안녕하세요. 지금부터 런닝머신 조의 헤이 런닝머신, 이 노래 반응이 어때?의 발표를 시작하겠습니다.

저희는 이 슬라이드와 같이 역할 분담을 하였습니다.

다음과 같은 순서로 발표를 진행하도록 하겠습니다.

저희는 sound cloud 사이트에 있는 음반에 대해 사람들이 어떻게 생각하는지 간편하게 알아보고자 이 주제를 선정하게 되었습니다. 기존의 sound cloud 경우, 특정 음악에 대해 개개인의 평점이 표시되지 않고 단순히 comment만 남길 수 있기 때문에, 해당 음악에 대해 어떻게 생각하는지 간단하게 파악하는 것이 어려웠습니다. 이를 개선하여, sound cloud 음반들의 감정별 리뷰를 확인하고, 요약된 리뷰를 확인하는 모델을 생성하는 것을 최종 주제로 선정하게 되었습니다.

프로젝트를 시작하면서 sound cloud는 댓글 관련 API를 제공하지 않는다는 제약이 있어, 직접 음악들에 관한 댓글들을 크롤링 하기로 하였습니다. 또한, 요약 모델을 만들기 위해서 사전에 요약된 텍스트 데이터가 존재해야 하는데, 이를 해결하기 위해 아마존 음반 리뷰 데이터를 학습 데이터로 사용하였습니다.

이 때, 아마존 데이터에서 사용자들이 사용한 단어들과 sound cloud 사용자가 사용하는 단어가 비슷할 것이라는 가정을 하였습니다.

이 프로젝트에서, 이진 분류 모델 생성 시, 긍정적인 리뷰가 부정적인 리뷰 대비 많으므로, 이들의 비율을 맞춰주는 것이 필요하고, 요약 모델 생성 시 리뷰에 쓰인 단어들로 단어 사전을 만들 때 자주 사용되지 않은 단어들에 대해 처리하는 과정이 요구됩니다.

아마존 리뷰 데이터는 아마존에서 판매하는 음원에 대해 음원 구매자가 리뷰를 올리는 것이고, sound cloud는 음원에 대해 단순 comment 성격이 강하기 때문에 두 데이터로부터 생성되는 단어 사전이 다를 것이고, 아마존 리뷰 데이터에 없는 음원에 관하여 결측값 문제 발생을 우려하였습니다.

이 모델을 통해 얻고자 하는 최종 목표는 사람들이 sound cloud 음악에 대해 어떻게 생각하는 지를 요약된 정보로 빠르게 파악하고, 감정별 리뷰를 확인하는 것입니다.

저희는 학습 데이터로 아마존 물품 리뷰 데이터의 여러 카테고리 중 디지털 음반 리뷰 데이터를 이용하였습니다. 이 데이터는 image, overall, vote 등 여러가지 feature를 가지고 있지만, 저희는 판매하는 음반에 대한 개개인의 평점인 overall과 리뷰 데이터인 reviewText 그리고 이를 요약한 데이터인 summary feature를 사용하여 모델을 생성하였습니다. sample수는 총 약 158만개 이고, 이 중 6만개를 사용하기로 했습니다.

테스트 데이터로는 sound cloud에 있는 음악 중 party girl 이라는 음악에 대한 댓글들을 크롤링 하였습니다. 테스트 데이터의 feature로는 리뷰 텍스트가 있고, 이 음악의 리뷰 개수는 4250개 였습니다.

sound cloud의 댓글들을 크롤링하기 위해 웹페이지 소스 보기를 통해 사이트의 HTML이 어떤 구조를 가지고 있는지 살펴보고, 댓글들은 뷰티풀 솦을 사용하여 특정 태그를 추출하고 따로 저장하였습니다.

이후 아마존 digital music 리뷰 데이터와 party girl 음악의 댓글들을 전처리 하였습니다. 이들의 전처리 방법으로는 null값을 제거하였고, 영어 알파벳의 대문자는 소문자로 전환하고 불용어를 모듈 nltk를 이용하여 제거하고, 이모티콘을 제거하였습니다. 또한, 원문 중 중복되는 글을 제거하고 표제어를 추출하였습니다.

저희가 설계한 모델은 다음과 같이 sound cloud에 업로드된 음악의 댓글들을 인풋으로 넣으면, 미리 학습한 이진 분류 모델로 긍정, 부정을 분류합니다. 그 다음 모델이 예측한 클래스의 댓글들을 한 문장으로 요약하였습니다. 또한, 감정 클래스에 속해있는 댓글의 개수 비율을 계산하여 한 음악에 대한 전반적인 평가를 알 수 있도록 하였습니다.

먼저 이진 분류 모델을 학습하기 위해, 아마존 데이터를 이용하여, 평점에서 1점과 2점은 부정인 0으로 변환하였고, 4점과 5점은 긍정인 1로 변환하였습니다.

긍정과 부정 label을 붙이고, 데이터를 분석해본 결과, 긍정 데이터 개수는 27482개이고 부정 데이터 개수는 1432개 였습니다. 따라서 성능 평가의 신뢰도를 높이기 위하여, 긍정 부정 데이터 개수를 각각 1432개로 맞춰주었습니다.

다음으로 인풋으로 들어가는 텍스트 데이터를 벡터화 시키는 과정을 거쳤습니다.

(클릭)

분석한 총 단어 집합의 크기는 11519개 이지만, 등장 빈도가 6이하인 희귀 단어가 9701개 있어, 이를 제외한 1818개를 이용하여 단어사전을 생성하였습니다.

(클릭)

그 다음 sequence의 크기를 맞추기 위해 zeropadding을 하였습니다.

다음으로 train 할 모델은 keras의 lstm을 사용하였습니다. sequential한 모델을 생성하여 1개의 layer를 가지고 있는 LSTM을 더하였습니다. 모델 평가 방법은 cross validation을 이용하였습니다. 마지막 층에서 이진 분류를 위해 softmax function을 사용하였고,

(클릭)

loss function은 확률 값이 더 높은 label을 선택하기 위해 binary cross entropy를 선택하였습니다. 적절한 하이퍼 파라미터 값을 정하기 위해 여러가지 수를 바꿔보며 성능을 비교한 후 성능이 가장 좋은 것으로 골랐습니다.

먼저, LSTM의 activation parameter를 결정하기 위해 linear한 모델과 tanh(하이퍼볼릭 탄젠트)를 검증하였습니다. 그 결과 linear한 모델은 테스트 정확도가 0.6167인 반면에 tanh (하이퍼볼릭 탄젠트)모델의 경우 테스트 정확도가 0.8571이 나와, tanh(하이퍼볼릭 탄젠트)를 채택하였습니다.

또한, LSTM의 활성화된 숨겨진 유닛의 수가 1, 50, 100, 1818인 경우의 정확도를 비교해보았습니다. unit수가 너무 크지 않으면 비슷한 테스트 정확도를 가진다는 사실을 알았고, 이 중 0.8920으로 정확도가 가장 높았던 unit 수인 100을 선택하였습니다.

validation loss가 어느정도 줄어들다가 어느 순간 다시 커지는 상황에서 overfitting을 막기 위해 학습을 멈출 수 있는 parameter인 patience를 지정하였습니다. patience가 1,2,3,4인 경우 정확도가 비슷하게 나왔지만, 그 중 0.8955으로 정확도가 가장 높았던 patience가 3인 경우를 채택하였습니다.

앞의 과정을 통하여, 살펴본 것 중 가장 성능이 좋았던 파라미터들을 지정하여 모델을 학습하였습니다. LSTM unit수를 100으로 설정하고 activation function을 tanh(하이퍼볼릭 탄젠트)로 결정하였을 때, 0.89의 성능을 낼 수 있었습니다. patience를 3으로 설정하였을 때 다음과 같은 그래프를 얻었습니다.

그 다음 요약모델에 관한 설명입니다. 요약 모델에 필요한 하이퍼 파라미터를 설정하기 위해 아마존 reviewText 데이터에 대한 분석을 진행하였습니다. 리뷰 데이터에서 중복 데이터를 제거하고 각 문장의 길이를 측정한 후, outlier를 제외하고 boxplot으로 그려보았습니다. 리뷰데이터의 경우 3사분위 수가 30에 있고 최대 55까지 늘어납니다. 요약 데이터의 경우 3사분면의 수가 8정도에 위치해있습니다. 따라서 최대 길이 991까지 고려했을 때, 텍스트 제한 길이 하이퍼파라미터를 각각 100, 8로 지정하였습니다. 다음으로 모든 문장에서 쓰인 단어의 개수를 측정하여 단어 집합의 크기를 6500으로 지정하였습니다.

입력 시퀀스가 길어지면 출력 시퀀스의 정확도가 떨어지는 현상을 보완하기 위해 attention 함수를 이용하였고, encoder와 decoder 층 사이에 attention layer를 추가해주었습니다.

(클릭)

attention layer를 이용한 요약 최종 모델의 summary 입니다.

아마존 리뷰 데이터에서 제공한 원문과 요약문 그리고 저희 모델이 예측한 요약문의 결과입니다. 다음 5가지 예시를 보면 실제 요약문과 예측 요약문이 비슷한 것을 알 수 있습니다. 예를 들어, 실제 요약은 good song인데 예측 요약문은 great song으로 생성하였고, 실제 요약문은 love인데 예측 요약문은 love it으로 생성하였습니다.

요약 모델의 평가를 위해 저희는 ROUGE-N을 이용하였습니다. ROUGE(라우지)는 텍스트 자동 요약, 기계 번역 등 자연어 생성 모델의 성능을 평가하기 위한 지표로, 모델이 생성한 요약본을 미리 만들어 놓은 요약본과 대조해 성능 점수를 계산하는 것입니다. 기존 요약 모델 연구에서의 모델 성능을 고려했을 때, 저희 모델의 ROGUE-1는 약 20~44% 이상, ROGUE-2는 약 10~21% 이상의 값을 기대했습니다.

이 중 rouge-2는 원래 요약문과 예측 요약문이 겹치는 biagram수를 보는 지표입니다.

(클릭)

8831개의 original summary와 predict summary에 한 문장씩 적용시킨 후, rouge1과 rouge2의 값을 얻은 결과, rouge1은 약 0.21, rouge2는 약 0.15가 나왔습니다.

이진 학습한 이진 분류 모델과 요약모델을 바탕으로 sound cloud 음반 댓글을 테스트 데이터로 넣었을 때 도출한 결과입니다. 긍정문을 요약한 결과 great album, great song, five stars등의 결과가 나왔고, 부정문을 요약한 결과 one star, one, disapppointed의 결과가 나왔습니다.

도출한 결과를 실제 웹페이지에 적용한 예시입니다.

이상으로 런닝머신 조의 발표를 마치도록 하겠습니다.