**인공지능**

**6조**

**Seq2Seq 모델을 이용한 뉴스 요약**

iot인공지능 융합전공

165987

구태희

**1. 팀원 구성 및 역할**

|  |  |  |  |
| --- | --- | --- | --- |
| **전공** | **학번** | **이름** | **역할** |
| IoT인공지능융합전공 | 165987 | 구태희 | 팀장 및 코드작성 |
| 조경학과 | 160076 | 최호준 | 데이터분석 및 전처리 |
| 전자컴퓨터공학부 | 154678 | 김선명 | 발표 준비 및 발표 |
| 지능형모빌리티융합전공 | 171234 | 김민영 | 자료조사 및 발표준비 |

**2. 주제선정**

매일 수많은 뉴스기사가 생성되지만, 모든 뉴스를 읽고 내용을 파악하기란 현실적으로 쉽지 않다. 뉴스기사에는 내용을 압축적으로 표현하는 헤드라인을 작성하지만 실제 헤드라인과 내용이 매번 일치하지는 않다. 만약 뉴스기사의 전문을 축약하여 핵심 내용을 알 수 있다면 시간을 절약하고 얻을 수 있는 정보 또한 증가할 것이다. 딥러닝을 활용하여 뉴스기사를 축약하고 전달하기 위한 방법을 만들기 위해 뉴스를 요약하는 모델이라는 주제를 선정하였다.

**3. 개발환경**

Notion을 활용하여 팀원들과 문서를 공유하고 진행상황을 기록한다. 또한 매주 목요일 오후 7시 A.I 융합대학 3층 라운지에서 정기적인 모임을 가져 진행 과정을 확인하고 프로젝트의 진행 방향을 논의한다. 개발환경은 Google의 Colaboratory으로 모델 설계 및 학습을 진행한다.

**4. 모델 선정**

뉴스기사를 요약하는 것은 우리가 일상 생활에서 사용하는 언어, 즉 자연어의 의미를 분석하고 컴퓨터를 통해 처리할 수 있도록 하는 것이다. 단순히 문장을 몇 가지 단어로 요약하는 것이 아닌 핵심 문맥을 반영하여 새로운 문장을 생성하여 원문을 요약하는 추상적 요약(abstractive summarization)을 토대로 모델을 선정한다. 이러한 방법의 대표적인 모델 인 시퀀스-투-시퀀스(Sequence-to-Sequence, 이하; Seq2Seq) 모델과 향상된 결과물의 산출을 위해 어텐션 메커니즘(Attention Mechanism)을 추가로 이번 프로젝트에 사용하기로 결정하였다.

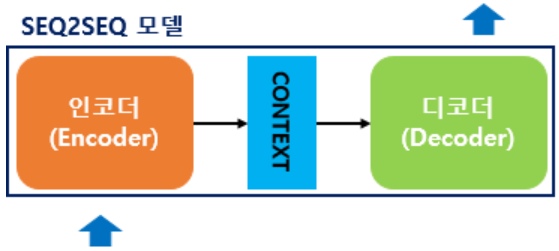
**5. Seq2Seq 모델**

**5-1. 워드 임베딩**

Seq2Seq모델에서 사용되는 모든 단어들은 임베딩을 통해 표현된 임베딩 벡터이다. 워드 임베딩(word embedding)은 단어를 밀집 벡터(dense vector)의 형태로 표현하는 방법이다. 밀집 표현(Dense Representaion)은 0과 1로만 표현되는 원-핫 벡터의 반대 표현으로 벡터의 차원을 단어 집합의 크기로 상정하지 않는다. 설정된 값으로 모든 단어의 벡터 표현의 차원을 맞추고 이 과정에서 실수값을 통해 단어를 표현한다.

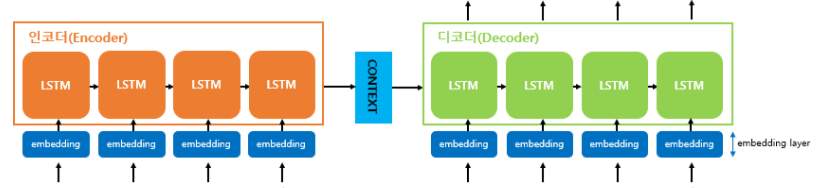
**5-2. Seq2Seq**

Seq2Seq모델은 2014년 12월에 발표된 모델로 인코더-디코더(encoder-decoder) 모델을 토대로 확장된 모델이다. Seq2Seq모델은 인코더와 디코더 두 개의 아키텍처로 구성된다. 인코드는 입력 문장의 모든 단어를 순차적으로 입력 받아 단어의 정보들을 압축하여 하나의 벡터, 컨텍스트 벡터(context vector)를 생성한다. 입력 문장의 정보를 하나의 컨텍스트 벡터로 압축하여 인코더는 디코더로 전송하고 디코더는 컨텍스트 벡트를 받아 단어를 한 개씩 순차적으로 출력한다.



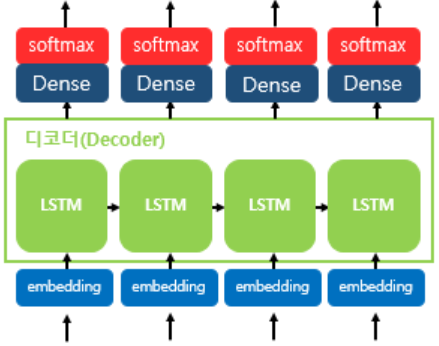
인코터 아키텍처와 디코더 아키텍처는 두 개의 RNN 아키텍처로 입력 문장을 받는 RNN 셀이 인코더, 출력 문장을 출력하는 RNN 셀을 디코더라 한다. 인코더에서 입력되는 문장은 단어 토큰화를 통해서 단어 단위로 쪼개지고 단어 토큰 각각은 RNN 셀의 각 시점의 입력이 된다. 인코더 RNN 셀은 모든 단어를 입력 받아 인코터 RNN 셀의 마지막 시점의 은닉 상태를 디코더 RNN 셀로 넘겨주고 이를 컨텍스트 벡터라 한다. 컨텍스트 벡터는 디코더 RNN 셀의 첫 번째 은닉 상태로 사용된다

Seq2Seq모델을 훈련 과정에서 인코더가 컨텍스트 벡터와 실제 정답 문장을 받았을 때 예측을 통해 정답이 나오도록 정답을 알려주며 훈련을 진행한다.



이러한 과정에서 사용되는 단어들은 워드 임베딩을 통해 표현된 임베딩 벡터이다. 임베딩 단계를 거치며 임베딩 층(embedding layer)이 쌓이게 된다.

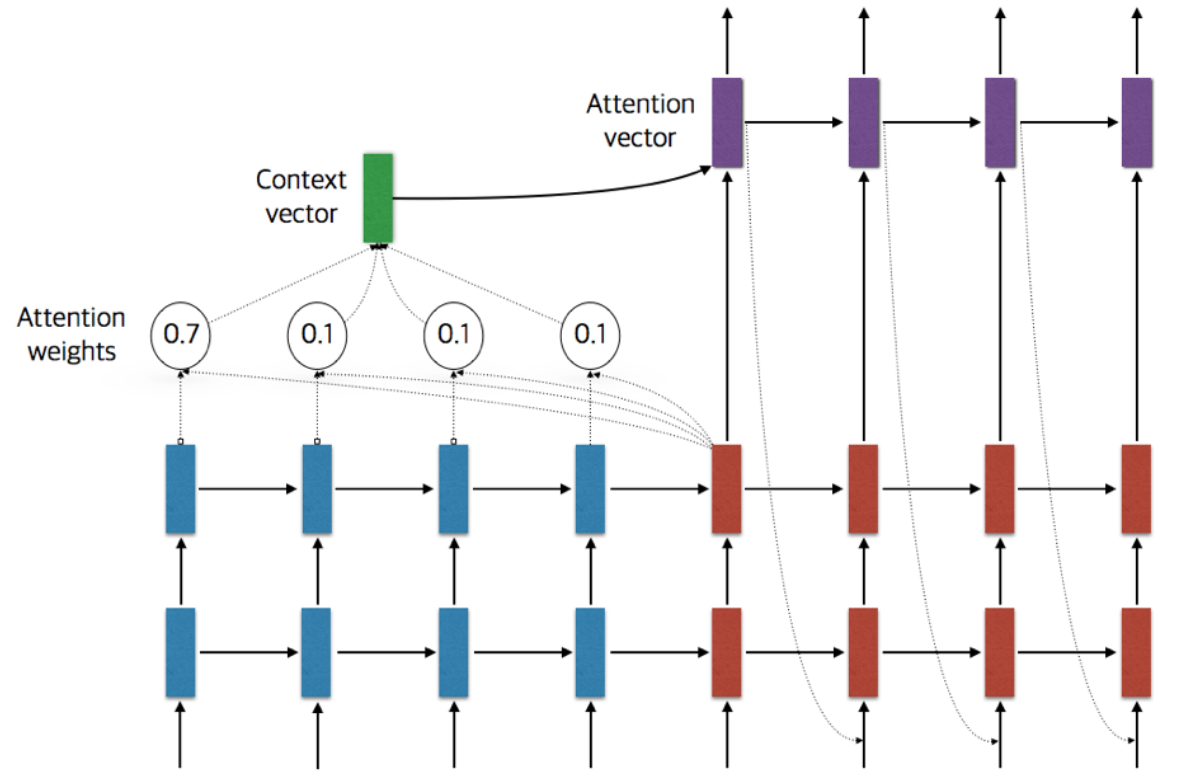
테스트 과정에서 디코더는 초기 입력으로 문장의 시작의 의미하는 <sostoken>이 들어간다. 이후 <sostoken> 입력 이후 다음에 등장할 확률이 가장 높은 단어를 time step마다 예측하고 예측된 단어를 다음 time step의 디코터 RNN 셀의 입력으로 사용한다. 이러한 과정을 문장의 끝을 의미하는 심볼 <eostoken>이 다음 단어로 예측될 때까지 반복한다. 이러한 예측에서 소프트맥스(softmax)함수를 이용하여 출력 시퀀스의 각 단어별 확률값을 반환하고, 디코더는 출력 단어를 결정한다.



**5-3. 어텐션 메커니즘(Attention Mechanism)**

Seq2Seq 모델의 경우 RNN 기반의 모델로 단점이 존재한다. 고정된 크기의 벡터에 모든 정보를 압축하여 정보 손실이 발행할 수 있다. 그리고 기울기 손실(Vanishing Gradient)문제가 존재한다. 이러한 단점은 입력 문장의 길이가 길어질수록 모델의 결과가 부정확해지는 결과가 나타난다.

이러한 단점을 해결하기 위해 어텐션 메커니즘을 활용하여 좀 더 복잡한 디코더를 사용한다. 어텐션 메커니즘의 디코더는 매 time step 별로 새로 생성되는 토큰을 결정하는 과정에서 원래의 문장에서 가장 가까운 관계를 가지고 있는 토큰을 결정하는 구조이다. 어텐션 메커니즘의 도입을 통해 인코더의 결과를 고정 길이 벡터에 담아야 하는 문제가 해소된다. 인커더의 time step 마다 생성되는 벡터가 어텐션에 사용되어 시퀀스의 길이에 비례하여 더 많은 정보가 활용된다.



**6. 데이터 선정**

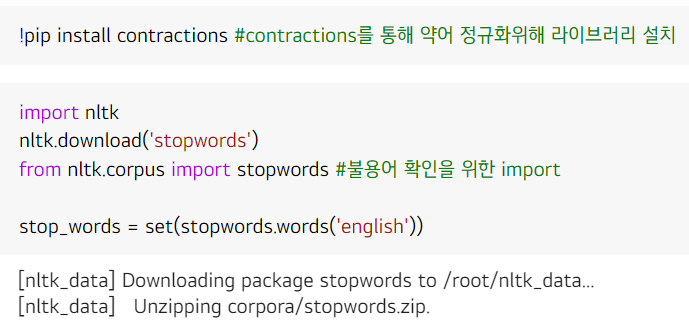
Seq2Seq 모델의 경우 지도 학습의 방법으로 훈련을 위해서는 뉴스 원문과 실제 요약문이 존재하는 데이터를 구해야 한다. 이러한 원문과 실제 요약문이 있는 데이터로 kaggle의 NEWS SUMMARY의 news\_summary\_more.csv를 사용한다. 98401개의 영어 뉴스(원문)와 헤드라인(요약문)를 담고 있다. 팀원 모두 자연어 처리를 해보는 것이 처음으로 상대적으로 한국어보다 난이도가 쉬운 영어기사를 선정하였다.



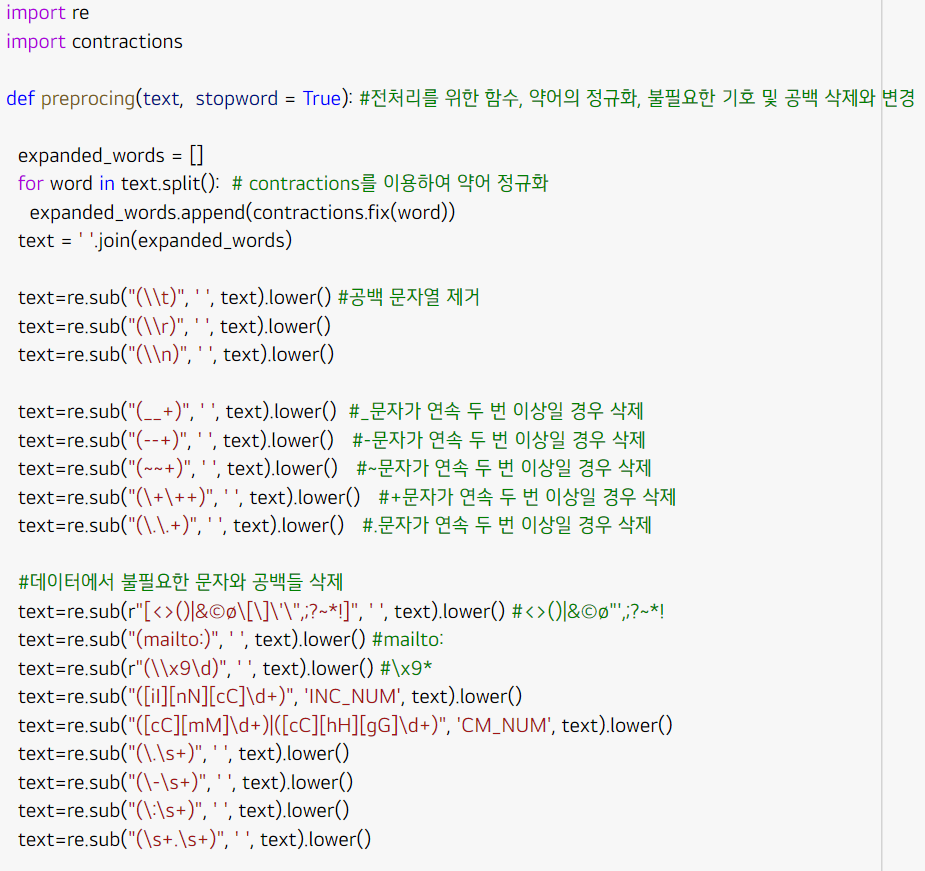
데이터의 headlines 열이 요약문을 담고 있어 직관적인 이해를 위해 열의 이름을 summary로 변경한다.

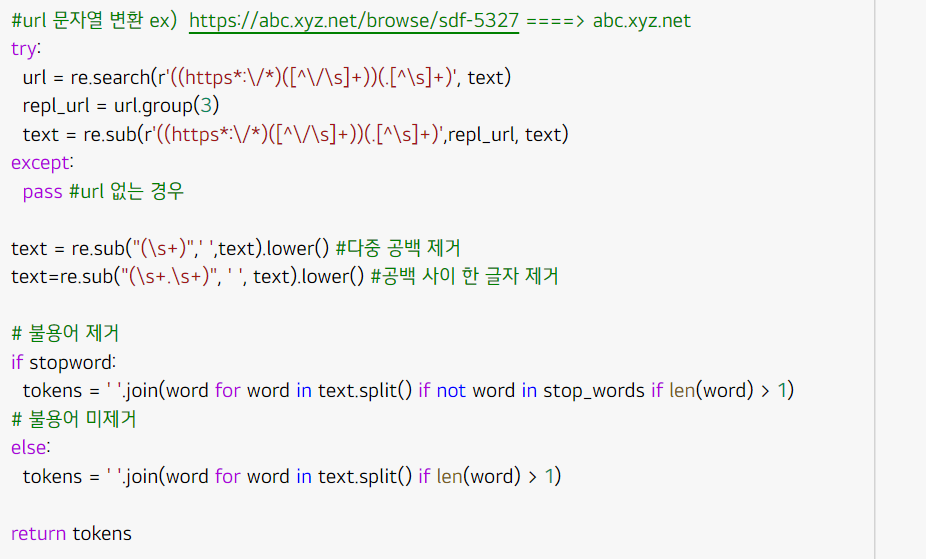
**7. 데이터 전처리**

영어의 경우 단어를 요약하여 동일한 의미를 가지고 있지만 스펠링이 다른 단어들이 존재한다. 이러한 단어를 분리하기 위해 약어들의 정규화가 필요하다. 또한 문장 내에서 문맥에 영향을 주지는 않지만 문맥적인 활용을 위해 존재하는 불용어들이 존재한다. 이러한 불용어를 제거하여 많은 의미를 담고 있는 단어를 위주로 학습을 진행할 필요가 있다.



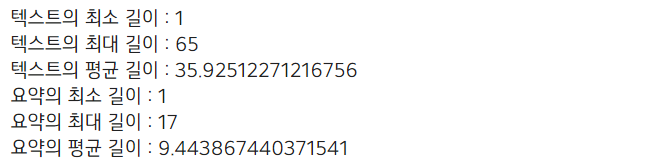
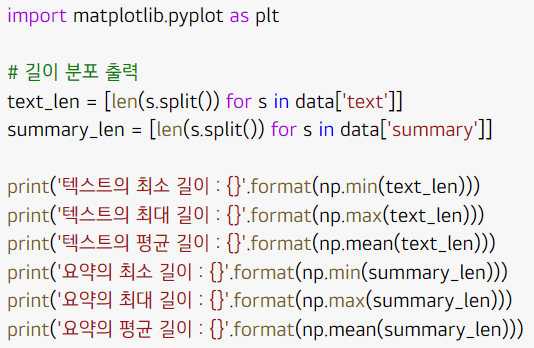
활용하는 데이터의 특성에 따른 전처리 역시 진행한다. 데이터의 불필요한 문자(특수문자, 기호…)나 공백이 연속으로 들어가는 경우 또는 기사에서 기자의 메일이 담겨있는 경우도 존재한다. 따라서 이러한 데이터를 정제하기 위한 전처리 함수를 정의하고 데이터 전처리를 진행한다.

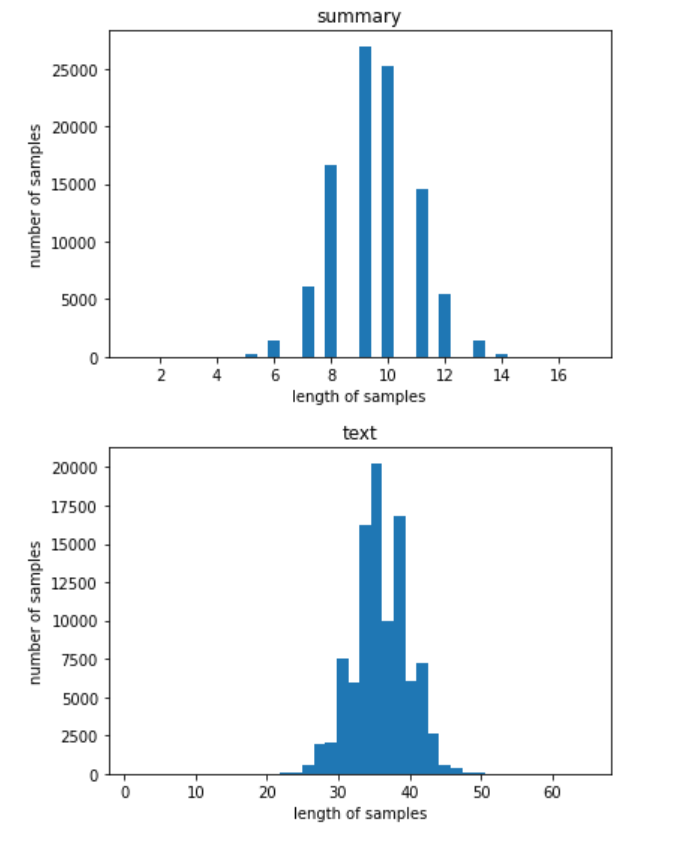
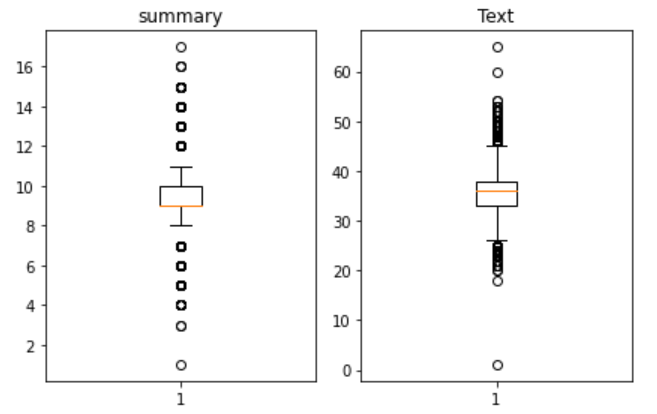




**8. 데이터 분포 확인**

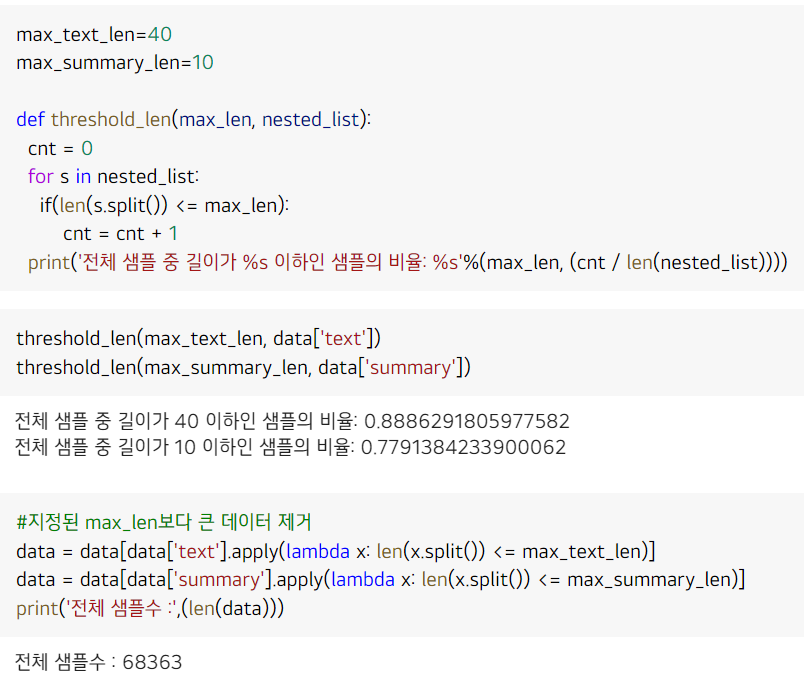
Seq2Seq 모델의 경우 문장의 길이가 길어지면 정확도가 떨어지는 단점이 존재한다. 따라서 데이터의 길이 분포를 확인하여 학습에 사용하기 앞서 적당한 길이의 데이터로 정제한다.





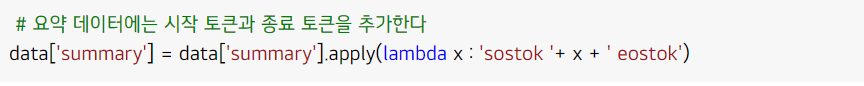
원문 텍스트의 경우 대체로 45이하의 길이를 가지며 평균 길이 35.9로 이루어져 있다. 요약문의 경우 대체적으로 12 이하의 길이를 가지며 평균 길이 9.4로 이루어져 있다.

패딩의 길이를 결정하기 위해 max\_len을 지정고 지정된 max\_len를 넘어가는 데이터를 제외한다. 원문의 max\_text\_len의 경우 40, 요약문의 max\_summary\_len의 경우 10으로 지정한다.



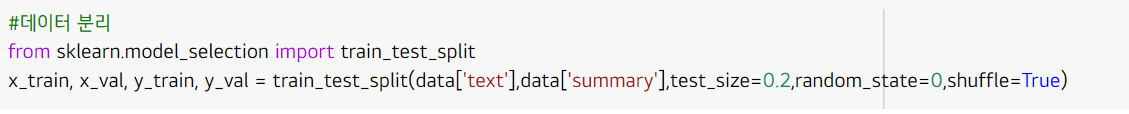
**9. 토큰화**

Seq2Seq 모델 학습을 위해서는 시작 토큰과 종료 토큰을 추가할 필요가 있다. 시작 토큰은 ‘sostok’, 종료 토큰은 ‘eostok’이라 이름하고 이를 레이블에 해당하는 요약문에 추가한다.



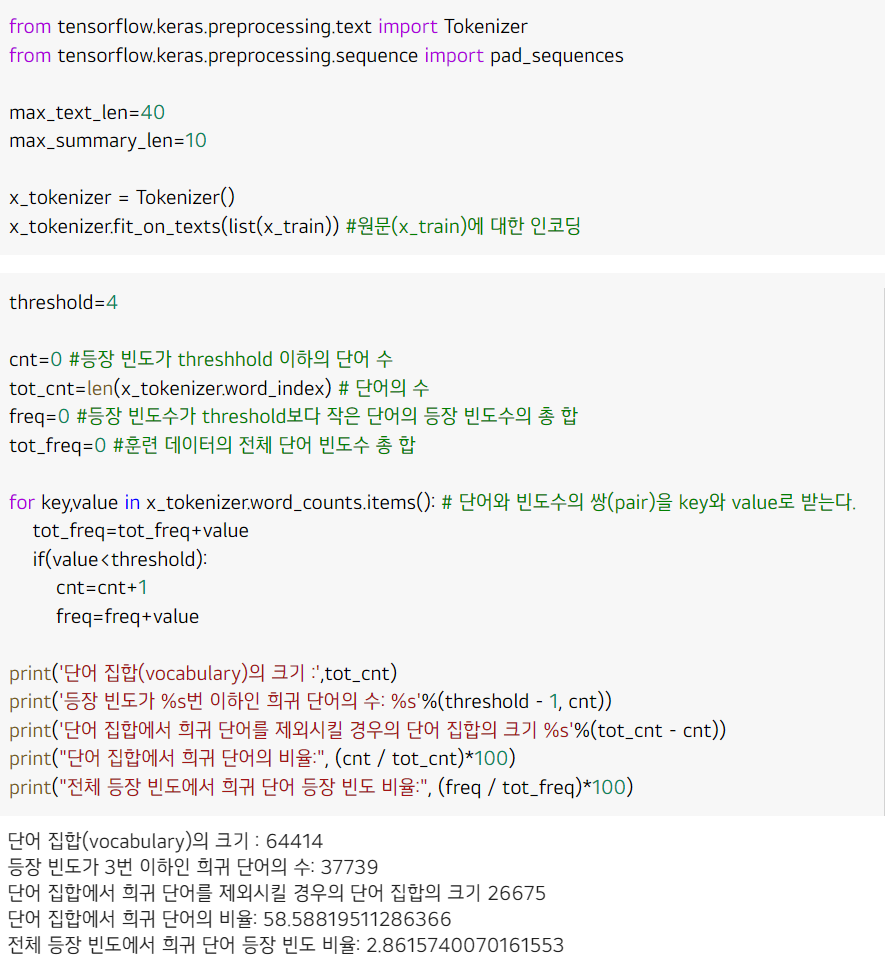
**10. 데이터 분리**

모델 학습을 위해서는 train 데이터와 test 데이터의 분리가 필요하다. train 데이터 80%, test 데이터 20%로 지정하여 전처리가 진행된 데이터를 분리한다.

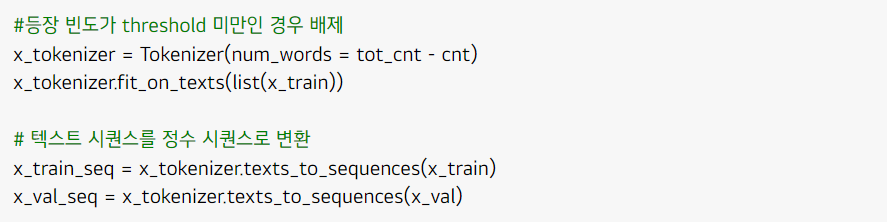


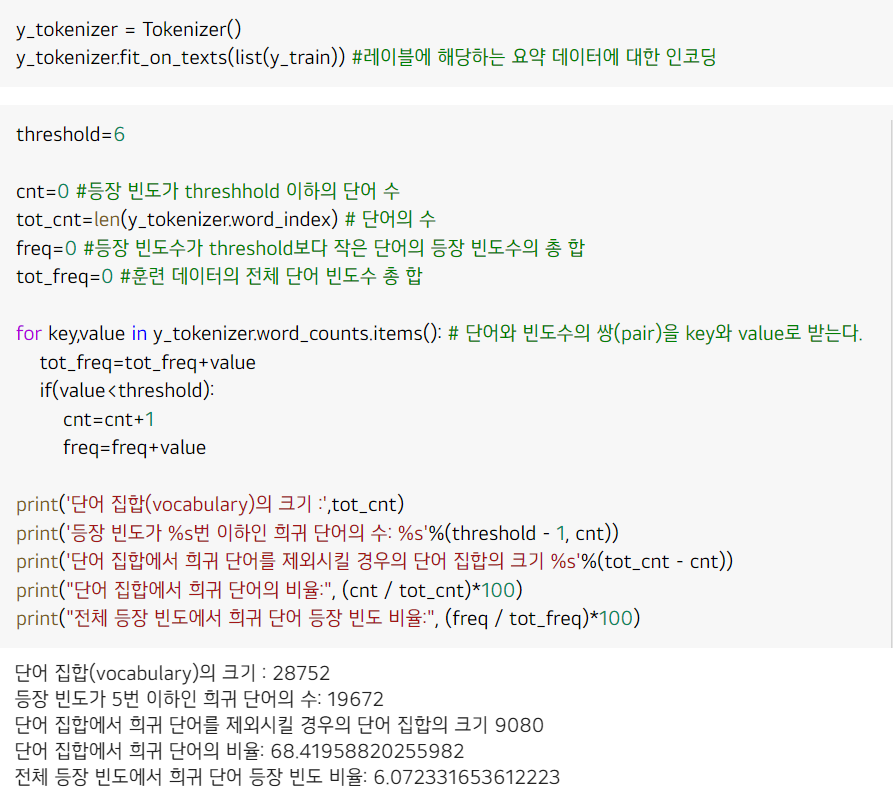
**11. 정수 인코딩**

데이터의 존재하는 자연어를 컴퓨터를 통해 처리할 수 있도록 텍스트를 숫자로 변환한다. 단어 집합의 생성과 동시에 각 단어에 고유한 정수를 부여하며 단어 집합(vocaburary)를 생성한다. 이때 단어 집합 내 등장 빈도를 확인하여 등장 빈도가 적은 단어들은 인코딩 과정에서 제외한다.

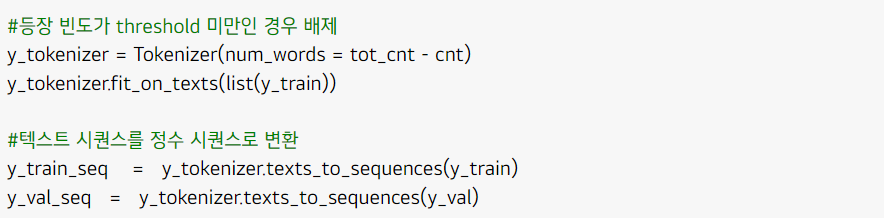


x\_train의 경우 threshold를 4로 지정한다. 등장 빈도가 3번 이하의 단어는 전체 등장 빈도에서 약 2.8% 만을 차지하고 있다. 이를 배제하고 정수 인코딩을 진행한다.



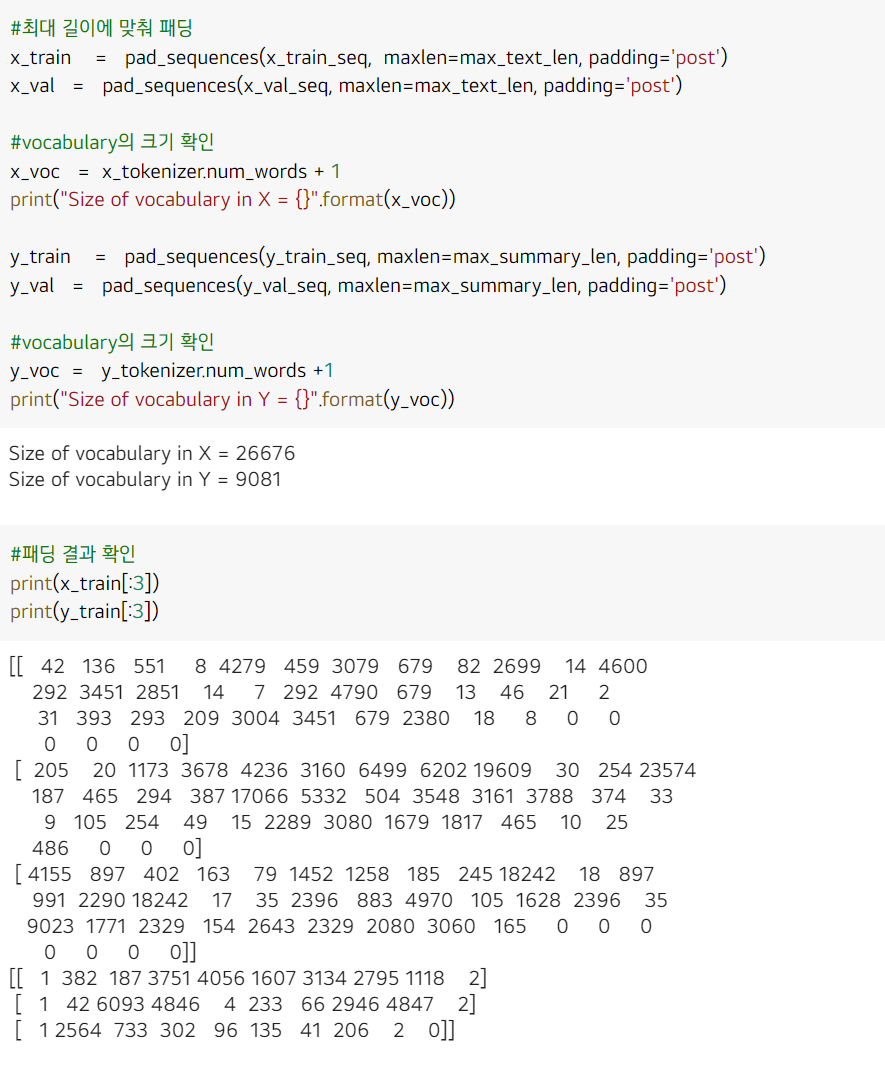


y\_train의 경우 threshold를 6으로 지정한다. 등장 빈도가 5번 이하의 단어는 전체 등장 빈도에서 약 6.0% 만을 차지하고 있다. 이를 배제하고 정수 인코딩을 진행한다.

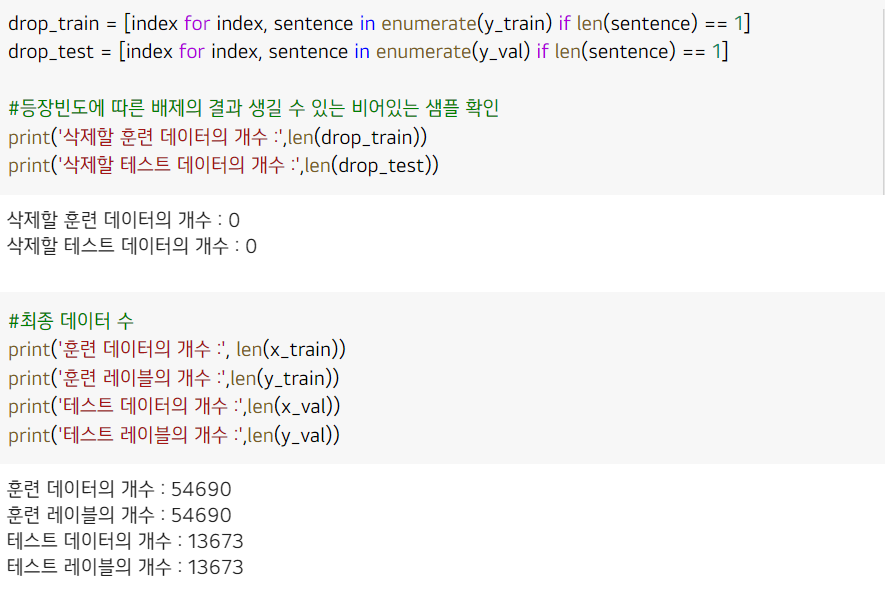


**12. 패딩**

앞서 지정한 최대 길이에 맞추어 train 데이터와 test 데이터에 대해 패딩 작업을 수행한다. 패딩 작업을 통해 최대 길이보다 짧은 문장의 경우 빈 단어는 0으로 채워진다.

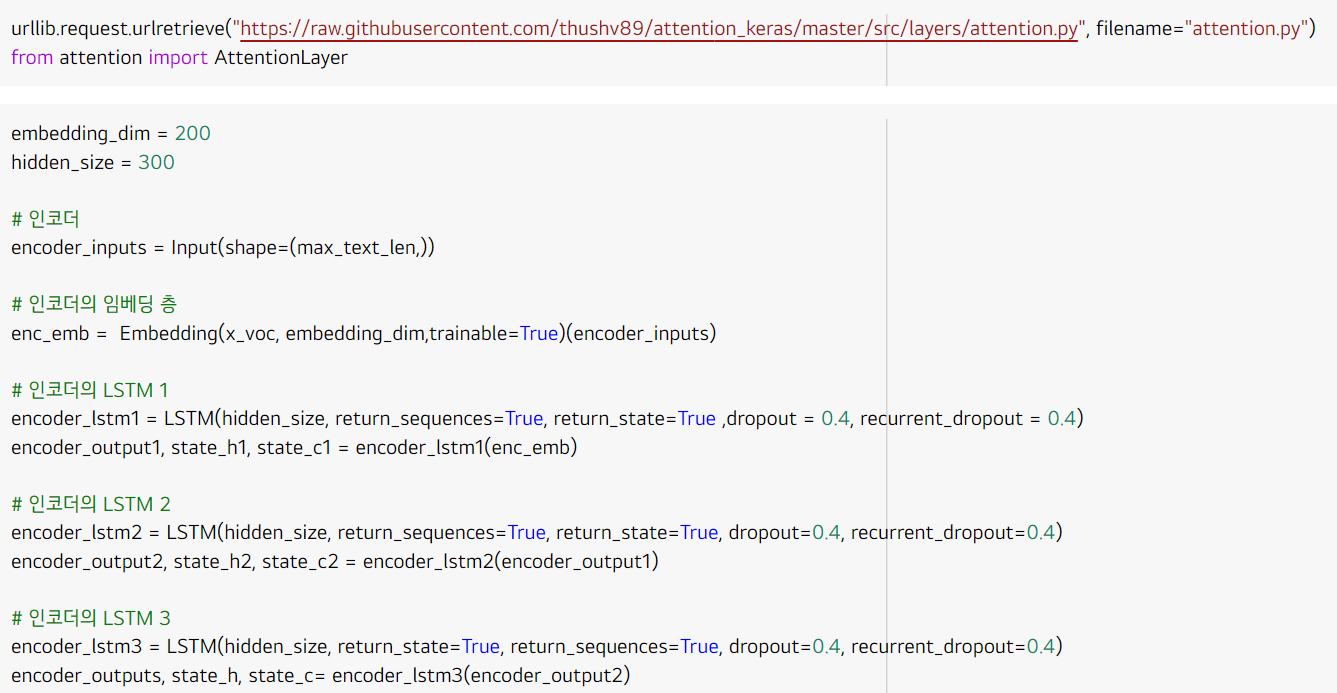


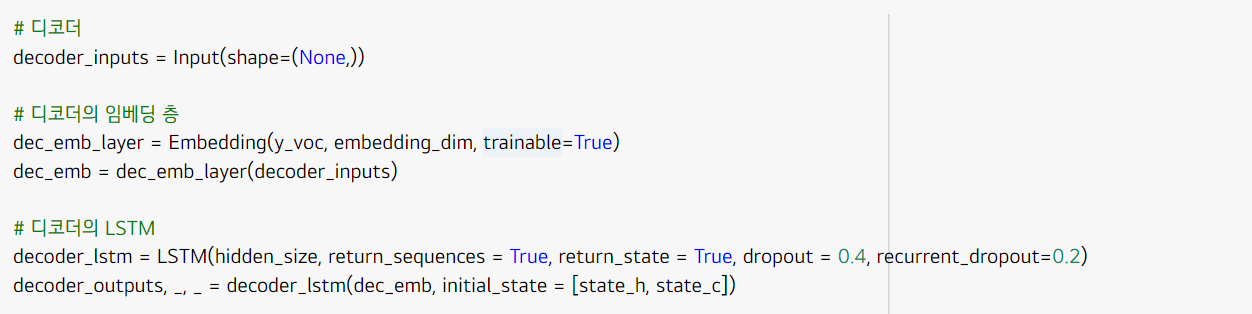
정수 인코딩과 패딩 작업을 수행한 이후 누락된 데이터를 확인하고 train 데이터와 test 데이터의 개수를 확인한다.

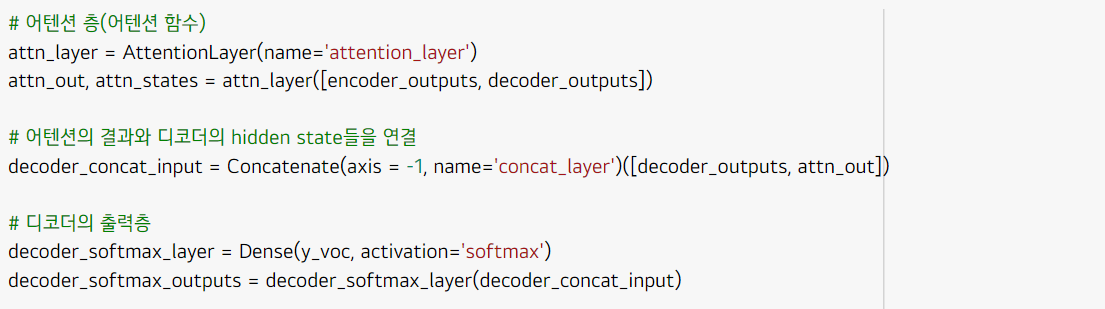


**13. 모델 설계 및 학습**

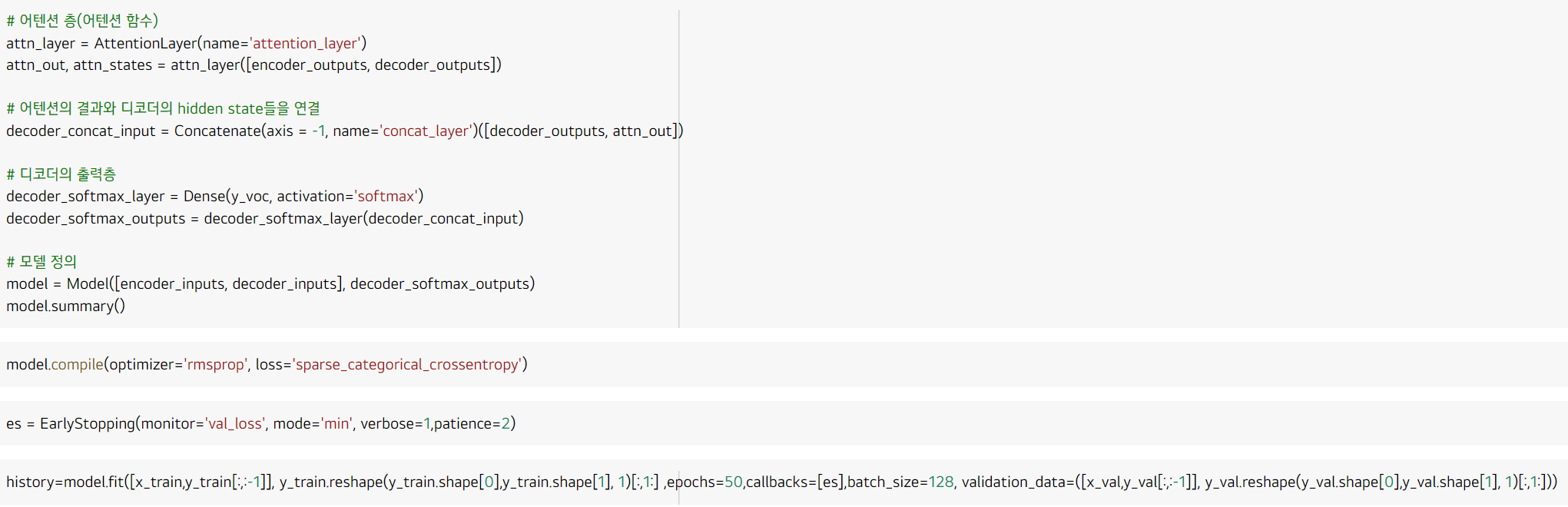
임베딩층의 크기와 은닉 층의 크기를 결정하고 인코더를 설계한다. 인코더의 경우 LSTM 층을 3개 쌓아 설계한다. 디코더 설계에서 어텐션 메커니즘을 결합하기 위해 오픈소스를 사용한다. 디코더의 설계에서 초기 상태는 인코더의 상태를 준다. 어텐션 함수를 정의하고 어텐션의 결과와 디코더를 연결한다. 이후 소프트맥스 함수를 이용한 디코더의 출력층을 설계한다.





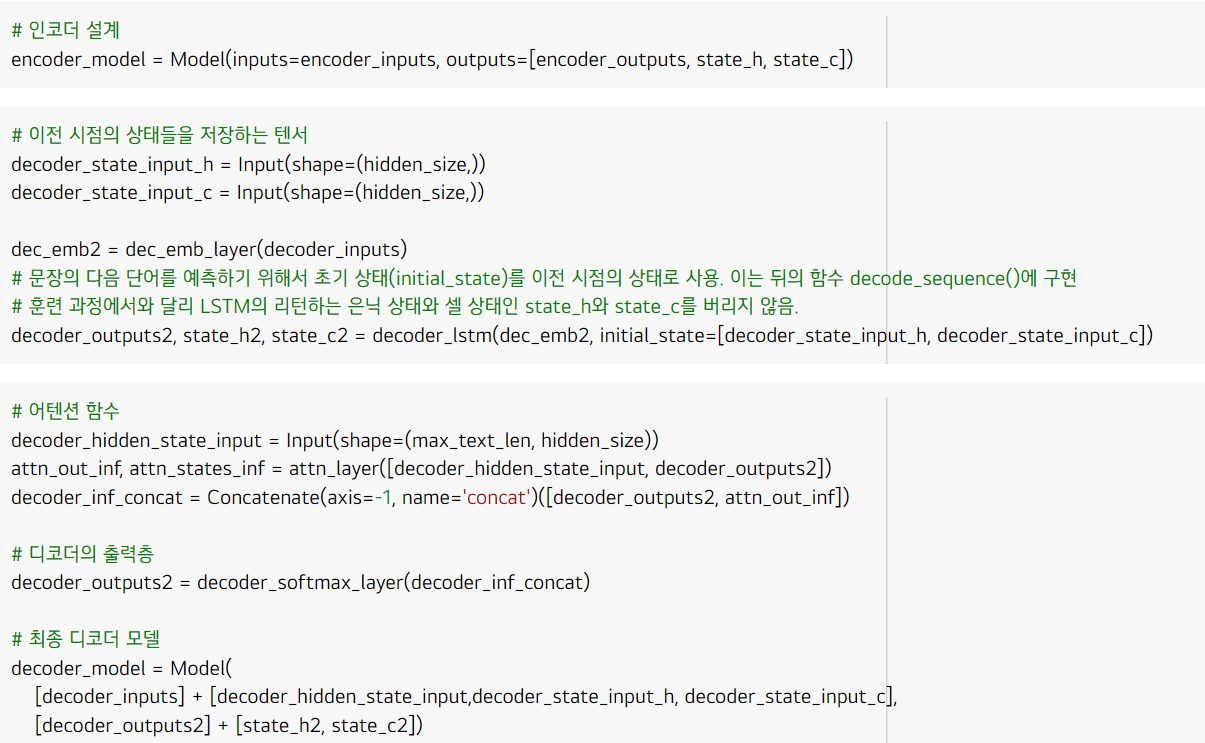


모델을 정의하고 컴파일한다. 조기 종료 조건을 설정하여 학습과정에서 과적합이 발생하는 경우를 방지하고 모델을 학습힌다.

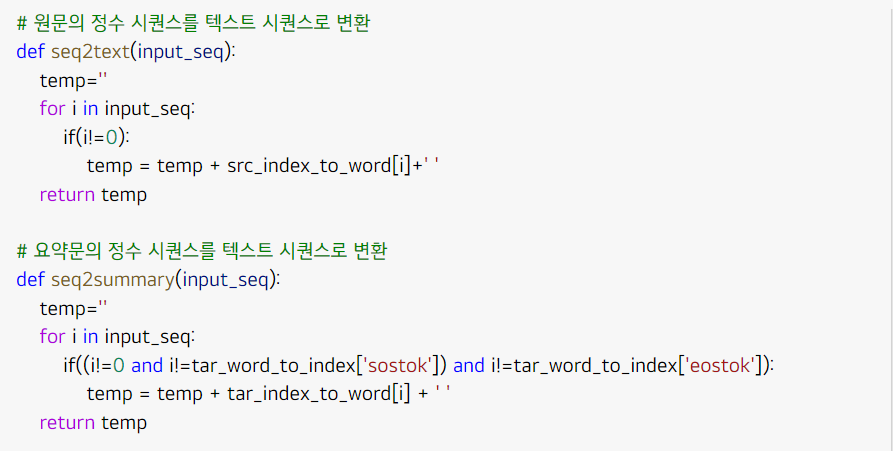


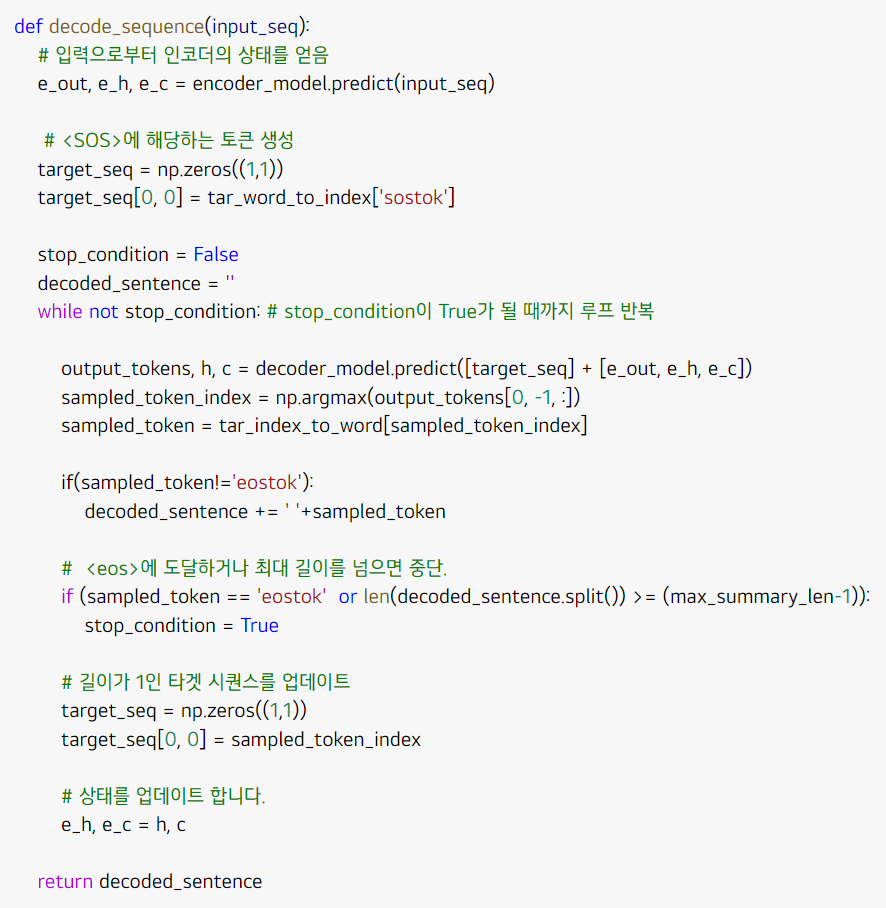
**14. 모델 테스트**

Seq2Seq 모델의 경우 학습 단계와 테스트 단계가 서로 다르기 때문에 테스트 단계의 모델을 별도로 설계한다. 새로운 인코더를 설계한 이후 어텐션 함수를 적용한 디코더 출력층을 설계한다.

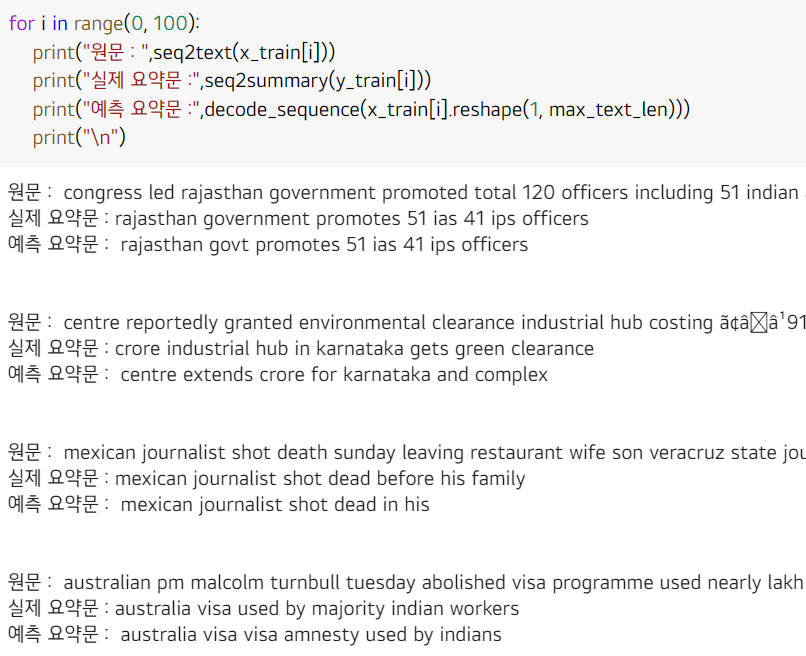


학습된 결과를 확인하기 위해 정수 시퀀스로 변환된 원문과 요약문을 다시 텍스트 시퀀스로 변환하는 함수를 정의한다. 모델이 예측하여 요약한 요약문을 얻기 위해 인코더의 상태를 얻어와 <sostok>를 시작으로 단어를 예측하고 <eostok>에 도달하거나 최대 길이가 넘는 경우 중단하여 요약문을 생성하고 출력하는 함수를 정의한다.





정의된 함수를 이용하여 최종 결과물을 확인한다.



**15. 보완점 및 느낀 점**

어텐션 메커니즘을 활용하여 Seq2Seq 모델을 사용할 경우 어느정도 개선 방향을 찾을 수 있을 알았으나 여전히 한계점이 존재한다.

모델 선정 과정에서 BERT를 비롯하여 다른 모델에 대한 방법도 생각하였으나 학습과정에서 Colab을 사용할 경우 제한된 세션 시간으로 인해 무거운 모델보단 상대적으로 가벼운 모델을 선정하기로 결정하였다. 또한 본인을 비롯하여 팀원 모두 이번 프로젝트를 통해 자연어 처리를 처음 접해 자연어 처리에 대한 이해도가 부족한 상태에서 고수준의 학습이 이루어진 모델을 선정하는 것 보단 자연어 처리에 대한 이해도를 쌓아가며 프로젝트를 진행하기 위해 대표적인 모델로 Seq2Seq를 선정하였다.

정해진 기간 내에 결과물을 도출하기 위해 여러 오픈소스를 참고하며 진행하였다. 프로젝트를 진행하고 나서 자연어 처리와 모델 학습에 대한 이해도를 쌓아왔기 때문에 좀 더 높은 수준에 다가가지 못한 점이 아쉽다.

**16. 참고자료**

<https://wikidocs.net/book/2155> 딥러닝을 이용한 자연어 처리 입문

<https://tykimos.github.io/2019/01/22/colab_getting_started/> 코랩 시작하기

<http://www.dreamy.pe.kr/zbxe/CodeClip/3769485> 코랩 파일 업로드/다운로드

<https://www.kaggle.com/sunnysai12345/news-summary> News\_summary

<https://www.kaggle.com/harshavemu/text-summarization-with-seq2seq-model> text summarization

<https://hiddenbeginner.github.io/deeplearning/2019/09/22/optimization_algorithms_in_deep_learning.html> 딥러닝 옵티마이저 총정리

<https://didu-story.tistory.com/27> 손실함수 종류

<https://brunch.co.kr/@kakao-it/155> [카카오AI리포트] 신경망 번역 모델의 진화 과정