[NER]

- Successfully incorporated ‘invoking SageMaker endpoint’ process -> model

(obtained results and installed necessary modules)
>> will I be using a container? How will I incorporate this modules in Lambda/ entry_point?
>> 밑에

- Need to verify the NER/ prediction output I’m getting is the correct one

>> what should I be getting as an output for
1등잡티세럼 40ml+마스크 미니어쳐 증정 OR ["1등잡티세럼 40ml+마스크 미니어쳐 증정", "노란로션 (로션+/젤)”]?

>> get input (parse arguments)

- ERROR: tried multi-product input for testing -> tensor error

>> tried feeding in multi-product list to endpoint -> tensor error (should be 19 in length, not 13)
>> 이메일 답장:
첫번째는, 제가 금일 model3and4_test.ipynb 가장 아래쪽에 추가한 셀처럼,
inference를 여러번 진행하는 것입니다.
데이터의 양이 많지 않은 경우에는 해당 방식으로 데이터를 처리하는 것이 간단할 듯 합니다.
- 장점 : 가변적인 문장 길이 가능, 구현 간단함
- 단점 : 타 방안에 비해 속도가 느릴 것으로 예상됨

두번째는, 문장 길이를 사전에 정의하여
모든 문장의 길이를 같게 만드는 것입니다.
(모든 문장의 뒷부분에, 길이가 모자란 만큼 dummy 값을 padding하는 방식입니다.) >> 0
- 장점 : 속도 개선 가능성 있음
- 단점 : 고정된 길이 이상의 문장 입력 불가, 모델 성능 소폭 하락 가능

[Lambda vs. entry_point]

- Why Lambda?

A. Current inference code needs input for final decoded result
B. main -> lambda_handler 후 업로드, 끝
>> external modules could be deployed to Lambda as a .zip file containing Lambda function code and dependencies-what about Python modules?

>> model/ pre and post processing (includes retrieving model predictions/ serving framework
>> modularizes components by their functions
>> Lambda used for ‘parsing’ purposes only
>> the serving side: do we necessarily need Flask? (Just use API Gateway resources)

- Why entry_point? (Inference pipeline-script in **containers**)

A. It combines a model and a preprocessor to a final 'model', pipeline model to be exact.
(A preprocessor includes pre/post processing scripts)
*our entry_point can't just be a script since we are required to use python modules

[AWS]
- I/O handling architecture

>> deploy to Lambda/ test
- Tags
- Container type
- Batch transform >> expense

- ✅ Tomorrow 10AM: AWS
- ✅ Wednesday 4PM: mid-review (architecture -> io handling)