**LSTM 기반의 다국어 대응 웹페이지 Boilerplate Removal Model**

|  |  |
| --- | --- |
| **프로젝트 기간** | **2023.03.02 ~ 2023.10.30 (총 8개월)** |
| **프로젝트 팀원** | **이재욱 (컴퓨터소프트웨어학부, 4학년) 손승일 (컴퓨터소프트웨어학부, 4학년)** |
| **지도교수** | **차재혁 교수님** |
| **프로젝트 명** | **LSTM 기반의 다국어 대응 웹페이지 Boilerplate Removal Model** |
| **프로젝트 내용** | The modern web page includes areas with independent functionalities such as banners, menus, and ads, which are separate from the main content. The process of removing these elements to extract the main content of a web page is referred to as boilerplate removal. Due to the development of most boilerplate removal algorithms and models focusing on English web pages, there is a tendency for them to not perform well on non-English languages, especially East Asian languages such as Korean. In this project, the goal is to enhance the BoilerNet model to improve its boilerplate removal capabilities for East Asian languages, including Korean. For our specific scope and approach to this project, we primarily focused on boilerplate removal for Korean webpages. |
| **기대효과 및 개선방향** | We developed an Android app that takes a URL from a webpage, which in turn reads out the main content of that website. Through more development and implementation, we expect to see versatile ways to apply this functionality to help people with disabilities that prohibit them from smartphone usage. |

|  |  |
| --- | --- |
| **SW프로젝트 결과보고서** | |
| **프로젝트명** | **LSTM 기반의 다국어 대응 웹페이지 Boilerplate Removal Model** | |
| **프로젝트**  **요약** | Modern web pages contain distinct areas like banners, menus, and ads with independent functions, separate from the main content. Boilerplate removal is the process of eliminating these elements to extract the essential content of a webpage. However, many removal algorithms and models are primarily developed for English web pages, leading to suboptimal performance on non-English languages, particularly East Asian languages like Korean. This project aims to enhance the BoilerNet model, focusing on improving its boilerplate removal capabilities for East Asian languages, including Korean. The goal is to address the limitations of existing models and algorithms when applied to diverse linguistic contexts, ensuring more accurate and effective extraction of main content from web pages in languages beyond English. | |
| **프로젝트**  **기간** | **2023.03.02 ~ 2023.10.30 (총 8개월)** | |
| **산출물** | 졸업 작품 ( o ), 졸업 논문 ( ) | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **학과** | **학번** | **학년** | **이름** | **연락처** |
| 컴퓨터소프트웨어 | 2017030128 | 4 | 이재욱 | [pntitgld@gmail.com](mailto:pntitgld@gmail.com)  010-5113-9376 |
| 컴퓨터소프트웨어 | 2018008504 | 4 | 손승일 | [ajeseung@naver.com](mailto:ajeseung@naver.com)  010-5934-1604 |

**목 차**

1. 프로젝트 개요

1.1 프로젝트 목적 및 배경

1.2 프로젝트 최종 목표

2. 프로젝트 내용

3. 프로젝트의 기술적 내용

4. 프로젝트의 역할 분담

4.1 개별 임무 분담

4.2 개발 일정

5. 결론 및 기대효과

**1. 프로젝트 개요**  
 Extracting the main content from web pages is a crucial task with applications ranging from enhancing usability, such as reader views for news articles, to supporting information retrieval and natural language processing. Current methods often rely on extensive hand-crafted features for classification, making them tailored to specific web page distributions, potentially lacking generalization ability. Our proposed solution introduces a neural sequence labeling model that dispenses with hand-crafted features, using only HTML tags and words as input. This approach enables the development of a browser extension that directly highlights the content of diverse web pages within the browser. Additionally, we present such boilerplate removal that supports the Korean language.

**1.1 프로젝트 목적 및 동기**

Web pages serve as valuable information repositories but are often entangled with advertisements, banners, and other boilerplate content, complicating the extraction of primary information. Extracting primary content from web pages is a fundamental task with significant implications for retrieval models and other content extraction endeavors.

Our project focused on BoilerNet, utilizing an LSTM NLP model to eliminate unnecessary boilerplate from modern web pages. However, BoilerNet lacked support for boilerplate removal from Korean web pages. The primary objective of our project was to develop a text-to-speech application specifically designed for Korean boilerplate removal.

This application was envisioned as a valuable tool, particularly beneficial for individuals with disabilities. The goal was to enhance accessibility and facilitate a more streamlined web browsing experience for users facing challenges related to extraneous content on web pages.

**1.2 프로젝트 목표**

The central aim of our project was to conceive and construct a specialized text-to-speech application tailored explicitly for the removal of boilerplate content from Korean web pages. This application was strategically envisioned as a robust tool, with a primary focus on providing significant benefits to individuals with disabilities. Our overarching objective was to contribute to the improvement of accessibility by creating a solution that facilitates a more efficient and user-friendly web browsing experience. By addressing the challenges associated with extraneous content on web pages, our project sought to enhance the overall usability and inclusivity of online platforms, with a particular emphasis on supporting users with diverse needs and abilities. The development process involved careful consideration of the unique linguistic and structural characteristics of Korean web pages, ensuring the effectiveness of our text-to-speech application in addressing the specific challenges posed by boilerplate content in this context.

The project commenced in March, encountering initial challenges in the ideation phase. The idea discussion extended until May, ultimately converging on the development of a text-to-speech application. Subsequent to this decision, a substantial period was dedicated to studying and delving into BoilerNet's code and model. This phase involved foundational work, including the exploration of basic concepts in natural language processing (NLP). By August, the project reached a milestone with the establishment of basic functionalities, allowing the model to run on Colab. However, a significant hurdle remained in creating a Korean dataset. Drawing inspiration from the Japanese BoilerNet project, which translated GoogleTrends2017 raw HTMLs from English to Japanese to construct their dataset, we adopted a similar approach. By September, the model successfully underwent training with our dataset, paving the way for its integration into our application. A month later, we initiated the alpha testing phase for our app. Further implementation efforts yielded the incorporation of voice recognition functionality, enhancing the overall capabilities of the application.

**2. 프로젝트 내용**

**Model:**

Extensive preliminary research was conducted in the development of a Text-to-Speech (TTS) application focused on Korean webpage main content. Our approach began with a thorough examination of academic papers provided by the professor, aiming to comprehend the principles underlying Natural Language Processing (NLP) models and architectures. BoilerNet, the LSTM model at the core of our project, underwent training with GoogleTrends2017 raw HTMLs. Supervision involved feature vectors containing DOM-tree nodes, classifying nodes as either content or boilerplate. Initially considering the replacement of LSTM with a proven Korean-friendly model, we explored the use of a transformer model like BERT.

However, challenges emerged in incorporating Korean into BoilerNet. The dataset's limitations, particularly the insufficient HTMLs with feature vectors for supervised training of the language model, posed a significant obstacle. Additionally, the scarcity of Korean data further complicated our efforts. Drawing inspiration from the Japanese BoilerNet project, which translated GT2017 to Japanese for training, we developed scripts to translate GT2017 to Korean. These scripts enabled us to preserve the original meanings of webpages while expanding our dataset for more effective training of the Korean version of BoilerNet.

**App:**

The application we aim to develop through the project utilizes Text-to-Speech (TTS) to convert text extracted from web pages into audio. The ultimate goal of the project is to create an application that enables users to effectively listen to web content, with specific objectives including the implementation of the audio conversion feature, enhancement of user experience, multilingual support, and the addition of storage and sharing functionalities.

In the planning phase, we are using Android Studio to develop the application. Currently, the development is organized into UI layer, Domain layer, and Data layer, with the aim of implementing features in each layer to complete a fully functional application.

For foundational research, we are referring to the Android Studio website and acquiring basic knowledge about Android app development and TTS technology through books, papers, and online resources. Additionally, we are exploring research and development trends to overcome current technological limitations and staying informed about the latest technological advancements.

The planned development includes the integration of the TTS module, adding multilingual support, improving UI/UX design, and implementing storage and sharing features. Through these efforts, we aim to provide users with a convenient and intuitive application.

**3. 프로젝트의 기술적 내용**

**Model:**

In the development of our model, the primary programming language utilized was Python3, along with a diverse set of libraries supporting machine learning. To enhance computational efficiency, we leveraged the T4 GPU and occasionally the A100 GPU on Google Colab for model training. This project necessitated an in-depth exploration of the codebase and folder structure of open-source models.

A critical early challenge revolved around selecting an appropriate tokenizer conducive to the nuances of the Korean language. Initially planning to transition from the LSTM model to a transformer model like BERT, we encountered a hurdle during version migration due to Google Colab's discontinuation of support for TensorFlow versions below 2.0. Recognizing the time-intensive nature of migrating BoilerNet to a newer version, we opted to abandon this approach after a month of challenges.

Fortuitously, we discovered the Japanese BoilerNet project, serving as a valuable reference point to reinvigorate our idea. Their use of a multilingual tokenizer, enabling contextual understanding in Japanese, and the straightforward translation of raw HTMLs from GT2017 English to Japanese, while retaining essential feature vector encodings and HTML tags, inspired our revised strategy. We adapted the Japanese BERT tokenizer for our Korean dataset, incorporating various Python parsing and translating scripts developed in-house. This approach proved successful in overcoming our initial obstacles and aligning the model with our project objectives.

**App:**

Our project is developed using Android Studio with Java as the primary language. The choice of Android Studio enables the creation of a user-friendly and efficient Android application. The hardware aspect of the project considers devices running the Android operating system.

In terms of applied technologies, a central focus is on the Text-to-Speech (TTS) feature. TTS functionality is not inherently provided as a built-in feature for TTS in Android among the embedded APIs. Consequently, we implemented the TTS pause feature ourselves. To identify the paused position, we utilized commonly used punctuation marks '. ? !' to delineate sentences. Each occurrence of these punctuation marks increments a count, allowing us to track the last-read sentence. This approach ensures precise resumption of audio playback from the exact position.

One of the significant technical challenges encountered during development was the establishment of a robust backend server and the implementation of multi-threading. The backend server is crucial for data processing and ensuring smooth communication between the application and the TTS functionality. Multi-threading was particularly crucial for efficiently handling concurrent tasks. Challenges related to backend server setup and multi-threading were addressed through thorough research, consultation with relevant documentation, and iterative testing. The solutions were custom-developed to meet the specific requirements of our Android application, ensuring optimal performance.

The development environment, Android Studio, provides a comprehensive toolset for Android app development. Java serves as the primary programming language, offering compatibility and flexibility with the Android platform. TTS-related functionalities involve the integration of libraries and APIs facilitating seamless text-to-speech conversion. Dependency management, code optimization for efficiency, and ensuring compatibility across various Android devices were integral parts of the development process.

4. 프로젝트의 역할 분담

4.1 개별 임무 분담

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **번호** | **학과** | **학번** | **학년** | **이름** | **담당업무** |
| 1 | 컴퓨터소프트웨어 | 2017030128 | 4 | 이재욱 | 데이터셋 구축,  모델 관련 일들을 주업무로 담당 |
| 2 | 컴퓨터소프트웨어 | 2018008504 | 4 | 손승일 | 앱 관련된 일들을 주업무로 담당 |

4.2 개발 일정

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **구분** | **추진내용** | **수행기간** | | | | | | | | | | | | |
| **1월** | **2월** | **3월** | **4월** | **5월** | **6월** | **7월** | **8월** | **9월** | **10월** | **11월** | **12월** |
| 기획 | 프로젝트 내용 기획 |  |  |  |  |  |  |  |  |  |  |  |  |
| 공부 | 모델 관련된 내용 공부 |  |  |  |  |  |  |  |  |  |  |  |  |
| App 기획에 필요한 내용 공부 |  |  |  |  |  |  |  |  |  |  |  |  |
| 코딩 | 모델 |  |  |  |  |  |  |  |  |  |  |  |  |
| 앱 |  |  |  |  |  |  |  |  |  |  |  |  |
| 종료 |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

5. 결론 및 기대효과

**Model:**

This project afforded us firsthand experience in deconstructing, analyzing, and comprehensively understanding the intricacies of developing machine learning models. Unlike many previous machine learning and AI-related courses, this endeavor provided a unique opportunity to actively implement a model and engage with the datasets specific to our tasks. It allowed us to immerse ourselves in the field of machine learning as software developers, offering a practical perspective on model development.

Furthermore, the project presented the opportunity to plan, develop, and implement an application entirely from the ground up. Structuring the app to seamlessly integrate the model and requisite scripts added another layer of complexity, providing valuable experience in project management and software architecture.

In terms of our contribution to the model, we recognized a challenge in BoilerNet's unique feature vector-based supervised learning, particularly in the collection of large amounts of labeled data. Given that all data requires labeling, we explored methods to extend labels from labeled to unlabeled data for Large Language Models (LLMs). Our exploration led us to a paper from FairGNN, which incorporated a classifier from adversarial learning atop their generative model.

Acknowledging the outdated nature of LSTM in the NLP field, we propose that replacing LSTM with a more modern model represents a meaningful contribution to BoilerNet. This transition necessitates significant amounts of labeled data suitable for BoilerNet. Leveraging transfer learning from a small set of labeled data (GT2017) to a larger set of unlabeled data, our aim is to efficiently train a boilerplate removal model with enhanced multilingual support.

**App:**

During the project period, I had the opportunity to engage in the planning process, marking my first experience in direct involvement with the technical aspects of project planning. While I had prior experience in application development using Android Studio and Java, contributing to the planning phase and actively shaping user experience and feature development was a novel experience. This allowed me to gain a deeper understanding of the entire software development process and enhanced my collaboration and communication skills with team members.

In terms of potential applications and expected outcomes, the addition of speech recognition features ensured a seamless experience for users listening to web content. Recognizing that such functionality could be utilized in various scenarios, it became evident that users could consume content through speech while engaging in other tasks. This holds the potential to enhance user efficiency across different contexts.

Looking ahead to future plans, the focus will be on optimizing multi-threading. While the current implementation utilizes multi-threading to concurrently handle certain tasks, there is room for more efficient multi-threading to handle a broader range of tasks simultaneously and improve the overall performance of the application. Additionally, user feedback will be actively considered for UI/UX enhancements, and efforts will be directed towards the development of additional features.