산학 프로젝트

완료 보고서

9조 노딩코예

서민철	오혁진	이재민	이준희	최재혁
2016125029	2016125040	2016125057	2016125060	2016125077

한국항공대학교

프로젝트 기획

▶ 주제

- 텍스트 데이터 감정 분류

▶ 기획 의도

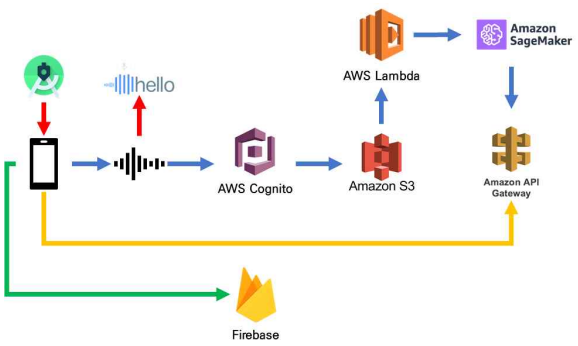
텍스트 데이터로부터 감정을 추출할 수 있을까? 이 가능성을 알아보기 위해 간단한 감정 분석을 통해 사용자에게 음악을 추천하는 어플을 기획했다. 사용자가 모바일 환경에서 해당 어플을 실행, 음성 데이터를 입력하면 문장의 텍스트로부터 현재 감정을 파악하여 분위기에 알맞은 노래를 유튜브 동영상을 통해 추천한다.

▶ 사용자 시나리오

1. 애플리케이션이 실행되면 음성 인식 버튼을 눌러 음성 녹음을 시작한다.
 + 사용자가 일정 시간 동안 음성을 입력하지 않으면 애플리케이션이 자동으로 녹음을 멈춘다.
2. 애플리케이션이 녹음된 음성 데이터를 텍스트 데이터로 전환한다.
3. 녹음이 끝나면 출력값을 받을 때까지 대기한다.
4. 출력값을 바탕으로 결과 메시지를 화면에 출력한다.
 + 긍정이면 사랑 노래, 부정이면 이별 노래를 추천한다.

Framework

▶ Framework



- 안드로이드 클라이언트는 Android Studio로 개발했다.
- 클라이언트에서 Google speech-to-text를 이용해 음성 변환을 진행한다.
- 클라이언트와 Amazon S3를 연결하기 위해 AWS Cognito를 사용했다.
- Amazon S3에 사용자가 음성으로 입력한 텍스트 데이터가 파일 형태로 저장된다.
- AWS Lambda가 S3의 객체 생성 이벤트를 감지하고 S3에 새로 생성된 객체에 접근한다.
- 텍스트 파일 내부의 텍스트 데이터를 읽어와 SageMaker에서 배포 중인 학습 모델로 전달한다.
- SageMaker에서 확률값을 반환하면 해당 값을 Amazon API Gateway Endpoint로 배포한다.
- 안드로이드 클라이언트가 Amazon API Gateway Endpoint에 접근해 확률값을 가져온다.
- 여러 개인 데이터들은 Firebase에 저장되어 필요할 때 시각화 자료로 사용된다.

상세개발 내용

▶ Android Studio

- 유저 접근성 및 편의성 부분에서 웹보다는 어플이 더 효율적일 것으로 판단했다.
- 어플 개발 환경으로 Android Studio를 선택했다.

▶ Google speech-to-text

- 유저의 음성 데이터를 텍스트로 변환하기 위해 사용했다.
- 사용자가 말한 문장을 눈으로 확인할 수 없었기 때문에 문장을 잘못 인식하는 때도 있었다.
- 실시간 번역 기능을 추가하여 사용자가 문장이 제대로 입력되고 있는지 확인하기 쉽게 했다.

▶ AWS Cognito

- AWS S3와 안드로이드 클라이언트를 연결하기 위해 사용했다.

▶ Amazon S3

- 유저의 음성 데이터가 변환된 텍스트 데이터를 저장한다.

▶ AWS Lambda

- AWS Lambda는 이벤트 트리거 기능을 제공한다.
- S3와 SageMaker 사이의 데이터 순환을 자동화시키기 위해 사용했다.

▶ Amazon SageMaker

- 문장에서 감정을 추출할 모델을 만들기 위해 선택한 머신러닝 개발 플랫폼이다.
- AWS Lambda와의 호환성이 뛰어나 사용했다.
- Jupyter Notebook의 복잡한 환경 구축 과정을 편리하게 지원받아 구현할 수 있었다.
- 초반에는 학습 데이터 양의 부족으로 모델의 성능이 완벽하게 나오지는 않았다.
- 이후 학습 알고리즘을 수정하여 기존의 데이터 양으로도 정확도를 높일 수 있었다.

▶ AWS API Gateway

- AWS Lambda가 SageMaker에서 받아온 확률 값을 Endpoint로 배포하기 위해 사용했다.
- 안드로이드 클라이언트에서는 해당 Endpoint에 접근하여 원하는 데이터를 받아온다.

▶ Firebase

- 유저의 누적 분석 내역을 저장하기 위해 사용했다.
- 저장되는 데이터로는 날짜, 감정 확률, 분석 결과 등이 있다.
- 위 정보들은 그래프 등으로 시각화되어 사용자에게 제공된다.

학습 알고리즘 변화 과정

▶ 첫 번째 시도

- 각각의 문장이 감정을 지닌 말로 생각 → 주제에 맞는 문서 분류와 동일한 문제로 판단
- 나이브 베이즈 알고리즘을 이용하여 문장을 각각의 감정으로 분류

▶ 첫 번째 시도: 결과

- 실패, 나이브 베이즈 알고리즘을 구현한 scikit-learn library를 사용하려 했다.
- 그러나 입력 데이터의 형태를 맞추기가 어려웠다.
- 근본적으로 단어 수만을 가지고 감정을 분류하기에는 무리가 있다고 판단했다.

▶ 첫 번째 시도: 해결책

- RNN 방법을 이용하기로 했다.
- 임베딩 벡터를 이용하여 단어의 유사도가 결정된다.
- 유사도를 이용하여 문맥을 통해 감정을 분석할 수 있다.

▶ 두 번째 시도

- RNN 방법을 적용, Tensorflow library 사용
- 네이버 영화리뷰 감정 분석 클론 코딩, COLAB을 이용하여 협업

▶ 두 번째 시도: 결과

- 실패, 정확도 50%, 훈련 중 loss 값이 음수
- loss 값은 훈련 시 결과와 목표의 차이 (음수는 비정상)

▶ 두 번째 시도: 해결책

- Tensorflow 모델 구축 파라미터와 사용 알고리즘 파라미터 수정

▶ 세 번째 시도

- 다진 분류 알고리즘 파라미터로 수정 (클론 코딩한 네이버 영화 리뷰 감정 분석은 이진 분류)

▶ 세 번째 시도: 결과

- 정확도 70%, 하지만 비정상적인 분석
- 원인: 데이터, 적절하지 않은 Tensorflow 모델 구현, 하이퍼 파라미터

▶ 세 번째 시도: 해결책

- 데이터 정리 작업 + Tensorflow 모델 수정 + 하이퍼 파라미터 수정

▶ 네 번째 시도

- 각 레이블에 대한 데이터 균등화, 높은 성능의 모델 구축을 위해 여러 가지 모델을 훈련
- 각 훈련에 대한 하이퍼 파라미터를 여러 번 수정해보며 성능 비교

▶ 네 번째 시도: 결과

- 최대 85%의 정확도, 아직도 비정상적으로 분석
- 임베딩 벡터에 대한 의문이 생김

▶ 다섯 번째 시도

- 임베딩 벡터를 공부하여 상황을 파악
- 미리 훈련된 임베딩 벡터를 가져와 Embedding 층에 주입

▶ 다섯 번째 시도: 결과

- 정확도 90%대, 꽤 정확한 분석
- 몇몇 데이터에 대해서는 분석해내지 못함
- 너무 적은 훈련 데이터, 텍스트 데이터의 한계

개발 추진 실적

► 2020.3.30. (3주차)
▷ 서민철, 오혁진
- S3와 Android Studio 연동 (진행)

▷ 이재민
- Android 음성 인식 기능 구현 (진행)

▷ 이준희
- 학습 및 테스트 데이터 수집 및 제작 (진행)

▷ 최재혁
- 데이터 준비와 수집 및 Sagemaker 구조 이해 (진행)

► 2020.4.06. (4주차)
▷ 서민철, 오혁진
- S3와 Android Studio 연동 (완료)
- S3의 데이터 클라이언트로 전송 (진행)

▷ 이재민
- Android 음성 인식 기능 구현 (완료)
- cloud-speech-to-text API 연동 (진행)

▷ 이준희
- 학습 및 테스트 데이터 수집 및 제작 (완료)
- 데이터 분석(구조, I/O의 관계) 및 전처리(구문, 의미 분석) (진행)

▷ 최재혁
- 데이터 준비와 수집 및 Sagemaker 구조 이해 (완료)
- 데이터 전처리와 가공 (진행)

► 2020.4.13. (5주차)
▷ 서민철, 오혁진
- S3의 데이터 클라이언트로 전송 (진행)

▷ 이재민
- cloud-speech-to-text API 연동 (완료)
- 음성 데이터를 텍스트로 변환 (진행)

▷ 이준희
- 데이터 분석(구조, I/O의 관계) 및 전처리(구문, 의미 분석) (진행)

▷ 최재혁
- 데이터 전처리와 가공 (진행)

► 2020.4.20. (6주차)
▷ 서민철, 오혁진
- S3의 데이터 클라이언트로 전송 (완료)
- 인터페이스 제작 (진행)

▷ 이재민
- 음성 데이터를 텍스트로 변환 (완료)
- 텍스트 데이터 S3로 전송 (진행)

▷ 이준희
- 데이터 분석(구조, I/O의 관계) 및 전처리(구문, 의미 분석) (진행)

▷ 최재혁
- 데이터 전처리와 가공 (진행)

► 2020.4.27. (7주차)
▷ 서민철, 이재민
- AWS Lambda 이해 (완료)
- S3와 Lambda 연동 (완료)
- SageMaker와 Lambda 연동 (진행)

▷ 오혁진
- 기초 인터페이스 제작 (진행)
- S3의 데이터를 이용해 음악 분류 (진행)

▷ 이준희
- 데이터 분석(구조, I/O의 관계) 및 전처리(구문, 의미 분석) (완료)
- 전처리된 데이터 레이블링 (완료)
- 학습 및 테스트 데이터 분할 (진행)

▷ 최재혁
- 데이터 전처리와 가공 (완료)
- 모델링 및 모델링 수정 (진행)

► 2020.5.04. (8주차)
▷ 서민철, 이재민
- SageMaker와 Lambda 연동 (완료)
- S3 이벤트 트리거 설정 (완료)
- API Gateway Endpoint 배포 (완료)
- 버그 수정 및 테스트 (진행)

▷ 오혁진
- 기초 인터페이스 제작 (진행)
- S3의 데이터를 이용해 음악 분류 (완료)
- APK 파일 제작 (진행)

▷ 이준희
- 학습 및 테스트 데이터 분할 (진행)

▷ 최재혁
- 모델링 및 모델링 수정 (완료)
- Android Studio 연동을 위한 배포 (진행)

► 2020.5.11. (9주차)
▷ 서민철
- 버그 수정 및 테스트 (진행)

▷ 이재민, 최재혁
- 모델링 및 모델링 수정 (진행)

▷ 오혁진
- 기초 인터페이스 제작(완료)
- 감정 데이터를 이용해 음악 분류 수정 (진행)

▷ 이준희
- 학습 및 테스트 데이터 분할 (진행)

► 2020.5.18. (10주차)
▷ 서민철
- 버그 수정 및 테스트 (진행)

▷ 이재민, 최재혁
- 모델링 및 모델링 수정 (완료)
- Android Studio 연동을 위한 배포 (진행)

▷ 오혁진
- 감정 데이터를 이용해 음악 분류 수정 (진행)

▷ 이준희
- 학습 및 테스트 데이터 분할 (진행)

► 2020.5.25. (11주차)
▷ 서민철
- 버그 수정 및 테스트 (진행)

▷ 최재혁
- Android Studio 연동을 위한 배포 (진행)

▷ 이재민
- Firebase Database 사용자 계정 및 기록 추가 (진행)

▷ 오혁진
- 감정 데이터를 이용해 음악 분류 수정 (진행)
- Firebase Database 연동 (완료)

▷ 이준희
- 학습 및 테스트 데이터 분할 (진행)

► 2020.6.01. (12주차)
▷ 서민철
- 버그 수정 및 테스트 (완료)
- 개요 작성 (완료)
- 유저 시나리오 구상 (진행)

▷ 최재혁
- Android Studio 연동을 위한 배포 (진행)

▷ 이재민
- 추가 기능 구현(진행)

▷ 오혁진
- S3의 데이터를 이용해 음악 분류 수정 (진행)
- 유저별 사용기록 Database에 저장 (완료)
- 추가 기능 인터페이스 제작(진행)

▷ 이준희
- 학습 및 테스트 데이터 분할 (진행)

► 2020.6.08. (13주차)
▷ 서민철
- 유저 시나리오 구상 (완료)
- 전체 프레임워크 제작 (완료)
- 상세개발계획서 요약 (완료)
- 최종 발표 PPT 제작 (진행)

▷ 최재혁
- Android Studio 연동을 위한 배포 (완료)
- 버그 수정 및 테스트 (진행)

▷ 이재민
- 추가 기능 구현(완료)

▷ 오혁진
- S3의 데이터를 이용해 음악 분류 수정 (완료)
- APK 파일 제작 (진행)
- 추가 기능 인터페이스 제작(완료)

▷ 이준희
- 학습 및 테스트 데이터 분할 (완료)
- 버그 수정 및 테스트 (진행)

► 2020.6.15. (14주차)
▷ 서민철
- 최종 발표 PPT 제작 (완료)

▷ 오혁진, 이재민, 이준희, 최재혁
- 버그 수정 및 테스트 (완료)

문제점 및 해결 내용

► Dynamo DB를 사용하지 못한 이유

초반 기획 단계에서는 클라이언트를 AWS와 연결할 때, AWS Amplify의 기능이 필수적으로 필요하지는 않았기 때문에 Amplify 대신 Cognito를 사용했다. 그러나 이후 추가 기능을 구현하기 위해 Dynamo DB를 사용하게 됐는데, 이를 사용하기 위해서는 Amplify가 필요했다. 인제 와서 다시 연결 구조를 Amplify로 바꾸기에는 시간이 부족할 것으로 판단했고, 비슷한 역할을 해줄 수 있는 Firebase를 사용했다. Dynamo DB를 사용했다면 모든 플랫폼을 AWS로 통일할 수 있었으나 그렇게 하지 못한 부분이 아쉽다.

► 부족한 학습 데이터로 인한 낮은 정확도

학습 알고리즘을 수정하여 같은 데이터양으로도 정확도를 높일 수 있었다.

계획 대비 효과 자체 평가

► 텍스트로부터 감정을 분석하는 것이 가능할까?

가능하다. 현재는 감정의 종류가 긍정, 부정, 부적합 3가지로만 분류되지만 충분한 데이터셋과 성능 좋은 학습 알고리즘, 이를 개발할 시간만 충분하다면 감정의 종류를 보다 세분화시킬 수 있을 것이다.

► 감정 분석을 어떻게 활용할 수 있을까?

불특정 다수의 반응을 관찰하고 알맞은 판매 전략을 세워야 하는 마케팅 분야나 사람의 감정과 심리 등의 분석이 필요한 의료 및 범죄 예방 등의 여러 분야에서 분명 유의미하게 쓰일 것이다.

► 현재 개발한 어플의 실효성 및 목적성

프로젝트 자체가 감정 분석의 가능성 유무를 판단하기 위한 목적으로 진행됐기 때문에 현재 개발한 어플 자체의 실효성과 목적성은 크게 없는 것이 사실이다. 그러나 충분한 시간과 데이터만 있다면 현재 프로젝트를 훨씬 발전시킬 수 있고, 앞서 구상한 아이디어를 실현할 수 있다는 것을 알았기 때문에 팀 자체적으로는 굉장히 만족하고 있다.