Virtual Korea: A Novel Online Discussion Platform Based on an AI moderator

Seok Kim, Jin Woo Park, Chung Soo Lee, Chae Yun Jang

Capstone Design Project

Abstract. In modern society, as digital transformation accelerates in various fields, the role of an online discussion platform is increasing. Debate requires critical thinking and verification of information, and fair discussion is especially important. Therefore, in this paper, we propose [Bangguseok Korea], an online discussion platform that introduces artificial intelligence moderators using deep learning (deep learning) techniques in the field of natural language processing. AI moderator mediate discussions on behalf of humans through two roles. First, it masks slanderous expressions in real time and penalizes the panelists for this. Second, after the discussion is over, the pros and cons are summarized so that a final decision can be made quickly. In this paper, a discussion platform mobile application was developed and implemented to mediate discussions in real time through both models.

Keywords: AI moderator · Text Classification · Text Summarization.· contents/services

1 Introduction

Several online discussion platforms currently exist, but a human moderator hosts and moderates the discussion. In addition, in existing platforms, users are mostly anonymous, and there is a limit to the method of passively suppressing negative and slanderous expressions. In this paper, to solve this problem, we introduce the concept of an AI moderator using deep learning (Deep Neural Network) technology, and propose a Korean platform that can act as the two roles of the existing moderator, and use it as a mobile app. implemented First, the AI arbiter processes the masking if there is abusive expression from both sides and mediates both sides fairly. Second, by organizing and summarizing the content of the discussion, it helps the audience and the debater make decisions. The artificial intelligence model presented in this paper summarizes conversations beyond the level of ordinary people, has learned to detect and mask subdivided abusive expressions, and shows through experiments that it can replace a significant part of the role of the existing human moderator.

Fig. 1. The logo image of Bangguseok Korea.

2 Background and Related Work

2.1 Background

Existing discussion platforms have very few or no specialized apps or websites for discussion. In most cases, there are no functions for discussion except for simple functions such as conversation functions, time limits, and topic raising functions. Therefore, we would like to create an IOS app discussion platform that is specialized in discussion. In the existing discussion, the host interferes subjectively for fair discussion by blocking excessive commentator attitudes. However, human intervention is not always fair and neutrality is not guaranteed. Therefore, people want to see a fair and neutral debate as a good one. Maintaining fairness in discussion is a very difficult subject, but what if it is a computer that is not interested? The model made through machine learning is not the answer, but the fact that at least two commentators are treated the same and are not entangled in interests will certainly serve as an advantage for people. To this end, a discussion platform was designed to allow fair discussion with artificial intelligence. In addition, we have newly reorganized the discussion structure so that the discussion is beneficial and fun, creating a platform for everyone to enjoy. So we've created an app that allows people to have fair and fun discussions online.

2.2 Related work

Some of the existing online platforms where discussions are available also have simple features to select, talk and vote on topics; 1) they are not optimized for discussion (e.g., Kakao Talk open chatroom apps), or 2) discussion platforms without real-time discussion or winning team selection support (e.g. Debating Day website). In addition, there is no system to replace the role of the host or the role, and it consists of a simple method of expressing pros and cons. In this paper, the Republic of Korea in the corner of the room proposed by this paper provides a function of conducting discussions fairly and summarizing discussion conversations while conducting them in real time like real TV discussions. In the case of artificial intelligence models, there are many conversation summary models or non-verbal detection models that have been previously learned through a large language corpus, but most of them are based on English. In this study, the model was fine-tuned with domain-appropriate data for two reasons: 1) based on a Korean dataset, and 2) language that comes and goes in a special situation such as discussion.

Text Classification and Text Summarization The Text Classification model is a model that classifies categories of documents using deep learning. Since each text has a label, it is an easy task to calculate and learn accuracy. In particular, since we classify non-anisotropic expressions, we will use comments left by people with different thoughts after seeing the same context as a dataset. This is similar to sentimental analysis rather than a general classification task. Since the understanding of sentences takes precedence over sentence generation, we will use the Bert-series model. Text summarization is a task that summarizes documents, and although there is no correct answer to the document summary, it uses the reference summary to learn the standard and sample text of the summary. There are two representative Korean summary models that have been pre-trained about this. BART (Bidirectional and Auto-Regressive Transformers) learns in the form of an autoencoder that adds noise to some of the input text and restores it to the original text.

3 Design for the Proposed Service

3.1 Application

The App was designed using MVVM pattern with clean architecture. The MVVM pattern was adopted to break the dependence between View and Model, and the ViewModel was implemented independently from the UIKit framework to create a testable structure. In order for ViewModel to create a UI-independent structure, transition codes between two view controllers are positioned in navigator.

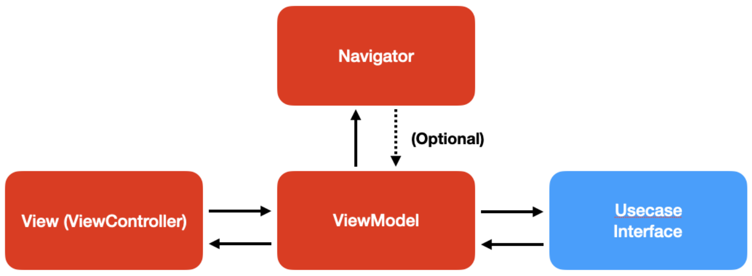


Fig. 2. mvvm pattern with navigator

The disadvantage of the MVVM structure is that there may be a problem that the ViewModel becomes massive. Clean Architecture can efficiently solve the problem. In addition, there is a need to create an app for macOS or an app for iPadOS, and the introduction of Clean Architecture has the advantage of efficiently reusing the Domain layer corresponding to business logic and the Data layer responsible for data processing logic. For these reasons, we adopt Clean Architecture for our mobile application.

The details of the Clean Architecture can vary from person to person. In this app, we use a method of dividing into domain, platform, and application layer.

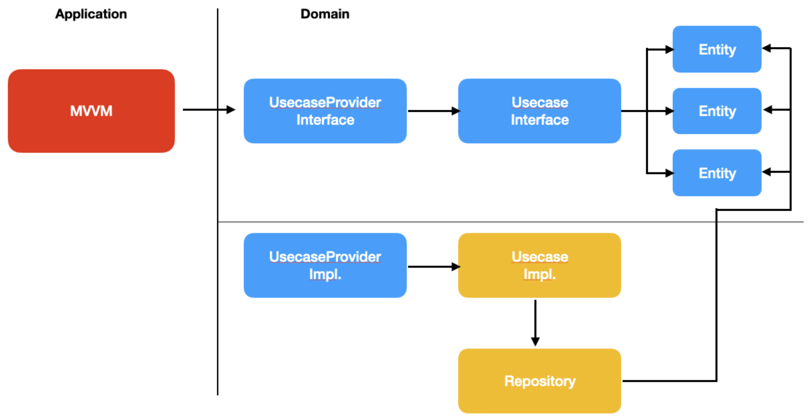


Fig. 3. clean architecture

Domain is a layer that defines the data structure to be used in the app with entity and expresses the functions that can be performed in the app with Usecase. Domain layer is not relying on UIKit or other layers, and usecase in domain layer consists of only Interface. Platform is a layer that implements Domain Layer using platform-dependent technologies such as iOS and macOS. Application is a layer that is in charge of input and output with the user. The Application Layer relies on the Domain Layer, which can make you not know how to implement it in detail. As such, it is the recommended design approach in object-oriented programming that makes it dependent on abstracted interfaces rather than on implementations.

Firebase Realtime Database and Firebase Cloud Storage were used to quickly implement chat and profile picture functions without separate chat servers and image servers.

3.2 Server

The model was distributed to the server. The method of embedding into an app has an overhead problem that must be stored on individual clients. In addition, performance problems arise because inference is made with the app's CPU or GPU. To address this problem, the model was deployed using TorchServe open source to service the PyTorch model.

The server distributed three models for text classification and one model for text summarization. The model for text classification consists of two artificial intelligence models (klue-roBERTa binary classification model, klue-roBERTa multi-label classification model) and a non-AI badwords classifier. In the text summarization, one artificial intelligence model (koBART) is distributed.

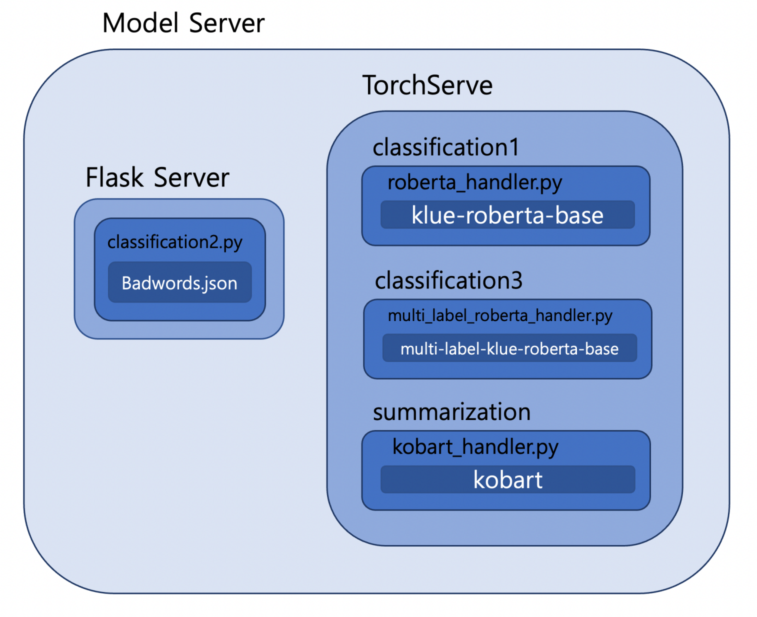


Fig. 4. Model Server Structure

3.3 Offensive Masking Model

The offensive masking model serves to detect and block toxic representations occurring in discussions. A classification layer was added to the bottom of the pre-trained roberta model with klue dataset, and fine-tuned with about 40K labeled as non-representation. Additionally, an additional model was created using a dataset of approximately 18K labeled with 11 non-specific classifications to further classify types of offensive. In order to effectively improve the performance of these two models, we update the weights using only the last four of the 12 attention layers of the Roberta model during fine tuning so as not to lose the pre-trained knowledge as much as possible. Following the total of two models, a swear dictionary containing swear words directly was constructed, and a model that detects non-malignancy through voting of the three models was selected as the final model. By adding this voting process, the sentence can be judged from a different perspective in that each model has been learned with different data, and through this, it is possible to effectively prevent using toxic comment in discussions.

3.4 Text Summarization Model

The discussion conversation summary model is a conversation summary model that helps the judges choose. In addition to the Korean Wikipedia of BART[4], the model used Kobart, a Korean-style BART that pre-learned various data such as news, books, everyone's corpus v1.0 (conversation, news, etc.) and the Blue House petition. BART is a denoising autoencoder that returns a corrupted document to an existing document, as seen in Figure 2. BART is implemented as a seq2seq model, and corrupted text is encoded by the bird directional encoder (BERT) and received by the left-to-right autoregressive decoder (GPT). For prior learning, the existing negative log lilihood was optimized. In the decoder, the ReLU activation function used by GPT was changed to GeLU and the parameter initialization was set to N(0,0.2). After contaminating the document, BART is trained to optimize the reconstruction loss, i.e., the output of the decoder and the cross entropy loss of the reference document. Unlike the denoising autoencoder, which used the existing customized noising scheme, BART can be applied to any type of document corruption. In extreme situations, even without all the information about the original, BART can behave the same as the original language model.

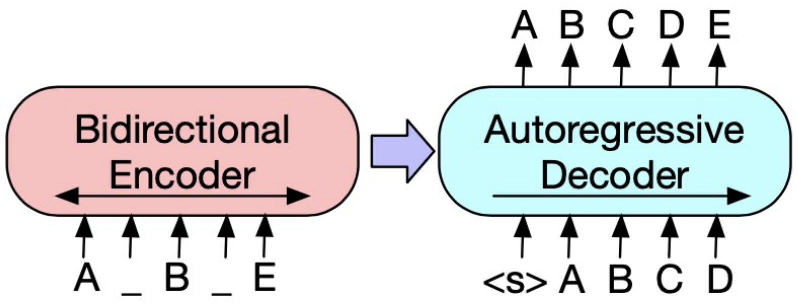
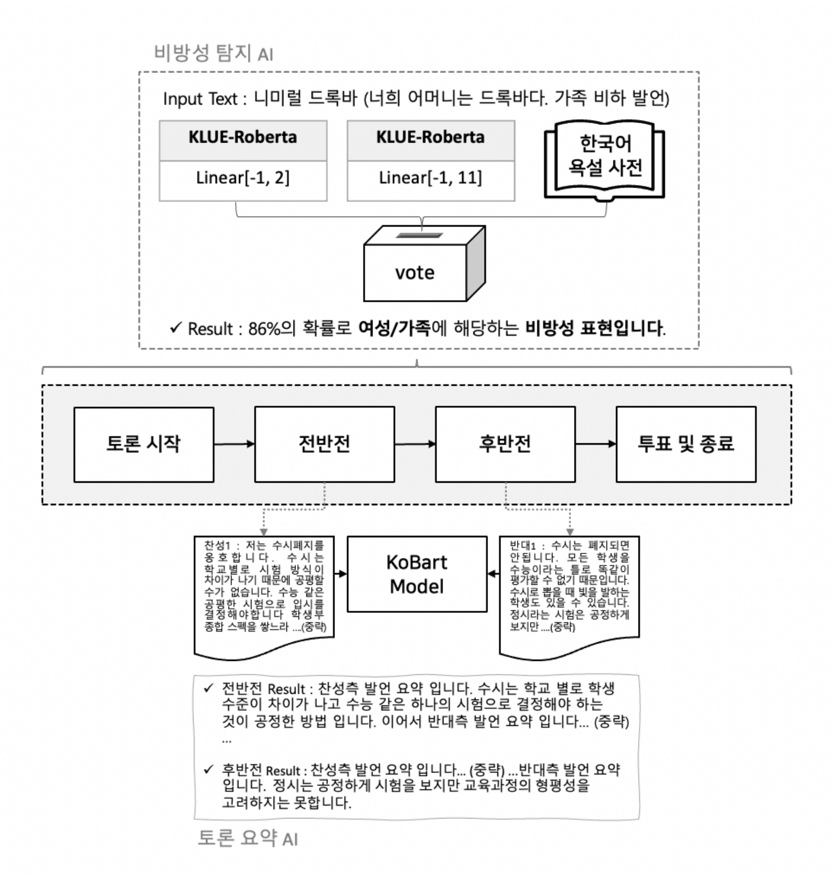


Fig. 5. BART achitecture.

3.5 The overall structure of our app

The discussion architecture we used is structured as shown in Figure 6. Prior to the start of the discussion, schedule an agenda to be discussed, the number of commentators, and make a reservation. Participants who want to discuss the subject make reservations in advance and conduct discussions before and after. When a non-malign expression is detected in all conversations during the discussion, the AI mediator judges it in real time and masks it. The first and second half consists of introduction, free discussion, and conclusion, and in between, participants take a break from strategic meetings. When the discussion is all over, the AI mediator summarizes the discussion. After the discussion, the judges vote, and the AI mediator informs whether to win or lose and ends the discussion.



Input Text : 니미럴 드록바 (너희 어머니는 드록바다. 가족 비하 발언)

✓ Result : 86%의 확률로 **여성/가족**에 해당하는 **비방성 표현입니다**.

찬성1 : 저는 수시폐지를 옹호합니다. 수시는 학교별로 시험 방식이 차이가 나기 때문에 공평할 수가 없습니다. 수능 같은 공평한 시험으로 입시를 결정해야합니다 학생부 종합 스펙을 쌓느라 ....(중략)

반대1 : 수시는 폐지되면 안됩니다. 모든 학생을 수능이라는 틀로 똑같이 평가할 수 없기 때문입니다. 수시로 뽑을 때 빛을 발하는 학생도 있을 수 있습니다. 정시라는 시험은 공정하게 보지만 ....(중략)

✓ 전반전 Result : 찬성측 발언 요약 입니다. 수시는 학교 별로 학생 수준이 차이가 나고 수능 같은 하나의 시험으로 결정해야 하는 것이 공정한 방법 입니다. 이어서 반대측 발언 요약 입니다... (중략)
...

✓ 후반전 Result : 찬성측 발언 요약 입니다... (중략) ...반대측 발언 요약 입니다. 정시는 공정하게 시험을 보지만 교육과정의 형평성을 고려하지는 못합니다.

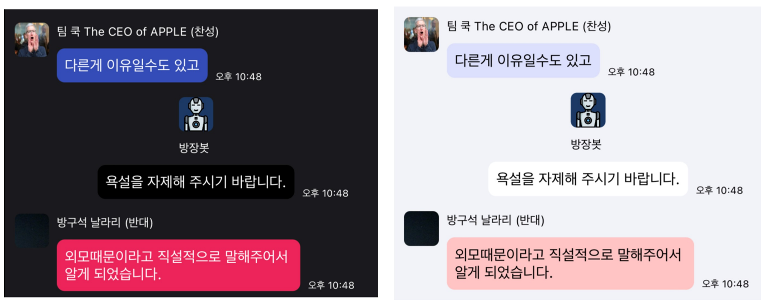
Fig. 6. Overall application architecture.

4 Implementation

4.1 Application

The iOS app continuously improved the UI using internal feedback for the best user experience, and was implemented in consideration of Apple's Human Interface Guidelines (HIG) and accessibility. For example, if the user turns on the app for the first time, a screen to set up the profile appears, preventing the user from using the app. So we implement the user can browse the app without setting the profile in consideration of HIG. In addition, it was implemented using a progress indicator so that the user could recognize that the work was in progress when performing a long-time task.

Examples implemented in consideration of accessibility include VoiceOver and Dark Mode support. VoiceOver refers to the function of reading the screen by voice assistant Siri, and can increase accessibility for users with low vision and visual impairments. In addition, Dark Mode refers to a UI in a dark environment, which can increase accessibility of users who are sensitive to bright light and users in an environment with dark lighting.



팀 쿡 The CEO of APPLE (찬성)
다른게 이유일수도 있고 오후 10:48
방장봇
욕설을 자제해 주시기 바랍니다. 오후 10:48
방구석 날라리 (반대)
외모때문이라고 직설적으로 말해주어서 알게 되었습니다. 오후 10:48

팀 쿡 The CEO of APPLE (찬성)
다른게 이유일수도 있고 오후 10:48
방장봇
욕설을 자제해 주시기 바랍니다. 오후 10:48
방구석 날라리 (반대)
외모때문이라고 직설적으로 말해주어서 알게 되었습니다. 오후 10:48

Fig. 7. Comparison of dark mode and light mode

So, our app supports profile setting, creating a chat room, chatting in a chat room, adding discussion in a chat room and discussion function.

4.2 Server

All messages spoken by users during the discussion must go through a text classification model. When a user sends a message from the app, it sends three http requests to the server. Models are distributed on the server, making an interference with the requests sent to each model, and sending a response back to the app. Then app hard-votes three responses so that the AI moderator can take appropriate action.

When the phase of the discussion is over, the discussion messages are combined for each user and a request is sent to the text summarization model.

Likewise, in the summary model, after an interference, the response is sent back to the app to output the summary of the AI moderator.

The model server performs a pre-process for http requests, an interference, and a post-process for the contents to be sent in response. These processes are contained in the handler file. The handler file, model file, and model parameters file are compressed into one mar file and distributed to the server.

The server was powered by the GTX-1050 Ti GPU in Windows10. Since the development environment for model distribution was linux, the compatibility problem was solved with docker. Since the GPU resource had to be used, nvidia-docker and others were installed so that it could be inferred using the GPU resource in the docker environment. The response time on the server is about 0.1 seconds on average for text classification model and 5 seconds for text summarization model. After that, the server was run through DDNS configuration and port forwarding, allowing requests to be sent to the server from anywhere.

4.3 Offensive Masking Model

Our offensive masking model is designed to alert users to their language habits by classifying toxic comments, not just whether it is toxic comments, but also by classifying the types of offensive and warning them specifically. In addition, following the establishment of two artificial intelligence models, a abusive dictionary was established to classify new words or non-verbal terms that the model did not learn, adding functions for continuous improvement of the Host bot.

4.4 Text Summarization Model

The data we use to train the summary model is a text dataset to train AI to understand text and generate core content. The first dataset we used is the document summary dataset about 300K, a summary dataset with a domain similar to a discussion dialogue that deals with suggestive issues among the Korean dataset disclosed on AIhub. The second dataset is 50K, a summary of the creation of Korean documents posted on DACON. Of the total 350K datasets, we excluded datasets from domains that might interfere with discussion conversation summaries and used them for learning.

5 Evaluation

Offensive comment classification we evaluate offensive classifier as two criteria. first is F1 score and second is accuracy. The reason that we check accuracy score is this score is used to measure the model performance in terms of measuring the ratio of sum of true positive and true negatives out of all the predictions made. our model have to give the warning to user so it is important that model doesn't say real not true as true. our result could see below. and because binary model is better performance than multi-label model. we conclude using two model both.

model	binary model	distilled binary model	multi-label model
F1 score	81.03	49.65	72.02
Accuracy	0.82	0.73	0.50

Table 1. Evaluation Metric

Debate conversation Summarization We set three evaluation criteria for a good summary statement for the evaluation of the discussion conversation summary model and evaluated by humans accordingly. Is the evaluation criteria 1) well identified and summarized the key keywords? 2) Does the summary contain the arguments and grounds in the original text? 3) Does it summarize the contents of the original text by compressing it well without bringing it as it is? The test dataset used for evaluation collected 2.5K of discussion conversation data from the debating day and randomly evaluated 200 samples among them. The score was evaluated under the assumption that the general person's summary evaluation ability was about 5 points based on a perfect score of 10, and the evaluation score of each model is shown in Table 1 below. The models have different epoch for each version, in order of 10, 20, 30, 40, and 50. We used version 4 with the highest score.

model	human	version 1	version 2	version 3	version 4	version 5
Human Evaluation Score	5.0	4.38	5.64	5.93	6.71	6.17

Table 2. Human evaluation score

6 Limitations and Discussions

Offensive masking model In the offensive masking model, there have often been cases where it is detected as abusive expression, not abusive expression. Each and every one of our models did not produce entirely reliable performance. Thus, as a suboptimal, we sought to detect abusive expression by taking a policy that hard-voting two artificial intelligence models and one non-artificial intelligence classifier. However, despite the application of these policies, there have often been cases of incorrect detection of abusive expression. Two solutions were devised. First, it is determined that the problem can be solved by lowering the threshold of the artificial intelligence model, and the threshold is determined as a heuristic evaluation method. But the fundamental problem lies in the data. In the end, therefore, it will be necessary to increase the performance of the model by adding and training more refined datasets.

Text summarization model In the text summarization model, there were occasional cases of vague summarization. The text summarization model was learned with a well-written article and a summary of the article. However, conversations on the discussion platform are far from well-written writing. Therefore, in this case, a vague summary was progressed. While some people are good at making logical arguments, others will make their own arguments in a hurry. The room manager bot should also summarize the rambling. There are two solutions designed. The first is to improve the performance of the text summarization model by adding a daily conversation summary to the dataset, making the interactive discussion message also good at summarizing. Second, instead of improving the performance of the model, it is the direction of refining the input message well. By limiting the right to speak in the debate and providing an environment in which one can logically organize and argue, well-written text can enter the input of the model.

Server The limitation in server is that our servers are personal desktop. The response time averaged about 0.1 seconds for text classification models and less than 5 seconds for text summarization models, but very occasionally, the response time for text summarization models took more than 30 seconds. It is difficult to know whether it is a problem with the network or the inference process. However, if the platform expands, a stable server will have to be used.

Application Lastly, in the application, there is a performance inefficient part of the app. when a user enters the chat room, all chat data of the room is loaded at once. The current implementation is somewhat regrettable because it is more efficient to load and display the last 50 or so chat data, and then support additional fetching functions when the user scrolls up for the previous chats.

7 Conclusion

In the corner of the room, Korea is a new mobile-based online discussion platform that combines the advantages of deep learning technology and existing discussion methods. This paper designs a discussion architecture that not only secures the fairness of discussion through the introduction of artificial intelligence mediators, but can also form a sound discussion culture by suppressing non-expressions arising from anonymity. In addition, the Republic of Korea in the corner of the room can reduce entry barriers to discussion by simplifying the discussion structure based on the excellent summary performance of AI moderators.

References

1. Devlin, Jacob, et al. Article title."Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
2. Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).
3. Liu, Yang. "Fine-tune BERT for extractive summarization." arXiv preprint arXiv:1903.10318 (2019).
4. Park, Sungjoon, et al. "Klue: Korean language understanding evaluation." arXiv preprint arXiv:2105.09680 (2021).
5. Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-tosequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, (2019).
6. [Summarization model] https://github.com/SKT-AI/KoBART
7. [Alamofire] https://github.com/Alamofire/Alamofire
8. [Firebase] https://github.com/firebase/firebase-ios-sdk
9. [Kingfisher] https://github.com/onevcat/Kingfisher
10. [RxKeyboard] https://github.com/RxSwiftCommunity/RxKeyboard
11. [RxSwift] https://github.com/ReactiveX/RxSwift
12. [SideMenu] https://github.com/jonkykong/SideMenu
13. [SnapKit] https://github.com/SnapKit/SnapKit
14. [Torchserve] https://github.com/pytorch/serve
15. [Badwords] https://github.com/organization/Gentleman/blob/master/resources/badwords.json
16. [Icon] https://www.flaticon.com/authors/konkapp