Pusan National University Computer Science and Engineering Technical Report 2023-10

멀티모달 기반 스팸필터링 플랫폼 개발

박혜경

이승현

천영채

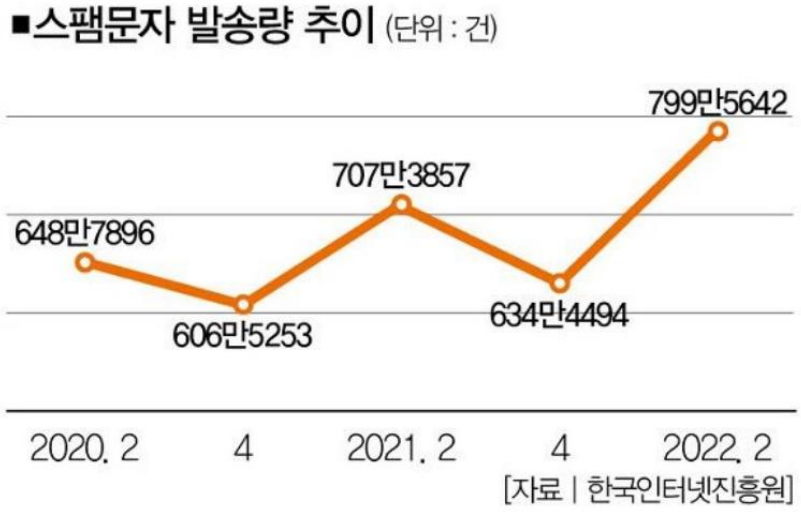
지도교수 최윤호 교수님

목 차

1. 서론

1.1. 연구 배경

스팸은 요청하지 않은 메시지(이메일, 문자, 또는 전화)를 대량으로 불특정 다수에게 전송하는 활동을 의미한다. 이러한 스팸 활동은 인터넷 기술의 진보로 더욱 빈번해지고 있으며, 이로 인해 개인정보 침해와 스팸 문제가 더욱 악화되고 있다. 한국인터넷진흥원의 2020년 조사에 따르면 약 800만 건의 스팸 문자가 발송되었다.



특히, 최근 몇 년 동안 정부의 재난 지원금이나 손실 보상과 관련된 스팸이 금융회사를 사칭하여 대출을 제안하는 형태로 증가하고 있다. 이러한 스팸은 과거의 스팸과 달리, 더욱 악의적인 목적을 가지고 개인정보를 탈취하고 데이터를 삭제하며 금전을 요구하는 경향이 있다.

이러한 악의적인 스팸은 사용자 경험을 저해하고 불편함을 초래할 뿐만 아니라, 바이러스에 감염될 가능성도 내포하고 있다. 이러한 문제는 사용자에게 시간 낭비를 초래하며, 더 나아가 개인정보 유출 및 금전 손실과 같은 심각한 결과를 초래할 수 있다.

1.2. 기존 문제점

기존의 스팸 필터링 시스템은 이미 신고된 스팸 데이터베이스와 비교하여 동일한 문자나 이미지를 필터링하는 방식을 사용한다. 그러나 이 방식은 스팸 발신자가 문구를 미묘하게 수정하거나 특수문자를 사용하는 경우, 정확한 판별이 어렵다. 한편, 딥러닝을 기반으로 한 스팸 필터링은 인공지능이 반복적인 학습을 통해 단어 패턴을 인식하

고 스팸을 식별하는데 사용된다. 이를 통해 스팸 필터링의 정확도를 높일 수 있어, 더욱 정확한 스팸 필터링이 가능하다.

그러나 주로 필터링 플랫폼이 문자 메시지, 이메일과 같은 텍스트 데이터에 집중되어 있기 때문에, 이미지 스팸이나 보이스 피싱과 같은 다른 데이터 유형에 대한 스팸 필터링은 여전히 어려운 과제로 남아있다.

1.3. 연구 목표

1.3.1. 다양한 데이터 타입에 대한 스팸 필터링 기능

현재, SNS 및 보안이 취약한 웹페이지의 회원가입 등으로 인해 개인정보가 널리 유출되고 있는 상황이다. 이로 인해 많은 사람들이 개인 이메일 주소, 휴대폰 번호 등을 통한 스팸 메시지를 받고 있으며, 이는 개인에게 불필요한 내용을 전달하는 것뿐만 아니라 보이스피싱과 같은 사기나 범죄의 피해자가 될 수도 있는 심각한 문제이다. 이러한 문제를 예방하기 위한 효과적인 대응책이 필요한 시점이다.

스팸 메시지를 효율적으로 차단하기 위해서는 텍스트뿐만 아니라 이미지, 음성 등 다양한 형태의 데이터로 존재하는 스팸 메시지를 식별할 수 있어야 한다. 이러한 데이터 유형에 따라 다양한 형태의 스팸이 존재할 수 있으며, 이를 구별하는 능력이 필요하다. 데이터 유형과 형태는 다음과 같다.

- 텍스트
  - 이메일
  - 문자 메시지
  - 채팅
- 이미지
  - 이미지 첨부파일
  - SNS의 프로필 이미지
- 음성
  - 음성 메시지
  - 통화

성공적인 스팸 필터링을 위해서는 기계학습을 활용하여 다양한 스팸 데이터를 학습하

고, 새로운 데이터가 주어졌을 때 스팸 여부를 정확하게 판단해야 한다. 또한 사용자가 설정한 키워드와 규칙에 따라 스팸으로 분류하는 규칙 설정이 가능해야 한다.

딥러닝과 머신러닝을 기반으로 하여 위와 같은 조건들을 충족하는 효과적인 스팸 필터링 모델을 개발해 나갈 필요가 있다.

1.3.2. 스팸 필터링 플랫폼 제공

서비스 제공 웹페이지를 통해 사용자로부터 텍스트, 이미지, 음성 데이터를 입력 받는다. 입력받은 데이터에 대한 스팸 필터링 결과를 시각적으로 제공하기 위해 막대 그래프를 활용하여 웹페이지에 나타낸다. 또한 사용자가 결과를 쉽게 확인하고 이해할 수 있도록 스팸 또는 햄(정상 메시지)으로의 분류를 시각적으로 표현한다.

스팸 또는 햄으로 분류된 데이터에 대한 설명과 정보를 판단 근거로서 제공하기 위해 다음과 같은 기능을 웹페이지에 포함시키고자 한다.

- **하이라이팅**

  스팸 의심 단어를 강조하고 하이라이팅하여 사용자에게 어떤 부분이 스팸으로 분류되었는지 시각적으로 보여준다. 이를 통해 사용자는 스팸 여부를 더 잘 이해할 수 있다.

- **의심 확률**

  입력 데이터의 각 단어가 스팸 또는 햄일 확률을 백분율로 표시하여 제공한다. 이는 사용자에게 스팸 여부를 정량적으로 전달하여 이해를 돕는다.

- **그래프 및 차트**

  스팸 필터링 결과를 보다 시각적으로 나타내기 위해 막대 그래프 또는 차트를 활용하여 스팸 및 햄 데이터의 비율을 시각적으로 보여준다.

- **사용자 대화형 기능**

  사용자가 스팸 분류 결과에 대한 피드백을 제공하고 신뢰성 있는 모델을 개선하는 데 기여할 수 있도록 사용자 대화형 기능을 구현한다.

이러한 기능을 웹페이지에 효과적으로 통합하여, 사용자가 스팸 필터링 결과를 이해하고 관리하기 쉽도록 도와준다.

2. 연구 배경

2.1. 연구 개발 언어

주 개발 언어로는 Python를 주로 사용했으며, UI 구현 부분에는 JavaScript가 함께 사용되었다. Python은 모델 학습, 자연어 처리, API 연동에 사용하였다.

2.2. 연구 개발 도구

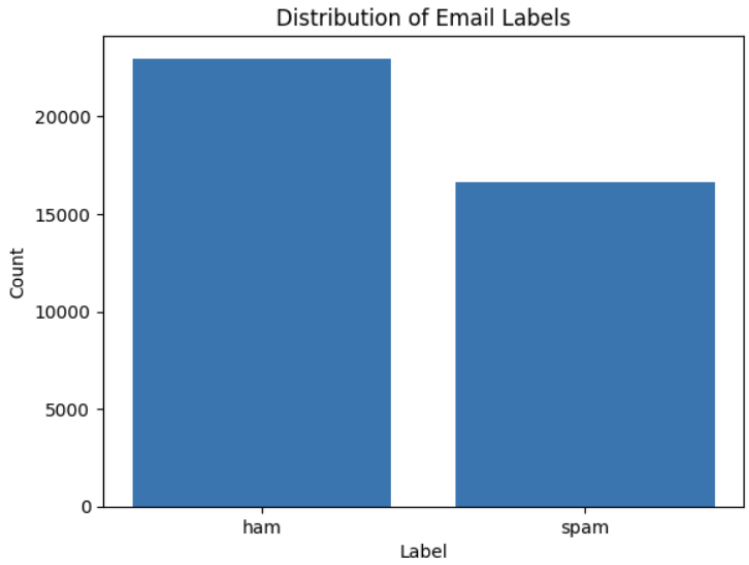
개발 도구로는 프론트엔드에는 React를, 백엔드는 Flask를 사용하였다. 스팸 필터링에 필요한 모델 학습에는 Scikit-learn, tensorflow, nltk, cuda가 사용되었다. 특히 tensorflow의 경우 gpu사용을 위해 2.11보다 이전에 배포된 버전을 사용하였다. Image Caption 생성 모델의 학습에 사용된 GPU는 NVIDIA GeForce RTX 3060을 사용하여 진행하였다.

2.3. 데이터 분석

스팸 데이터의 형식을 텍스트, 이미지, 음성 3가지로 고려하여 각 데이터의 특성을 반영할 수 있도록 고안하였다

2.3.1. 텍스트 데이터

텍스트 데이터의 경우 스팸 분류 모델의 학습에 사용한 데이터는 Enron-Spam Dataset, SMS Spam Collection, SpamAssassin Dataset을 사용하였다. 세 데이터를 모두 종합하여 총 39,610개의 텍스트 데이터를 학습에 사용하였다.



Enron-Spam, SMSSpamCollection, Sapm Assassin이 모두 포함된 전체 종합 데이터

Enron-Spam Dataset은 V. Metsis, I. Androutsopoulos 및 G. Paliouras가 수집한 데이터로 "Spam Filtering with Naive Bayes - Which Naive Bayes?"에 언급되어 있다. 해당 데이터는 본래 총 33,716개의 전자 메일로 이루어져 있다. 그 중 스팸("spam")은 17,171개, 비스팸("ham")은 16,545개이다. 그러나 실제로 다운로드하여 사용한 데이터는 총 30,494개의 이메일로 구성되어 있었으며 스팸과 비스팸의 비율은 48% : 52%로 비교적 비슷한 비율을 포함하고 있었다.

SMS Spam Collection은 휴대 전화 스팸 조사를 위해 수집된 SMS 레이블 메시지의 공개 데이터셋이다. 총5574개의 SMS 메시지로 이루어져 있다. 스팸("spam")과 비스팸("ham")의 비율은 13% : 87%로 비스팸이 스팸보다 약6배 정도 많은 비율로 구성되어 있다.

SpamAssassin Dataset은 Apache **SpamAssassin** Project에서 공개적으로 제공한 데이터셋이다. 총 6047개의 이메일로 이루어져 있다. 스팸과 비스팸의 비율은 31% : 69%로 비스팸이 스팸의 2배 정도 되는 비율로 구성되어 있다.

2.3.2. 이미지 데이터

이미지 데이터는 스팸 이미지를 학습하는 것이 아니라, 스팸 이미지의 특징에 대해 분석하여 활용법을 구상하였다. 대부분의 스팸 이미지는 광고 문구가 있다는 특징이 있었기에, 해당 광고 문구에 대한 텍스트를 이미지로부터 추출해서 스팸 이미지 분석이 가능하다고 분석하였다.

또한 이미지에는 텍스트에서 볼 수 없는 상황적 정보가 포함되어 있을 것이라 생각하였다. 상황적 정보를 추출하기 위해 이미지에 대한 캡션을 추출하는 것을 동시에 고안하였다.

이미지 캡션 생성 모델의 학습에는 Flickr 8K Dataset을 사용하였다. 해당 데이터는 Flickr라는 미국의 기업 야후의 온라인 사진 공유 커뮤니티 사이트에서 수집된 데이터로, 5개의 다른 캡션(이미지에 대한 설명)과 쌍을 이루는 8,000개의 이미지로 구성되어있다. 이미지는 6개의 다른 Flickr 그룹에서 선택되었으며, 어떤 유명한 사람이나 위치를 포함하지 않는 경향이 있으며 다양한 장면과 상황을 묘사하기 위해 수동으로 수집되었다.

A man in street racer armor is examining the tire of another racers motor bike.
The two racers drove the white bike down the road.
Two motorists are riding along on their vehicle that is oddly designed and colored.
Two people are in a small race car driving by a green hill.
Two people in racing uniforms in a street car.

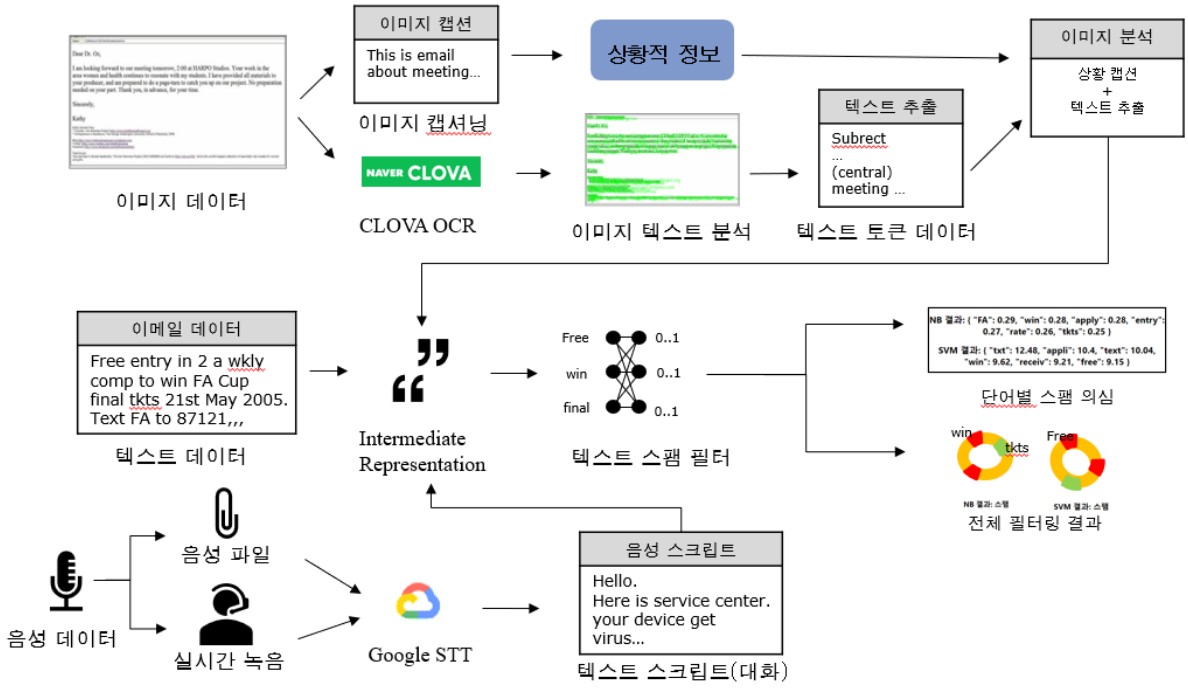
Flickr 8K 제공 이미지 및 캡션 데이터 일부

따라서 이미지를 분석함에 있어서 이미지의 특징인 이미지 캡션과 이미지에 적혀있는 텍스트를 함께 분석하는 방향으로 데이터 특징 분석을 하였다.

2.3.3. 음성 데이터

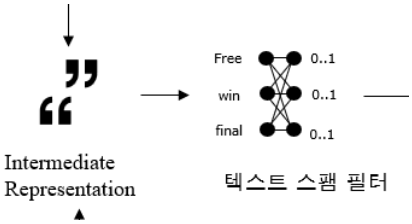
음성 데이터의 경우 이메일이나 문자로 받게 되는 스팸 내용이 음성으로 형식만 변형될 뿐 핵심 내용은 비슷한 점이 많다고 분석하였다. 따라서 음성을 텍스트 형식으로 변환하는 점에 집중하였고, 스팸 여부 분석에 대해서는 텍스트 분석과 동일하게 진행하였다.

3. 연구 내용



전체적인 흐름도는 다음과 같다

3.1. 모델



흐름도 중 위 부분을 담당한다.

3.1.1. SVM 모델

전처리된 데이터를 한 번 더 전처리 진행했다. NLTK라이브러리를 통해 토큰화를 진행하고 의미에 크게 영향을 주지 않는 stopwords나 stemming을 제거, 어간 추출 작업을 수행했다. 데이터를 벡터 형태로 변환하는 작업은 CountVectorizer를 사용하였고 scikit-learn에서 제공하는 SVC를 통해 SVM 모델을 사용해 학습하였다. 이 후 학습한 결과를 피클 형식의 파일로 저장하여 백엔드에 업로드 하여 실행시켰다.

3.1.2. NB 모델

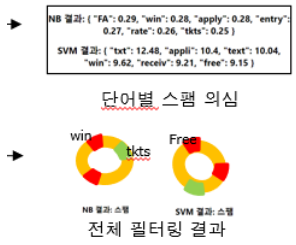
전처리를 진행한 데이터에 CountVectorizer를 사용하여 데이터를 벡터 형태로 변환하여 scikit-learn에서 제공하는 MultinomialNB를 통해 NB 모델을 사용해 학습하였다. MultinomialNB를 사용한 이유는 범주형 변수의 각 피처에 대한 평균을 계산하는 방식으로 GaussianNB보다 스팸 여부 판단에 적합했기 때문이다. 학습한 결과는 피클 형식 파일로 저장하여 백엔드에 업로드 하여 실행시켰다.

3.1.3. 이미지 캡션 생성 모델

이미지 캡션 생성 모델은 VGG16 모델을 사용하여 이미지의 특징을 추출하고, 해당 특징을 바탕으로 이미지의 캡션을 생성하도록 설계했다. 모델의 인코더는 VGG16에서 추출한 이미지 특징을 입력으로 받고, 디코더는 LSTM을 사용하여 이미지 특징을 바탕으로 캡션을 생성한다.

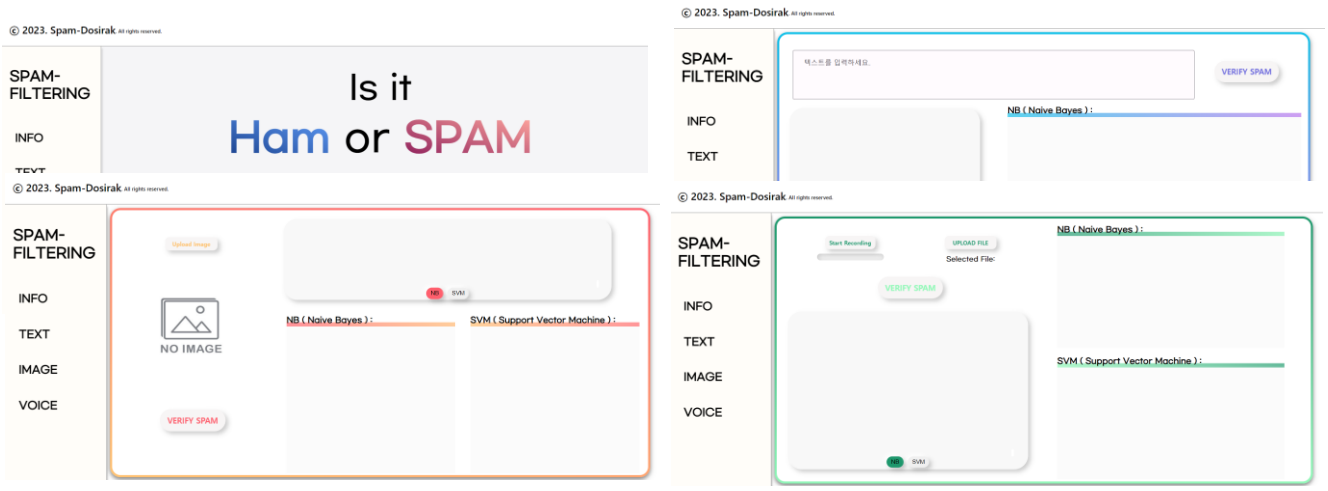
Flickr 8K Dataset을 통해 이미지 캡션을 생성하는 것을 학습했으며, 완성된 모델은 HDF5 형식의 파일로 저장되어 백엔드에 업로드하여 사용하였다.

3.2. React를 이용한 시각화 플랫폼

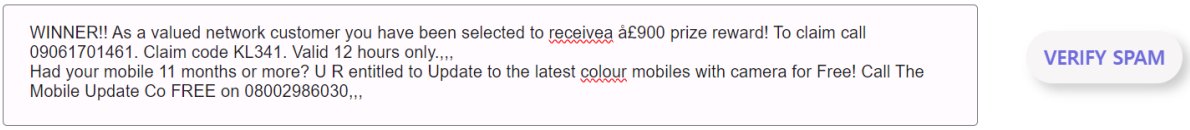


흐름도 중 위에 해당하는 부분으로 결과를 보여주는 부분이다.

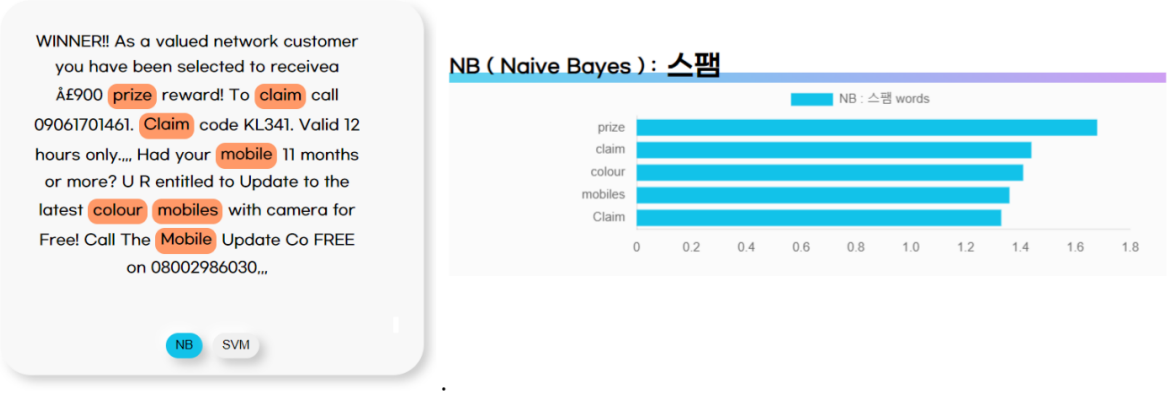
스팸 필터링 결과를 보다 쉽게 파악할 수 있도록 그래프와 확률을 이용한 시각화 플랫폼을 구현하였다. 이 플랫폼은 플랫폼을 개요와 기능을 간략하게 소개하는 메인화면, 텍스트 스팸 테스트 화면, 이미지 스팸 테스트 화면, 그리고 음성 스팸 테스트 화면으로 구성되어 있으며, 사용자에게 편리한 네비게이션 기능을 제공한다.



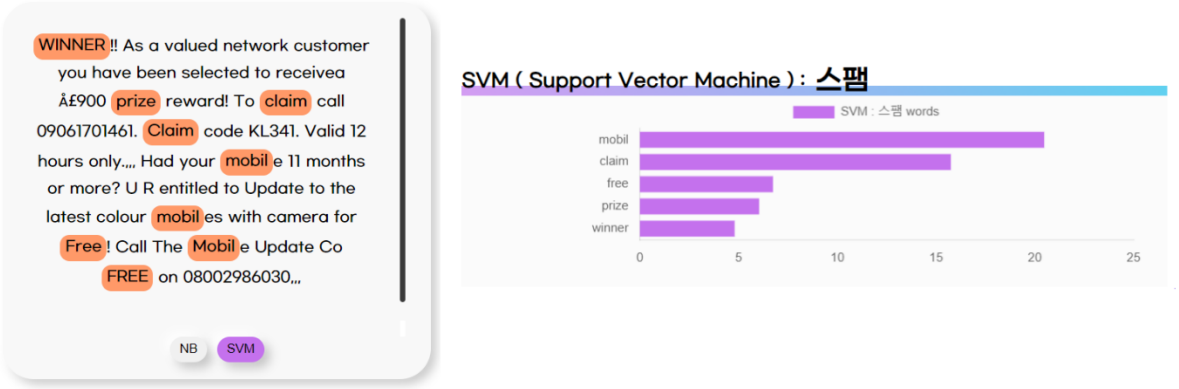
세 가지 기능 중 텍스트 스팸 필터링 동작 과정을 써보았다.



위 사진처럼 사용자는 입력 상자에 테스트하고자 하는 텍스트를 입력한 후 Verify Spam 버튼을 클릭한다. 입력된 메시지는 JSON 형식으로 Flask 서버에 POST 요청이 전송된다. 서버는 요청을 처리하고 응답을 생성한 후, 이 응답은 React의 useState를 통해 스팸 또는 햄 결과 및 필터링된 단어를 저장한다



NB모델의 결과를 하이라이트와 그래프로 확인할 수 있다.

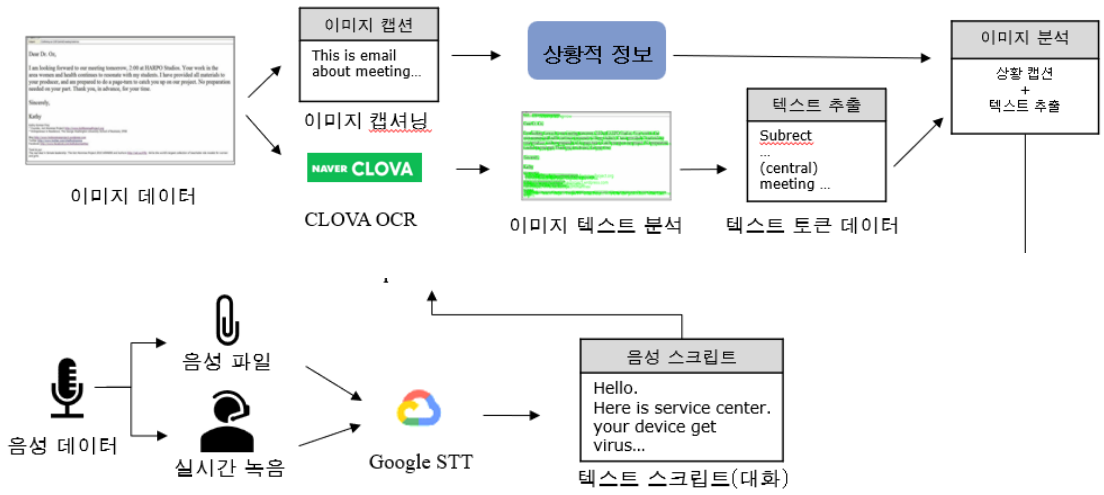


SVM모델의 결과를 하이라이트와 그래프로 확인할 수 있다.

입력된 텍스트가 왜 스팸 또는 햄으로 분류되었는지 이해하기 쉽게 하기 위해 해당 단어들을 하이라이트 처리하여 사용자에게 다시 표시되는 기능을 만들었으며, 이러한 단어와 관련된 확률 정보도 나타내주는 그래프 기능을 생성하였다.

플랫폼의 이미지 스팸 테스트와 음성 스팸 테스트 화면도 비슷한 방식으로 작동하며, 사용자에게 해당 형식의 데이터를 입력 받고 스팸 여부를 시각적으로 보여주는 기능을 제공한다.

3.3. Flask를 이용한 사용자 입력 데이터 및 요청 처리



흐름도 중 위의 일부분(이미지 캡셔닝 제외)을 담당한다.

3.3.1. 음성 데이터 처리

음성 데이터는 Google STT를 이용하여 음성 → 텍스트로 변환한다. 음성 데이터는 사용자로부터 실시간 녹음과 녹음된 음성 파일의 두가지 방법을 이용해서 필터링 모델에 돌릴 수 있도록 했다.

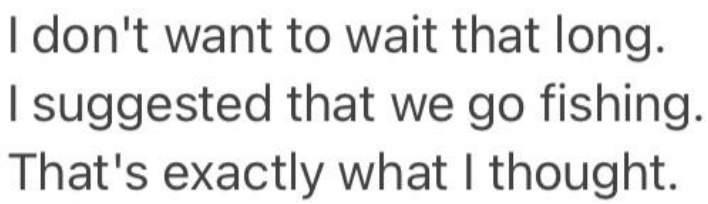


그림 1. Google STT 실시간 녹음 테스트 시 재생한 스크립트

테스트로 위 스크립트를 실시간으로 재생해 녹음 후 결과를 확인 했다.

```
I don't want to wait that long I suggested that we go fishing
 that's exactly what I thought
```

그림 2. Google STT 실시간 녹음 테스트 결과

변환이 잘 된 것을 확인할 수 있다.

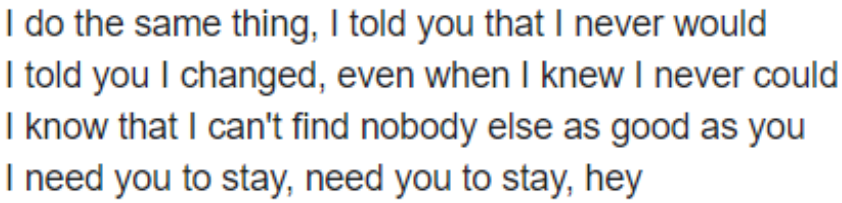


그림 3. Google STT 음성 파일 테스트 스크립트

위 이미지의 스크립트를 녹음한 파일로 음성 파일 변환 테스트도 해보았다.

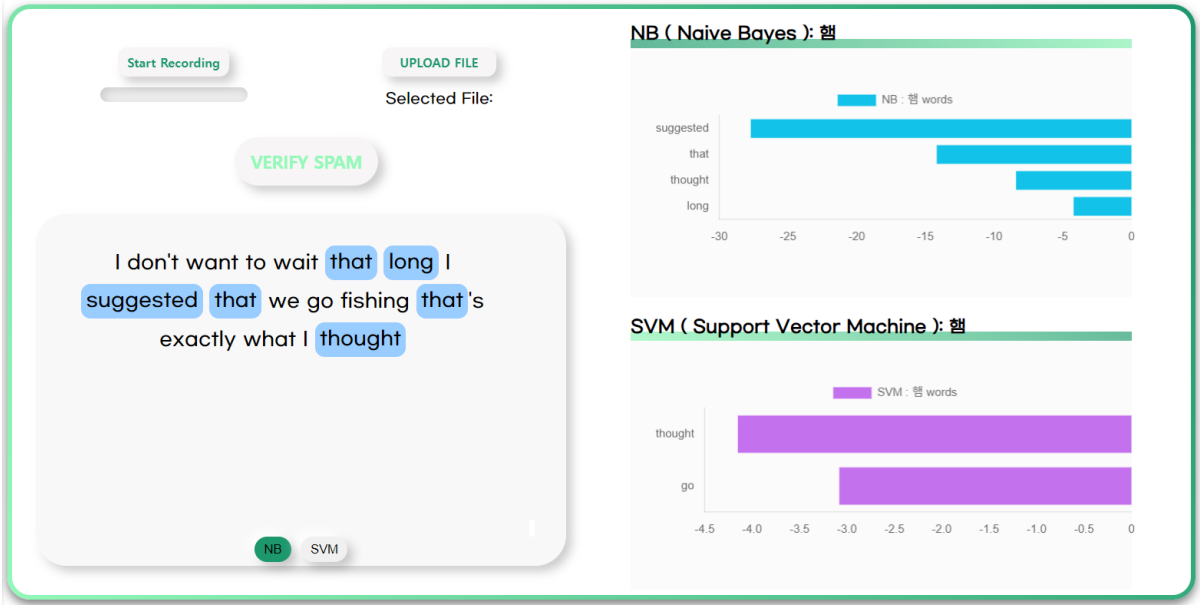
```
I do the same thing I told you that I never would I told you I changed even wh
en I knew I never could I know that I can't find nobody else as good as you I
need you to stay need you to stay hey
```

그림 4. Google STT 음성 파일 테스트 결과

마찬가지로 잘 동작하는 것을 볼 수 있다.

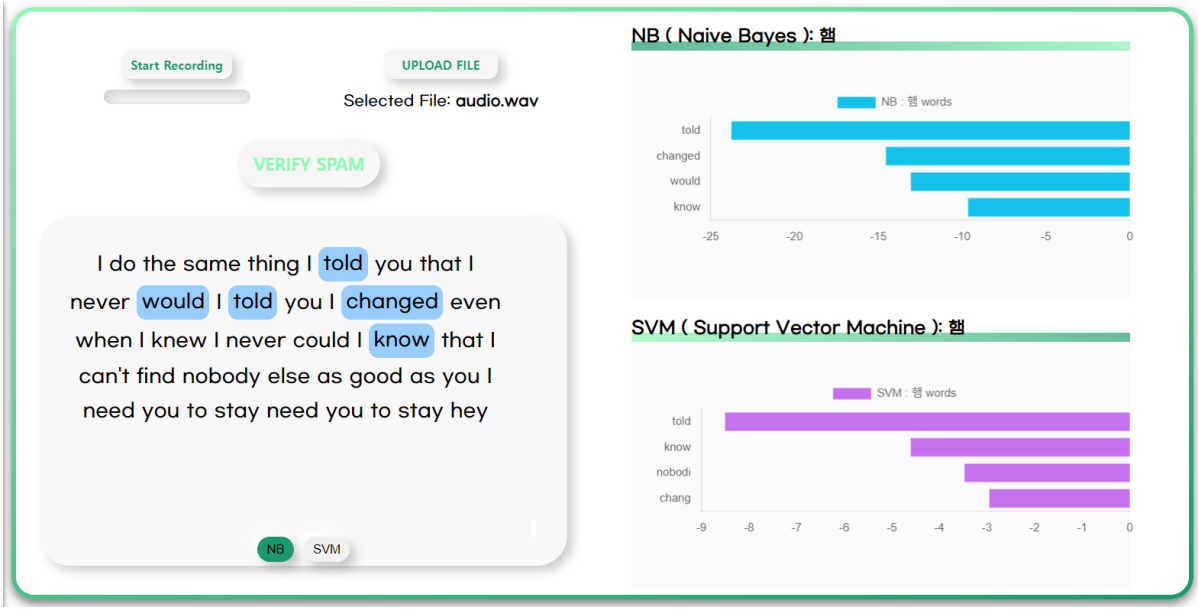
이제 실제로 웹에서 실시간 녹음 후 받은 데이터를 모델에 돌려 스팸 여부를 판별하고,

사용자에게 응답을 반환하는 과정까지의 동작에 문제가 없는 지 확인이 필요하다.



녹음 음성에서 텍스트로의 변환도 잘 되고 모델도 문제 없이 돌아가는 것을 확인했다. 결과 응답 또한 정상적으로 반환된 것을 알 수 있다.

다음으로 실제로 웹에서 음성 파일을 첨부해 모델을 돌렸을 때 동작에 문제가 없는 지 확인해보았다.



마찬가지로 음성파일에서 텍스트로 변환이 잘 됐고 모델도 문제 없이 돌아가는 것을 확인했다.

3.3.2. 이미지 데이터 처리

이미지 데이터는 pytesseract와 Clova OCR을 테스트로 사용해보았으나 Clova OCR이 더 우수한 성능을 보여 Clova OCR을 선택했다. 이를 이용해 이미지에 존재하는 텍스트를 추출한다.

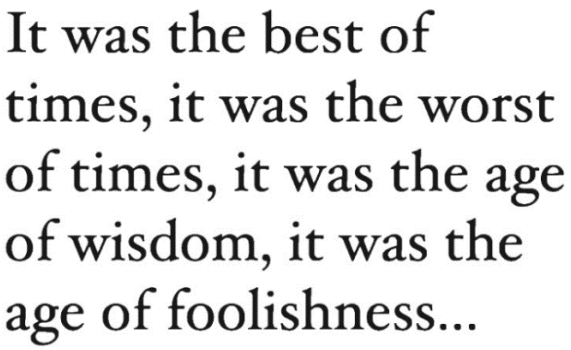


그림 5. Clova OCR 테스트 이미지

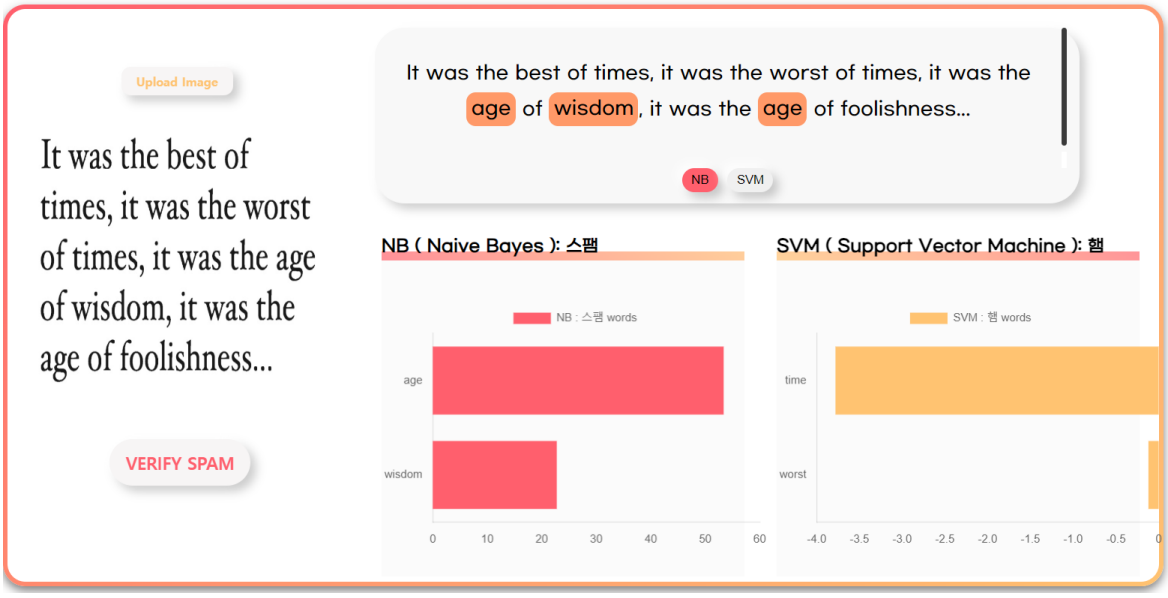
우선 텍스트만 존재하는 이미지로 글자가 잘 추출 되는지 확인한다.

```
It was the best of times, it was the worst of times, it was the age of wisdom,
 it was the age of foolishness...
```

그림 6. Clova OCR 테스트 결과

정확하게 잘 인식된 것을 확인할 수 있다.

실제로 웹에 이미지를 업로드 하여 모델을 돌렸을 때 변환, 스팸 필터링 등 전체 동작에 문제가 없는지, 변환된 텍스트와 스팸 여부 결과 반환도 정상적인지 확인해야 한다.



이미지에 있는 텍스트도 잘 추출되고 모델도 정상적으로 돌아가는 것을 확인했다.

3.3.3. 변환 데이터 전처리

이미지 및 음성 데이터를 텍스트로 변환하여 모델을 적용하기 위해서 텍스트 데이터의 적절한 가공이 매우 중요하다. 아래는 더 정확한 결과를 얻기 위한 텍스트 데이터 가공의 단계이다.

- **텍스트 토큰화 (Tokenization)**: 텍스트 데이터를 단어 또는 문장으로 나누는 토큰화 과정이 필요하다. 이것은 텍스트를 의미 있는 단위로 분해하는 과정이며, 일반적으로 공백이나 구두점을 기준으로 텍스트를 분할한다.
- **불용어 제거 (Stopword Removal)**: 불용어는 의미가 없는 단어나 너무 빈번하게 나타나는 단어를 말한다. 예를 들어 "the," "and," "is"와 같은 단어는 분석에 도움이 되지 않으므로 제거한다.
- **어근 추출 (Stemming or Lemmatization)**: 텍스트의 다양한 형태를 동일한 기본 형태로 변환하는 과정이다. 이것은 변형된 단어를 원형으로 돌리거나 어근을 추출하여 단어의 일반적인 형태로 통일시키는 데 도움을 준다.
- **특수 문자 및 노이즈 제거**: 특수 문자, HTML 태그, URL 및 불필요한 문자열을 제거하여 텍스트 데이터를 정리한다.
- **텍스트 정규화 (Text Normalization)**: 대소문자 통일, 줄임말 확장, 동의어 통합과 같은 텍스트 정규화 작업을 수행하여 데이터를 일관성 있게 만든다.
- **데이터 벡터화 (Data Vectorization)**: 모델을 훈련시키기 위해 텍스트 데이터를 수치 형태로 변환해야 한다. 해당 연구에서는 CountVectorizer를 이용한다.
- **데이터 라벨링 (Data Labeling)**: 각 데이터 항목을 스팸 또는 햄으로 라벨링하여 모델이 올바르게 분류할 수 있도록 한다.

이러한 단계를 따라 데이터를 가공하고 모델을 훈련시키면, 이미지 및 음성 데이터가 변환된 텍스트를 효과적으로 스팸 필터링 모델에 적용할 수 있다. 이렇게 가공된 데이터를 사용하면 더 정확한 스팸 필터링 결과를 얻을 수 있을 것이다.

4. 연구 결과 분석 및 평가

4.1. 연구 결과

4.1.1. 텍스트 스팸

해당 사진들은 텍스트 스팸 필터링 시스템의 작동 원리를 시각적으로 나타낸 것이다. Verify Spam 버튼을 클릭하면, 시스템은 Flask 서버로 Post 요청을 전송하게 된다. 이 Post 요청을 수신한 Flask 서버는 해당 텍스트의 스팸 여부를 확인하기 위한 작업을 진행한다.

시스템은 입력된 텍스트를 사전 처리하여 데이터를 준비한다. 이전에 사전 학습된 Naïve Bayes와 SVM 모델을 활용하여 벡터화 및 분류 모델링을 진행하고 스팸 또는 햄 단어를 추출하게 된다. 해당 과정은 Flask Terminal로 확인할 수 있다.

```
127.0.0.1 - - [06/Oct/2023 23:06:49] "POST /predict HTTP/1.1" 200 -
[('prize', 0.018873343768217317), ('Claim', 0.01774240751119614), ('mobiles', 0.016932323741015352), ('claim', 0.01548318453068172),
('receivea', 0.014969209099956972), ('Update', -0.012635362503266507)]
[('prize', 0.018873343768217317), ('Claim', 0.01774240751119614), ('mobiles', 0.016932323741015352), ('claim', 0.01548318453068172),
('receivea', 0.014969209099956972), ('Update', -0.012635362503266507)]
['spam']
```

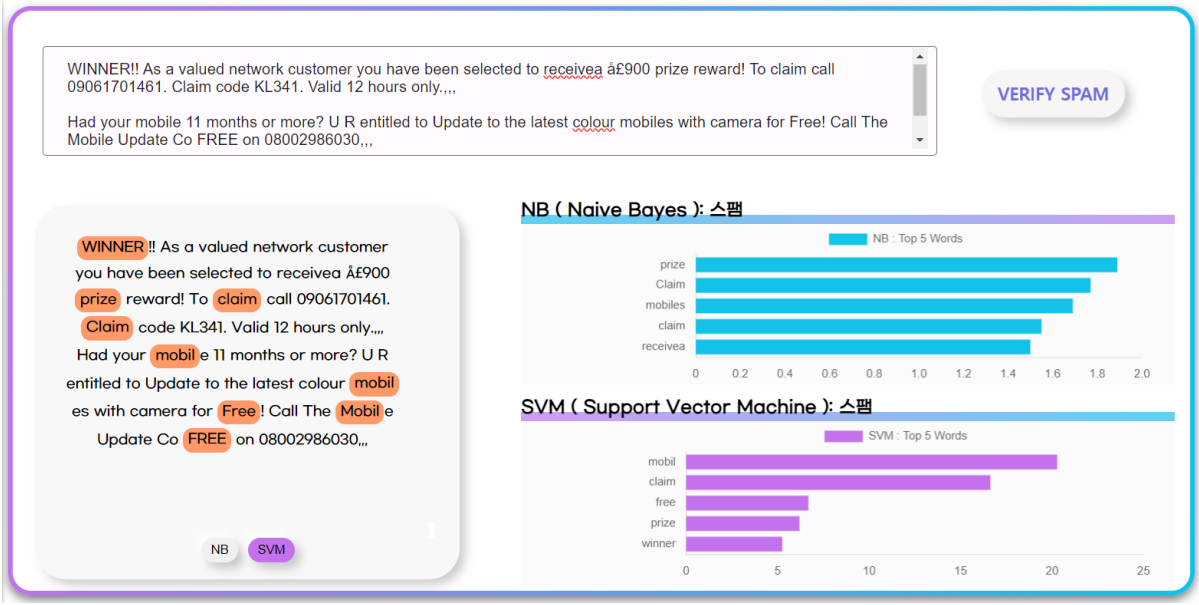
Flask Terminal을 통해 확인 가능한 단어와 확률 1

```
127.0.0.1 - - [06/Oct/2023 23:14:15] "POST /predict HTTP/1.1" 200 -
[('Lol', -0.01083191317906059), ('lunch', -0.010555161981784855), ('smth', -0.010149847299818646), ('Mark', -0.009209317338830533),
 ('Ard', -0.008506008583235686), ('lor', -0.00778728546983615 7)]
[('Lol', -0.01083191317906059), ('lunch', -0.010555161981784855), ('smth', -0.010149847299818646), ('Mark', -0.009209317338830533),
 ('Ard', -0.008506008583235686), ('lor', -0.00778728546983615 7)]
['ham']
```

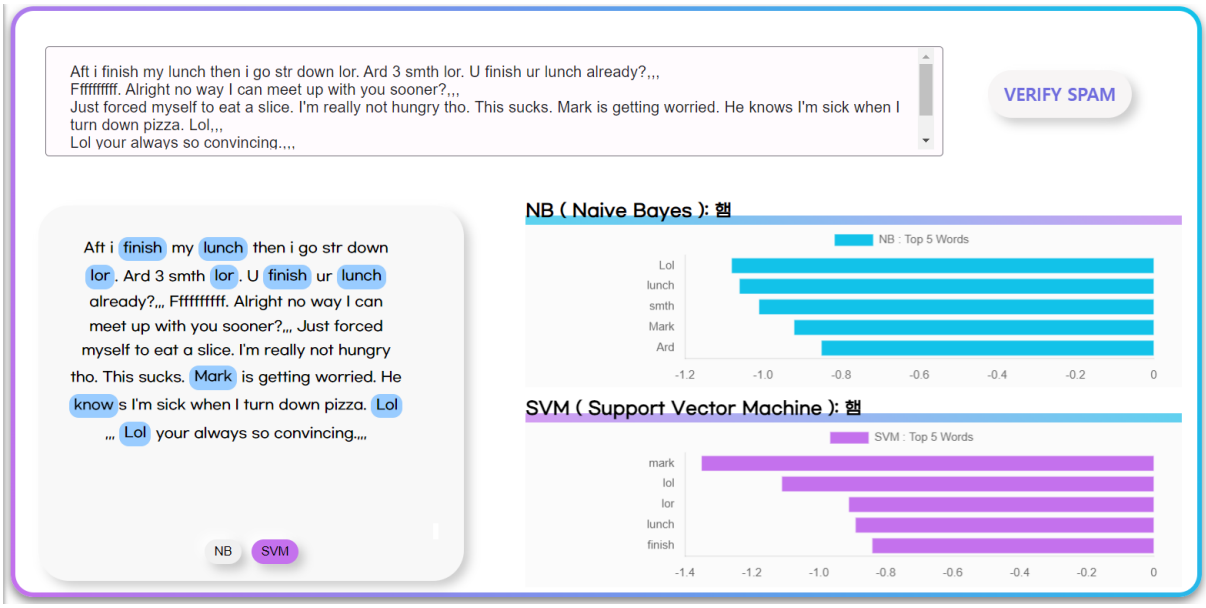
Flask Terminal을 통해 확인 가능한 단어와 확률 2

스팸으로 의심되는 텍스트를 입력하면 왼쪽 하단 박스에 해당 스팸 단어를 강조 표시하고, 마우스를 올렸을 때 해당 단어의 스팸 확률을 확인할 수 있다. 뿐만 아니라, 화면 우측 하단에는 Naïve Bayes 및 SVM 두 가지 기술을 이용한 필터링 결과와 단어의 스팸 확률이 내림차순으로 나열된다.

햄으로 의심되는 텍스트를 입력하였을 경우에도, 확률이 음수로 표시되는 것 외에는 스팸과 동일한 방식으로 정보가 제시된다.



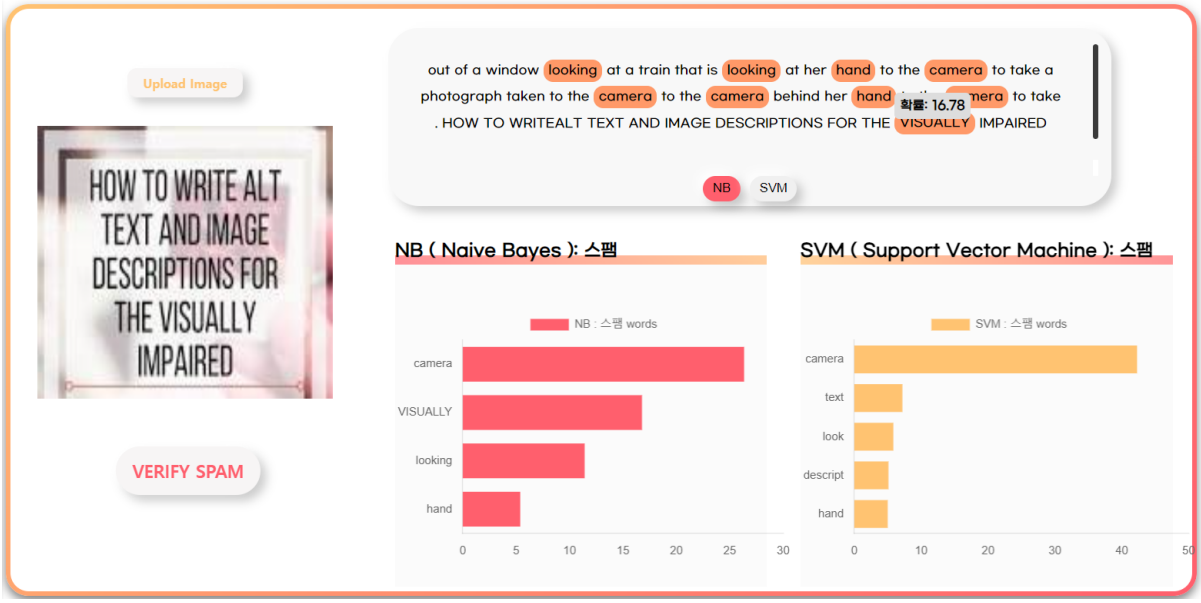
스팸으로 분류된 경우 화면



햄으로 분류된 경우 화면

4.1.2. 이미지 스팸

이미지의 스팸여부를 확인하기 위해 Image Captioning(이미지 캡셔닝) 기술과 Naver Clova OCR 기술을 사용하였다.



a. 이미지 캡셔닝

이미지 캡셔닝은 텍스트가 이미지 내에 존재하지 않을 때에도 스팸 여부를 판별하기 위해 사용하였다. 이를 위해 사전에 학습된 모델을 활용하여 업로드한 이미지와 해당 이미지의 캡션을 전처리하고 이미지에서의 문맥을 고려하고 그에 맞는 특징을 추출한다. 추출된 캡션은 텍스트 스팸을 판별하는 과정과 동일한 방식으로 스팸 여부를 확인하기 시작한다.

b. 이미지 내 텍스트 인식

Clova OCR API는 이미지 내 텍스트를 추출하기 위해 사용하였다. 이 API는 모든 유형의 이미지에서 자동으로 텍스트 영역을 감지하고 사전 학습된 모델을 기반으로 문서 패턴과 유형을 자동으로 분류하는 서비스를 제공한다. 사용자는 해당 API에 요청을 보내고 응답을 받은 후, 추출된 텍스트를 텍스트 스팸 분류와 동일한 방식으로 스팸 여부를 확인하기 시작한다.

4.1.3. 보이스 스팸

음성 스팸을 감별하기 위해서 두 가지 방법을 사용한다. 첫 번째 방법은 실시간 음성 녹음을 통한 테스트이다. 사용자의 음성이 실시간으로 녹음되며, 이 녹음된 음성을 스팸 여부를 확인하기 위해 텍스트로 변환한다. 변환된 텍스트는 벡터화 및 분류 모델을 사용하여 스팸 여부를 판단하고 사용자에게 표시한다.

두 번째 방법은 미리 준비된 음성 파일을 검사하는 것이다. 미리 녹음된 음성 파일을 시스템에 제공하고, 이 파일을 텍스트로 변환해 동일한 방식으로 스팸 여부를 판단한다.

음성을 텍스트로 변환하는 과정에서 보다 정확한 스크립트를 얻기 위해 Google STT(Speech-to-Text) API를 활용한다. 이 API는 음성을 텍스트로 정확하게 변환하며, 변환된 텍스트를 스팸 여부를 판별하기 위한 모델에 입력으로 사용된다. 이를 통해 음성 스팸 여부를 식별하고 사용자에게 정확한 결과를 제시할 수 있다.

4.2. 한계점

4.2.1. 모델 개선의 한계

보다 정확한 결과를 위해 SVM과 NB 모델을 기반으로 학습 알고리즘 자체를 재설계하는 방법과 더 많은 데이터를 모아 학습량을 대폭 늘리는 방법을 고려하였다. 하지만 학습 알고리즘을 설계하는 과정에서 시간과 정보의 부족으로 어려움을 겪었고 학습 데이터의 양을 늘리는 방안을 택하였다.

4.2.2. 이미지 캡셔닝의 성능 한계

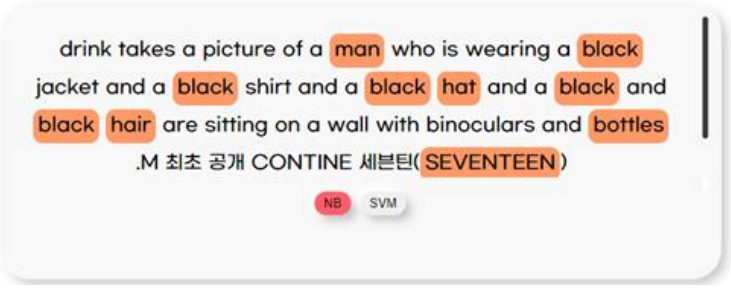
이미지 캡셔닝 기술을 처음 도입했을 때 성능이 좋지 못했다. 이미지의 문맥을 전혀 파악하지 못해 상관 없는 캡셔닝이 추출되는 문제점을 겪었다.

남자 연예인 사진 1

```
--------------------Predicted--------------------
startseq green dog with brown markings in a distorted in a field of clover and purple flowers in the background in a distorted in the grass and a dis
torted in the background it is playing in the grass
['startseq', 'green', 'dog', 'with', 'brown', 'markings', 'in', 'a', 'distorted', 'in', 'a', 'field', 'of', 'clover', 'and', 'purple', 'flowers', 'in
', 'the', 'background', 'in', 'a', 'distorted', 'in', 'the', 'grass', 'and', 'a', 'distorted', 'in', 'the', 'background', 'it', 'is', 'playing', 'in'
, 'the', 'grass']
green dog with brown markings in a distorted in a field of clover and purple flowers in the background in a distorted in the grass and a distorted in
```

남자 연예인 사진을 테스트 했을 때 dog, flower과 같이 상관 없는 캡셔닝이 추출된 것을 볼 수 있다.

이런 한계점을 보완하고자 Library의 데이터 양을 Flickr 8K(8000장)에서 Flickr 30K(30000장) 데이터로 변경해 학습 데이터 양을 늘려 정확도를 높여보려했지만 오히려 정확도가 더 떨어지는 결과를 얻었다. Flickr 30K 데이터로 학습하면 생성된 결과에 'rock'이라는 단어가 매우 많이 등장했고 이를 통해 오버피팅이 있을 수 있다는 점을 인지했다. 그래서 기존 Flickr 8K 데이터를 사용하되 CPU 학습에서 GPU학습으로 변환시켜 진행하였고, 처음보다 나은 이미지 캡션 결과를 얻었다.



위 남자 연예인 사진1을 테스트 한 결과 male로 인식하는데 성공하였다.

그러나 이미지 캡셔닝은 학습 양이 8000개의 이미지라는 한계와, 이미지에서 핵심 이미지 위주로 인식하기에 정확한 문맥을 파악해내는 부분에서의 한계가 있다고 생각되었고 해당 부분을 개선하는 것은 특정 분야에 편향된 이미지가 없는 데이터 학습 방법과 함께 추후 연구로 진행할 예정이다.

5. 결론 및 향후 연구 방향

5.1. 결론

이번 연구를 통해 SMS 및 메일 속의 텍스트 스팸 뿐만 아니라 메일에 포함된 이미지, 전화를 통한 음성 스팸 등을 필터링 할 수 있는 멀티모달 기반 스팸 필터링 플랫폼을 개발하고자 하였다. 이 과정에서 Python을 스팸 필터링을 위한 머신 러닝 기술과 학습모델의 성능을 최대치로 이용하기 위한 데이터 전처리 과정도 경험해 볼 수 있었다.

또한 Google STT, Clova OCR, 이미지 캡셔닝 등 텍스트가 아닌 형식의 데이터 처리 기술을 음성, 이미지 데이터 처리에 활용함으로써 목표했던 "멀티모달"의 의미를 실현해낼 수 있었다.

하지만 텍스트에 비해 이미지 및 음성 데이터는 자료가 많지 않고, 스팸 이미지 또는 보이스피싱 등 음성 스팸은 기존 텍스트 기반의 스팸과는 내용이 다른 경우가 많아 텍스트 스팸을 학습시킨 모델로는 정확도를 높이기가 어려웠다. 더 성공적인 필터링을 위해 데이터 타입 별 고유한 특징을 살린 데이터가 더 필요하다고 판단했다. 데이터 수집의 어려움과 모델 개선을 위한 시간의 제한성 등의 이유로 아직 더 개선이 필요하지만, 연구 목표에 부합하는 연구 결과를 이루어 냈다고 생각한다.

5.2. 향후 연구 방향

이번 연구에서는 정확도와 속도에 제약이 있지만, 미래에 더 많은 이미지 및 음성 데이터를 학습시키고 모델을 개선한다면 이미지 및 음성 데이터에 특화된 스팸 필터링 모델을 개발할 수 있을 것이다.

(1) 이미지 캡셔닝 모델 성능 향상

이미지 캡셔닝 및 문맥 이해 능력을 향상시킨다면 이미지 스팸 처리에 있어서는 완전한 성능을 보일 것으로 예상된다.

(2) 사용자 데이터 이용

사용자 동의를 얻은 후 해당 사용자의 스팸 필터링 기록 및 이전 결과를 저장하고, 사용자가 이를 다시 확인할 수 있는 기회를 제공할 수 있다. 추가적으로 사용자의 데이터를 활용하여 모델을 개선하고 정확도를 향상시킬 수 있을 것이다.

(3) 언어 확장

현재는 영어로 한정되어 있지만, 추후에는 한국어 및 다른 언어 데이터를 수집하고 분석한 후, 모델에 적용하여 다국어 지원을 확대할 수 있을 것이다.

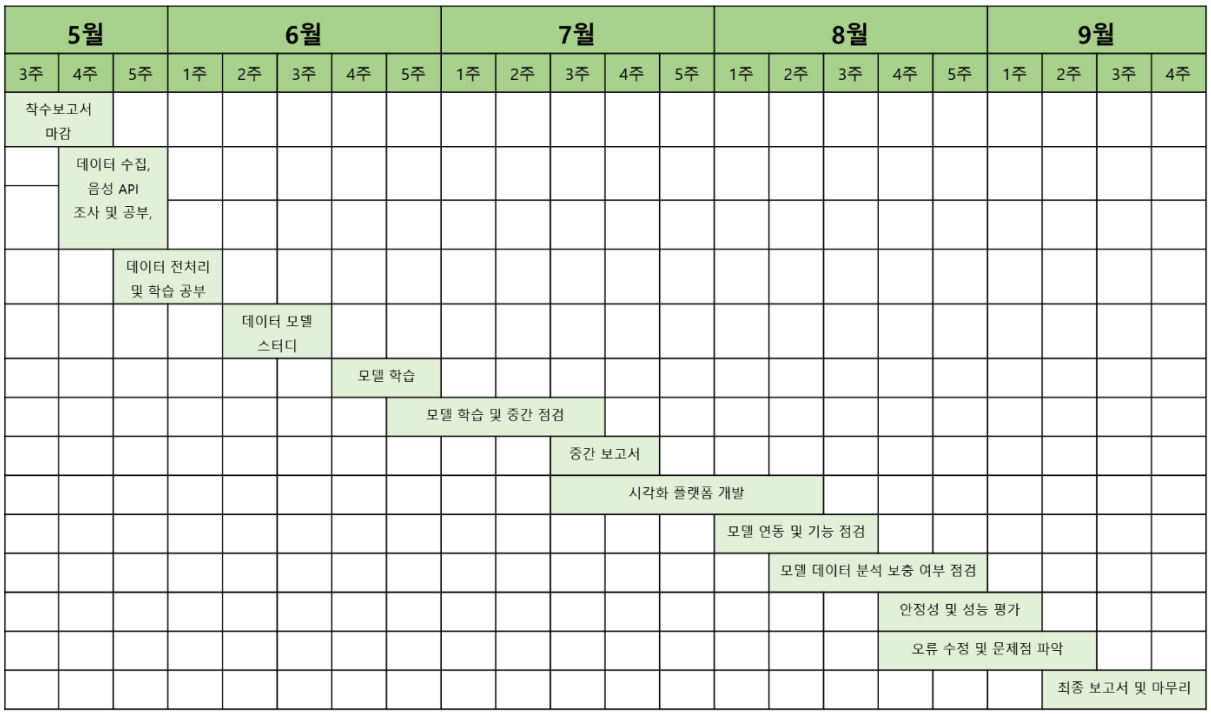
(4) 사용자 경험 조사

사용자 경험을 개선하기 위해 실제 사용자의 피드백을 수집하고 이를 반영하여 서비스를 지속적으로 향상시키는 것이 필요하다.

성능적인 측면과 사용자 경험 측면을 함께 고려하며 서비스를 고도화 한다면 향후 다른 서비스와 차별화를 이루고 더 나은 서비스로 발전시킬 수 있을 것으로 기대한다.

6. 개발 일정 및 역할 분담

6.1. 개발 일정



6.2. 구성원 별 역할

이름	역할
박혜경	데이터 수집 SVM 모델 이미지 캡션 생성 모델
이승현	NB 모델 React 플랫폼 개발 Flask & React 연동 및 Refactoring
천영채	Google STT를 이용한 음성데이터 변환 Clova OCR을 이용한 이미지 데이터 변환 Flask 서버 개발
공동	- 데이터 전처리 및 학습 공부 - Google STT & Clova OCR 등 데이터 변환을 위한 기술 조사 및 공부 - 발표 및 시연 준비

7. 참고 문헌

[1] V. Metsis, I. Androutsopoulos and G. Paliouras, "Spam Filtering with Naive Bayes - Which Naive Bayes?". Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA, 2006.

[2] 민도식, 송무희, 손기준, 이상조, "SVM 분류 알고리즘을 이용한 스팸메일 필터링 (SPam-mail Filtering Using SVM Classifier)", 한국정보과학회 03 봄 학술발표논문집(B) 2003 Apr.,2003년,pp.552-554

[3] Enron-Spam Dataset, Avaliable : http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam

[4] SMSSpam Collection, Available : https://archive.ics.uci.edu/dataset/228/sms+spam+collection

[5] SpamAssassin Dataset, Available: https://spamassassin.apache.org/old/publiccorpus/

[6] Flickr Image Dataset : Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting Image Annotations Using Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

[7] MathWorks SVM, Available : https://kr.mathworks.com/discovery/support-vector-machine.html

[8] Wikipedia NB Classifier[Online], Available : https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[9] Borneel Bikash Phukan, Amiya Ranjan Panda, "An Efficient Technique for Image Captioning using Deep Neural Network". Under review by an internationally recognized Scopus indexed journal 2020

[10] Deep Learning Bible 2. Classification - 한글[도서] : Understanding Vgg-16 Vgg-19, Available : https://wikidocs.net/164796

[11] NAVER CLOVA-OCR, Available : https://guide.ncloud-docs.com/docs/ko/clovaocr-overview