**종합설계 중간보고서**

**악성댓글 필터링 브라우저 확장 프로그램**

**(국문)**

**(영문)**

**2024 년 4 월 21 일**

과제 수행팀 : 프로젝트 마치

12183588, 허대현

12181681, 정상화

12181704, 한용인

**목차**

1. 과제 개요
   1. 개발 목표
   2. 개발 시스템
2. 과제 진행 내용
   1. 진행사항 요약
   2. 설계 결과
   3. 성능 평가 방법
3. 수행 연구 및 역할 분담
   1. 구성원 역할
4. 추진일정
5. 과제 수행 요청사항 및 기타사항

**1. 과제 개요**

(1) 개발 과제 목표

불쾌한 표현을 웹 서비스 이용자가 사이트를 열람하기 전에 사전에 차단하여, 커뮤니티 활동의 본질에 집중할 수 있도록 돕는 것이 본 과제의 목적이다.

과제의 최종 목표는 사용자가 요청한 커뮤니티 게시글 HTML 문서를 입력으로 받아, 출력으로 부정적인 언어표현이 사용된 댓글 태그를 필터링한 HTML 문서를 제공하는 웹 브라우저 확장 프로그램을 제작하는 것이다.

(2) 개발 시스템 specification

- 주요 기능, 동작 환경 및 개발환경 등 포함할 것

주요기능

* 웹사이트 댓글 태그 식별 및 댓글 내용 수집 기능
* 대상 웹사이트는 네이버 뉴스, 유튜브, 디시인사이드
  + 사이트마다 개별 parsing 로직을 적용하여 댓글 데이터를 가져온다.
* 댓글 데이터 부정적 언어표현 여부 판정 기능
  + 댓글은 총 10개의 태그로 멀티라벨링될 수 있음. (‘여성/가족’, ‘남성’, ‘성소수자’, ‘인종/국적’, ‘연령’, ‘지역’, ‘종교’, ‘기타 혐오’, ‘악플/욕설’, ‘개인지칭’)
  + 모델은 댓글이 분류된 라벨과 확률을 함께 제공한다. (eg. [‘여성/가족’:0.00132, ‘남성’:0.62131, …])
* 혐오 표현으로 판정된 댓글 태그의 내용을, 차단 텍스트로 전치하는 기능
* 차단 기능의 적용 여부, 적용 범위 등을 제어할 컨트롤 패널 UI
  + 10개의 태그 중 필터링을 원하는 태그를 선택할 수 있음.
  + 댓글이 특정 타입의 악성댓글로 분류될 확률을 기준으로 차단 강도를 설정할 수 있음.

동작환경

* 클라이언트는 크롬 확장 프로그램을 통해 웹 페이지의 댓글을 필터링 가능
* 댓글 필터링을 담당하는 서버와 모델은 클라우드 환경에서 실행된다. 클라우드 서비스는 아마존 AWS EC2를 이용 (서버 스펙 협의 후 기재 필요)

개발환경

* IDE의 경우, Visual Studio, IntelliJ를 이용하고 모델의 학습을 위해 Colab, 구름 IDE를 이용
* 서버는 스프링부트 3.x 버전을 이용해 개발, JDK 17 이상을 사용
* 모델은 (파이썬 버전, 넘파이 등 필요한 라이브러리 기재 필요) 사용
* 프론트 쪽도 기재 필요

**2. 과제 진행 내용**

(1) 진행 상황 요약

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 추진항목 | 백분율(%) | 추 진 일 정( . . ) | | | | | | | | | 진척도  (%) | 세부결과 |
| 3 | | | | 4 | | | | |
| 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| 브레인 스토밍 |  |  |  |  |  |  |  |  |  |  | 100 |  |
| 요구사항 정리 |  |  |  |  |  |  |  |  |  |  | 100 |  |
| UML 설계 |  |  |  |  |  |  |  |  |  |  | 100 |  |
| 프론트 개발 |  |  |  |  |  |  |  |  |  |  | 50 |  |
| 서버 개발, 환경 구축 |  |  |  |  |  |  |  |  |  |  | 50 |  |
| 모델 학습 및 검증 |  |  |  |  |  |  |  |  |  |  | 30 |  |
| 패킷 데이터 명세 |  |  |  |  |  |  |  |  |  |  | 100 |  |
|  | 100 | 과제진도율(%) | | | | | | | | | 53 |  |

(2) 설계 결과 (기능적 요구 명세를 위주, 6주차 강의 내용 참고할 것)

- 시스템 구조 (상위 설계)

- 주요 컴포넌트 설명

- 상세 설계결과 (컴포넌트 및 인터페이스, 시스템 block diagram, DB 설계등)

- 알고리즘(계획) 및 구현 방법 (계획)

시스템 구조

텍스트, 스크린샷, 폰트, 로고이(가) 표시된 사진

자동 생성된 설명

클라이언트 컴포넌트 설계

🡨-이미지🡪

클라이언트 데이터 플로우

텍스트, 도표, 평면도, 폰트이(가) 표시된 사진

자동 생성된 설명

1) 댓글 추출

1. 커뮤니티 별로 댓글 데이터에 대한 HTML 태그 및, ID, Class가 상이한 점을 대응을 위해 사이트별 Parser를 별도로 내장. 사이트에 따른 다른 Parsing 기법을 적용
2. 요청이 들어온 사이트가 지원하는 사이트임을 확인하기 위한 도메인 테이블 구현

2) 추출 댓글 패킷화 및 WAS에 전달

1. 추출한 댓글들을 WAS에서 수신할 수 있는 JSON 명세에 맞는 패킷으로 변환하여 WAS에 전송.
2. 이후 클라이언트는 WAS로부터 예측 결과를 담은 Packet을 수신하기를 대기함.

3) 차단 대상 댓글 내용 수정

1. 요청받은 HTML 파일과 WAS로부터 수신한 예측 결과의 Array Index를 비교하여, 매칭되는 댓글 Tag의 내용을 차단 텍스트로 전치하고 수정된 HTML를 만들어 낸다.

4) 컨트롤 UI

1. 차단 정도 및 범주 설정을 시각적으로 진행하기 위한 익스텐션 설정 페이지(별도의 HTML로 구현)
2. 저장된 설정 값은 Local로 저장. 설정 데이터는 WAS에 Packet 전달 시에 패킷에 첨부하여 전송.

서버 컴포넌트 설계

영수증, 텍스트, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

* CommentController: 클라이언트 요청을 직접적으로 수신하고 응답을 반환
* CommentService: 요청바디에서 개별 댓글을 추출하고 repository혹은 categrizer로부터의 댓글 분류 결과를 반환
* CommentRepository: 저장된 댓글 판독결과에 접근하는 기능을 제공하는 인터페이스
  + ReactiveCommentTagRedisRepository: 캐시저장소에 저장된 댓글 정보를 읽기/쓰기
  + PerformanceTestCommentRepository: 테스팅을 위한 모킹 객체
* CommentCategorizer: 댓글을 판독하는 기능을 제공하는 인터페이스
  + NLPModelClient: NLPModel과 통신하여 댓글 판독 정보를 획득
  + MockCommentCategorizer: 테스팅을 위한 모킹 객체

서버 데이터 플로우

도표, 평면도, 라인, 기술 도면이(가) 표시된 사진

자동 생성된 설명

1) 빈도 높은 텍스트 캐싱

가) 빈도 높게 등장하는 댓글의 경우 그 SHA-256 해시값과 대응되는 판정 결과를 Pair로 별도의 Table에 저장해 Caching.

나) SHA-256 해시값이 Table에 존재할 때는 미리 저장된 Prediction 값을 사용하고, 그렇지 않을 경우 Model 서버에 해당하는 Text에 대해 Prediction 요청.

2) 댓글 패킷 파싱 및 모델 서버 입력형식에 맞게 변환

가) 마찬가지로 모델 서버와 통신하기 위해, 사전에 정의된 명세에 맞도록 패킷을 구문분석함.

3) 매핑이 완료된 Response 패킷을 클라이언트에 전달

모델 컴포넌트 설계

🡨-- 이미지 -🡪

모델 데이터 플로우

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

1) WAS로부터 판정해야 할 Text를 수신

2) 사전 학습된 모델로부터 Text의 Prediction을 제공, (Prediction은 어떤 종류의 혐오 표현인지를 보이는 Label Pair로 출력)

3) WAS 서버로 예측 결과를 전달

4) 모델 학습 시 데이터 셋은 K-MHaS Dataset 및 Unsmile Dataset 을 활용. (CC-BY-SA-4.0 라이선스로 이용 가능함) 지도 학습 방법을 통해 모델 학습.

(3) 성능 평가방법(계획)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 평가항목 | 평가방법 | 평가기준 | 목표치 | 중요도(순위) |
| 1.모델의 정확도 | 테스트 케이스를 통해 테스트 진행 | F1-Score | 0.7 |  |
| 2.API의 응답속도 | API 테스팅 툴로 테스팅 진행 | 평균 응답속도 | 3s |  |
| 3. |  |  |  |  |
| 4. |  |  |  |  |

주1) 평가항목은 응답시간, 속도, 처리능력 등 성능평가의 기준이 되는 사항으로 타당성 있는 평가기준과 목표치를 수치로 제시하여야 함.

(단, 과제팀별 과제성격에 따라 자유기재 가능, 2가지 이상 권장)

주2) 비중은 최종목표에 대한 각 평가항목의 중요도를 백분율로 표시함.

**3. 수행 연구 및 역할 분담**

**3.1 구성원 역할**

|  |  |  |
| --- | --- | --- |
| 학 번 | 성 명 | 역 할 |
| 12183588 | 허대현 | 데이터셋을 활용한 모델설계 및 학습 및 성능 기준 확인  Flask or FastAPI 기반 모델 서버용 Back-end |
| 12181681 | 정상화 | WAS 서버 Back-end 설계/ 제작 및 Data Packet 명세 |
| 12181704 | 한용인 | Client 설계/ 제작 및 Data Packet 명세 |
|  |  |  |
|  |  |  |

주) 팀 구성원과의 세부적인 역할분담내용을 기술

**3.2 (최초 계획서 대비 수정사항이 있으면) 수정사항 정리**

**제안서에 기술된 내용에서 추가, 삭제, 보완하는 등의 변경사항에 대하여 상세하게 기술하고, 그 이유 또한 상세하게 기술할 것.**

**4. 추진일정 (예)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 업무내용 | 담당자 | 3 | | | | 4 | | | | | 5 | | | | 6 | | | | |  | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |  |  |  |  |  |
| 자료조사  Requirement  Analysis | 공통 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| System Design  (Block diagram, System specification) | MMM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 기능별 Coding | MMM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Test | MMM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 중간보고서 | 공통 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing 및 보완 | MMM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 최종제작 | 공통 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 최종보고서 | 공통 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**5. 과제 수행 요청사항 및 기타사항**