, 한국어 사전학습 모델에 기반한 고성능의 자연어 처리 모델을 자동으로 산출하는 것을 목표로 한다. 시스템은 웹 플랫폼의 형태로 구성하여, 사용자가 모델 학습과 산출을 위한 별도의 고성능 컴퓨팅 환경을 갖추지 않더라도 사용할 수 있도록 하고자 한다.

3.1. 설계 환경

시스템의 개발 및 운영을 위한 자원은 2기의 GPU를 갖춘 Ubuntu 20.04 기반의 리눅스 환경으로 Python과 Node.js를 주로 사용하여 구현이 이루어졌다. 기계 학습관련 업무의 수행을 위해 Tensorflow 및 Plotly가 사용되며, 사용자가 접근하는 온라인 플랫폼의 웹 프론트엔드(front-end) 구성을 위해 React가 사용된다. 웹서비스 이면에 구동되는 학습 프로세스의 관리를 위해 PM2가 사용되며, API 제공을 위해 Fastify가 함께 적용된다.

3.2. 프로젝트 생성

사용자는 하나의 프로젝트를 통해 하나의 데이터 문제를 지정하고, 이로부터 하나의 모델을 산출할 수 있다. 구체적인 프로젝트 생성 과정은 다음과 같다. 그림 3에서와 같이 사용자는 기반으로 사용하기 원하는 한국어 사전학습 모델과 다루려는 문제의 종류를 선택하면서 새로운 프로젝트를 생성한다. 현재 시스템은 SKT-AI가 제공하는 KoBERT[5], KoGPT[6]와 한국과학기술정보연구원(KISTI)에서 공개한 KorSciBERT[8], KorSciElectra[9]를 탑재하고 있으며, 텍스트 분류 문제에 대한 의사결정 모델 산출을 자동으로 수행할 수 있다.

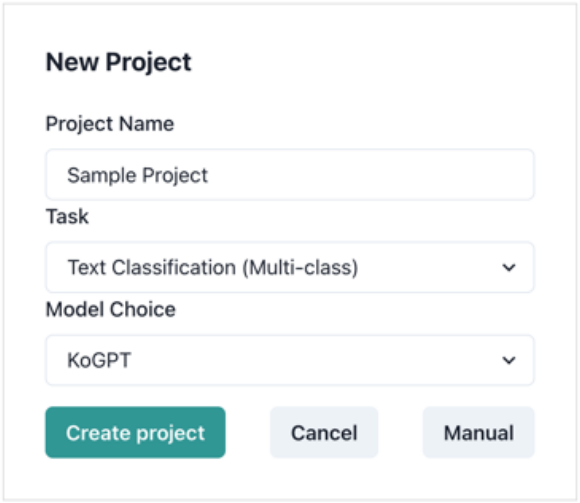

그림 3. 모델 및 문제 선택

선택 후 다음 화면에서 사용자는 사용하고자 하는 데이터를 업로드 한다. 업로드가 완료되면, 시스템은 그림 4와 같이 데이터의 행과 열 개수, 그리고 소수의 데이터 샘플을 화면에 보여주며 확인 과정을 거친다. 사용자는 데이터가 올바르게 업로드 되었는지 검토한 뒤, 모델의 입력과 출력이 되는 열을 지정하여 모델의 학습 과정을 준비한다. 혹은 플랫폼에서 기본적으로 제공하는 샘플 데이터셋을 학습에 사용할 수도 있다.

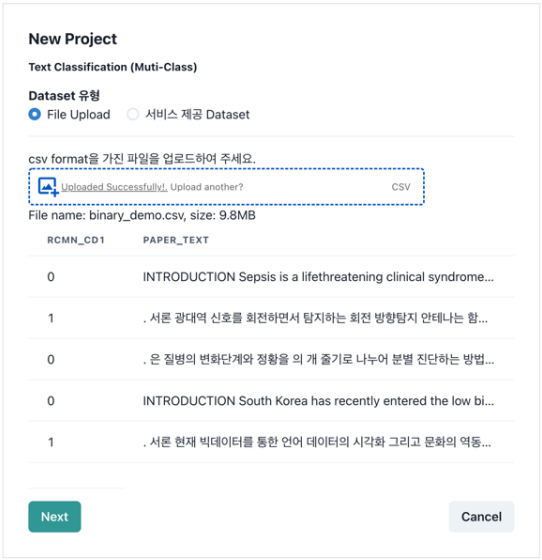

그림 4. 데이터 업로드

데이터가 준비된 후, 그림 5와 같이 모델 학습을 위한 하이퍼파라미터를 지정할 수 있는 화면을 제공한다. 사용자는 상기 소개된 것과 같이 최적의 성능의 모델을 자동으로 산출하는 옵션을 선택할 수도 있고, 보다 정밀하게 학습을 제어하고 싶은 경우에는 원하는 값을 수동으로 지정하여 모델 학습과정을 수행할 수도 있다. 자동으로 하이퍼파라미터를 찾는 방법을 선택한 경우, 베이지안 최적화(Bayesian optimization) 기법에 의해 사전에 정의된 범위 안에서 하이퍼파라미터 값을 수정해 나가면서 최적의 모델을 추적해간다. 이렇게 학습이 진행되기 위한 정보를 모두 입력하면, 시스템은 모델 산출을 위한 학습을 시작한다.

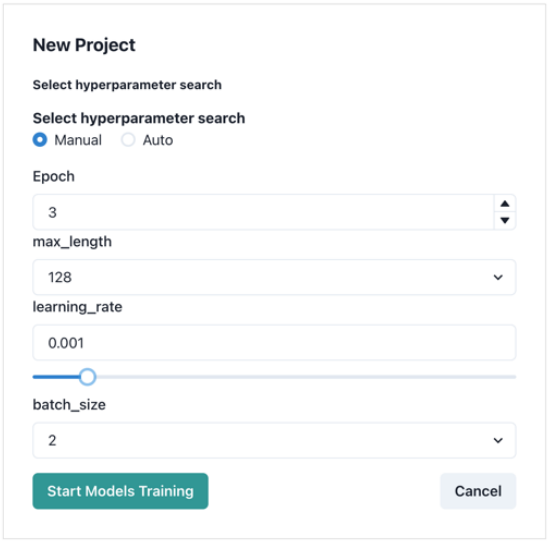

그림 5. 모델 하이퍼파라미터 입력

3.3. 모델 학습과정 수행 및 결과 확인

전술한 과정 이후, 본 시스템은 사용자 프로젝트를 통해 설정된 선택지들에 따라 모델 산출을 위한 학습(사전학습 모델에 기반한 추가 학습)이 진행된다. 학습을 위한 실제 코드는 템플릿 형태로 준비되어 사용자에게 제공되며, 실행 시 사전학습 모델과 데이터의 위치가 구체적으로 지정된다. 완성된 코드는 모델 산출 이후 사용자에게 제공되며, 섹션 3.3에서 이를 소개한다.

사용자는 그림 6과 같은 대시보드를 통해 학습 진행과정을 실시간으로 모니터할 수 있다. 즉, 제공되는 대시보드를 통해 정확도(accuracy), 손실(loss)와 같은 정보를 학습 곡선(learning curve)의 형태로 제공하여 학습이 적절하게 이루어지고 있는지를 실시간으로 추적할 수 있으며, 혼동 행렬(confusion matrix)과 ROC(receiver operating characteristic) curve, PR(precision-recall) curve도 함께 제공하여, 다양한 성능 지표를 통해 모델이 달성하고 있는 바를 구체적으로 파악할 수 있도록 도움을 주고자 한다.

그림 6. 학습 결과 대시보드

이러한 평가 지표들은 학습 횟수마다(epoch) 갱신되며, 동시에 모델의 체크포인트(checkpoint)도 동시에 저장하여 사용자가 추후 원하는 학습 횟수를 쉽게 추적할 수 있도록 하였다. 학습 로그를 포함하여 이러한 결과물들은 사용자가 직접 다운로드 받을 수 있도록 그림 7과 같은 파일 탐색기 인터페이스로 정리하여 제공하고, 학습이 완료된 최종 모델 산출물도 함께 다운로드 받을 수 있다.

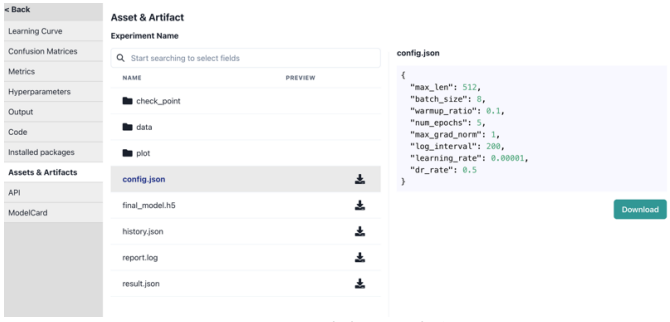

그림 7. 모델 산출물 다운로드

3.4. 학습 재현 및 배포

학습이 완료되면 사용자는 최종 산출된 예측 모델 뿐만 아니라, 학습 전 과정을 재현할 수 있는 코드와 재현 환경 구성을 위해 필요한 라이브러리 목록에 대한 액세스도 갖게 된다. 별도의 기계 학습 모델 학습환경을 갖추고 있는 사용자는 제공되는 라이브러리 목록을 활용하여 스스로 실행환경을 구축하고, 제공되는 코드를 실행하여 동일한 실험을 재수행 할 수 있다.

또한, 본 시스템은 산출된 자연어 처리 모델에 대하여 API를 제공하여 향후 제3의 프로그램이나 웹 서비스로부터 자유롭게 호출 및 활용할 수 있는 기능을 제공한다. 즉, 프로젝트마다 생성된 예측 모델에 대하여 REST(representational state transfer) API 형태의 서비스 엔드포인트(endpoint)를 필요에 따라 구성할 수 있으며, 이를 통해 사용자는 본 시스템의 웹 프론트엔드를 통하지 않고도, 생성된 모델을 호출하여 원하는 예측을 수행할 수 있다. 본 시스템은 대시보드를 통해 해당 API를 테스트해 볼 수 있는 샘플 인터페이스를 제

공하며, 그림 8은 이를 소개한다. 즉, 분류 모델의 경우에는 테스트할 내용을 입력하면, 학습된 모델을 기반으로 해당 내용이 어떠한 주제로 분류되는지를 제시한다.

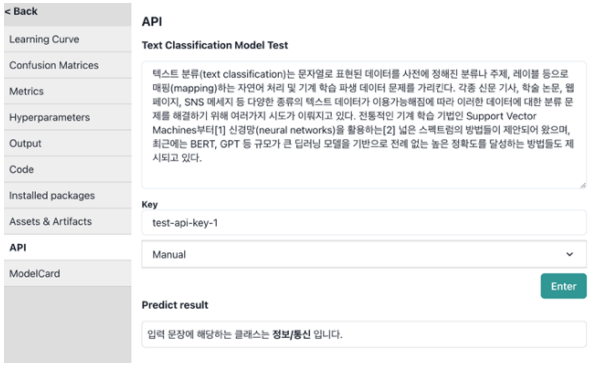

그림 8. 모델 테스트

3.5. 프로젝트 공유

본 시스템에서 생성된 모든 프로젝트는 모델 카드(model card)[11]라고 하는 메타 데이터(meta data)를 통해 관리된다. 모델 카드는 학습된 기계 학습 모델의 데이터 정보, 모델 정보, 평가 정보 등을 효과적으로 공유할 수 있도록 가이드하는 데이터 스키마(data schema)이다. 즉, 학습된 기계 학습 모델의 고유한 정보를 구체적으로 문서화할 수 있는 형식을 제공한다. 본 시스템은 모델 카드의 형식을 채용, 이를 보다 간소화된 형태[12]로 수정하여 프로젝트 관리에 사용한다. 그림 9는 이렇게 생성된 모델 카드의 예시 화면이다.

모델 카드는 시스템 내의 여러 사용자로부터 다수의 프로젝트가 생성되는 가운데, 이들 프로젝트에 대한 효과적인 검색을 지원하는 데에 활용된다. 향후 모델 카드에 기입된 내용을 바탕으로, 시스템 내에 동일한 내용의 프로젝트가 수행되었던 기록이 있다면, 기존 결과물을 공유 및 재사용할 수 있도록 시스템을 확장개발하는 데에도 활용하고자 한다.

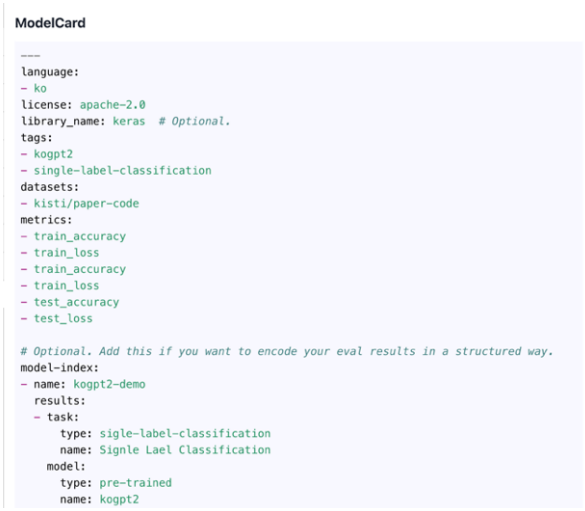

그림 9. 모델 카드

4. 결론

본 논문은 한국어 사전학습 모델을 활용하여 자연어 처리 모델을 자동 산출할 수 있는 AutoML 기반 기계 학습 시스템의 설계를 소개하고, 그 구현의 진행 현황을 보고했다. 사용자는 해결하려는 데이터 문제를 준비하고 간단한 과정을 거쳐 프로젝트를 생성하여, 고성능의 한국어 처리 모델을 코드 작성 없이 획득할 수 있다. 제안된 시스템은 관련 서비스인 AutoTrain과 달리 학습 과정을 상세하게 제어할 수 있는 기능을 제공하고, 학습 과정을 실시간으로 모니터링 및 보관할 수 있다는 장점이 있다. 또한 코드를 전혀 작성하지 않아도 자동으로 기본적으로 탑재된 한국어 사전학습 모델을 사용하여 고품질의 예측 모델을 산출할 수 있고, 온라인 서비스로 사용자가 별도의 연산환경을 준비하지 않아도 되며, 자동 생성되는 API를 통해 웹 프론트를 통하지 않고 생성된 모델을 사용할 수 있다는 점에서 높은 사용성을 제공한다고 할 수 있다. 나아가 내부적으로 프로젝트 기록을 생성, 보관하는 과정을 보다 정형화하기 위하여 모델 카드를 도입하였으며, 이는 향후 시스템 전체적인 효율성 재고를 위한 꼭지점으로도 활용해가고자 한다. 향후 더욱 다양한 한국어 사전학습 모델을 시스템에 탑재하고, 번역, 요약, 생성, 질의응답 등 텍스트 분류 외의 자연어 처리 문제도 지원하도록 시스템의 개발을 지속하여 서비스의 사용성 및 활용도를 증대시켜 가고자 한다.

감사의 글

본 연구는 2022년 한국과학기술정보연구원(KISTI)의 위탁연구 과제로 수행한 "기계학습 모델 개발, 공유 및 코드 품질 계측 방법론 연구"의 일부분임.

이 논문은 과학기술정보통신부의 소프트웨어중심대학 지원사업 (2017-0-00130)의 지원을 받아 수행하였음.

참고문헌

[1] A. Vaswani, et al., “Attention Is All You Need,” Advances in neural information processing systems, 30, 2017.

[2] J. Devlin, et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv preprint arXiv:1810.04805, 2018.

[3] T.B. Brown, et al., “Language Models are Few-Shot Learners,” Advances in neural information processing systems, 33, 2020.

[4] S. Z hang, et al., “OPT: Open Pre-trained Transformer Language Models,” arXiv preprint arXiv:2205.01068, 2022.

[5] SKTBrain, “KoBERT,” URL: https://github.com/SKTBrain/KoBERT

[6] SKT-AI, “KoGPT2,” URL: https://github.com/SKT-AI/KoGPT2

[7] X. He, et al., “AutoML: A Survey of the State-of-the-Art,” Knowledge-Based Systems, 212, 106622, 2021.
[8] 한국과학기술정보연구원, “과학기술분야 BERT 사전학습 언어모델 (KorSciBERT),” URL: https://doi.org/10.23057/46
[9] 한국과학기술정보연구원, “한국어 과학기술분야 ELECTRA 사전학습 모델 (KorSci-Electra),” URL: https://doi.org/10.23057/51
[10] N. Erickson, et al., “AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data,” arXiv preprint arXiv:2003.06505, 2020.
[11] M. Mitchell, et al., “Model Cards for Model Reporting,” In Proceedings of the conference on fairness, accountability, and transparency, pp. 220-229, 2019.
[12] HuggingFace, “Model Cards,” URL: https://huggingface.co/docs/hub/models-cards