Report on QIA2023: MBTI Prediction

Hyeonbin Hwang
School of Computing, KAIST
hbin0701@kaist.ac.kr

Abstract

*The competition was comprised of two phases - first phase focusing on the question-level prediction, and the latter of user-level prediction. By utilizing **klue**'s **roberta-large**, we fine-tune the given models, and achieve 78% and 56% of accuracy in the public test set. We demonstrate this has much potential space to be improved and explored yet if more specialized open-sourced models on korean were available. Yet at the same time, we raise the question of whether phase 2 dataset annotation is credible and valid.*

1. Datasets

The given dataset consists of N users, including their **MBTI** type, **age** (rounded to the nearest tenth), and their **responses to 60 predetermined questions**. Each user's response consists of a Yes, No, or Neutral option, followed by a text explanation providing further details about their answer. All the questions and answers are written in Korean.

1.1. Question-Level Prediction

For phase 1, there were 11,520 instances for training, comprised of 48 question-answer pairs of 240 different users. There are 15 users for each of 16 different MBTIs. The test set had 12 remaining question-answer pairs for the same 240 different users, which added up to 2,880 instances in total. However, the User_ID of each instance was hidden, so that the task objective would be to predict MBTI of the respondent based on a **single question-answer pair.**

1.2. User-Level Prediction

For phase 2, there were 7,200 instances for training, comprising of 120 users and their 60 question-answer pairs. In the similar manner, there were 7,200 instances for testing. Thus, the objective of phase 2 was to predict the MBTI of the user based on the **60 question-answer pairs** given.

1.3. Statistics

Now the primary concern is whether there exists data leakage between phase1 and phase 2. There was no data leakage, meaning that therew as no user overlap between phase 1 and phase 2. Yet, for phase 2, among 120 users of train dataset, **2 of them overlapped with its test dataset**, which is a major problem. More interestingly, **while all 60 question responses are exactly the same, their genders and age were presented differently from the training set, which leads us question the validity of the phase 2 dataset as a whole.** We will discuss about this further in Section 4.

Other than that, from a Web Page, we have figured out which dimension of MBTI each question tries to explore, and have visualized the correlation of the short answer depending on the user's MBTI as preliminary step. (Figure 1-4) As shown in the Figure, there are certain questions whether only one side favors specific answer (i.e. Yes), and the other side favors its counterpart (i.e. No). For example, consider question 16 for I vs. E.

However, there is no definite correlation nor formula that divides one from its complement, thereby suggesting need for considering multiple questions at the same time, and adhering to the Long Answer, the textual detail written by the user themselves.

2. Model

Despite of various attempts, we have empirically concluded that the best performance can be achieved by adopting conventional framework for both phase1 and phase2, with a slight modification from the baseline's architecture. Figure 3 displays the model architecture for phase1 and phase2, respectively. For phase 1, we concatenate gender, age, question, short answer and long answer then feed into our encoder. After, we utilize the [CLS] token embedding, which then we pass to 4 separate linear layers for each MBTI dimension prediction.

For phase 2, we separate the 60 question-answer pairs of a given user into 4 bins with respect to its dimension of interest (i.e. I/E, S/N, T/F, J/P) then randomly sample 5 question-answer. After concatenating them in random order, we feed into our main encoder followed by linear layer to perform binary classification. We have each model specialized for dimension-wise prediction, thus to fully predict

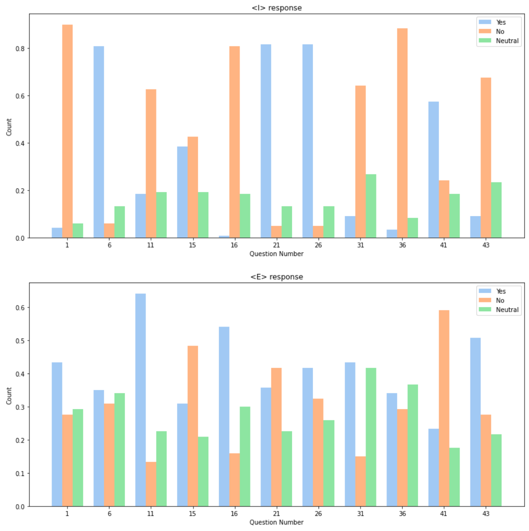


Figure 1. Question distribution on I vs. E (Introverted vs Extroverted)

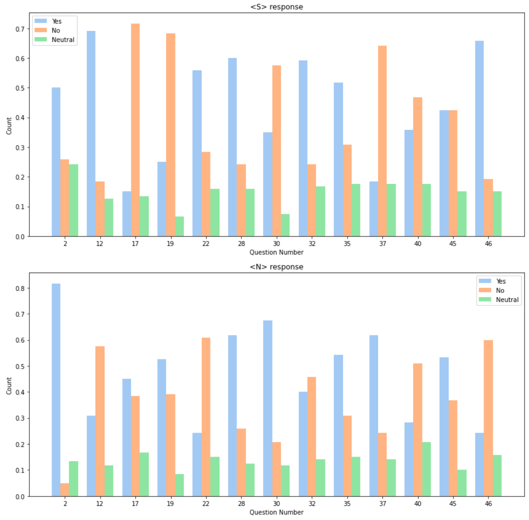


Figure 2. Question distribution on S vs. N (Sensing vs Intuitive)

one's MBTI, we repeat this process for each dimension, in total of 4 times.

As the training mostly suffered from early-stage overfitting, our primary focus lied on testing with various input forms and backbone models, which could minimize overfitting. In this light, we have concluded respective current input forms and **klue/roberta-large** [4] yielded the best result.

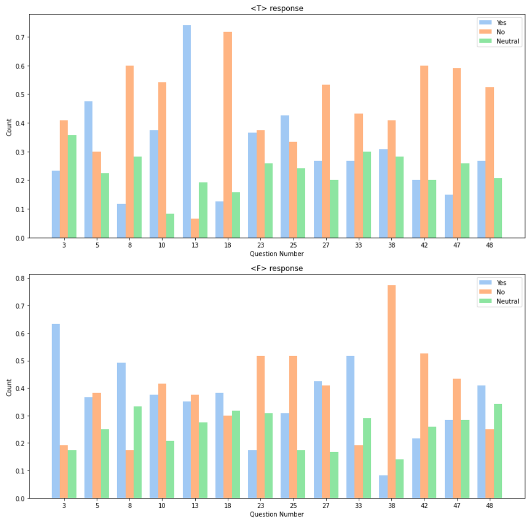


Figure 3. Question distribution on T vs. F (Thinking vs Feeling)

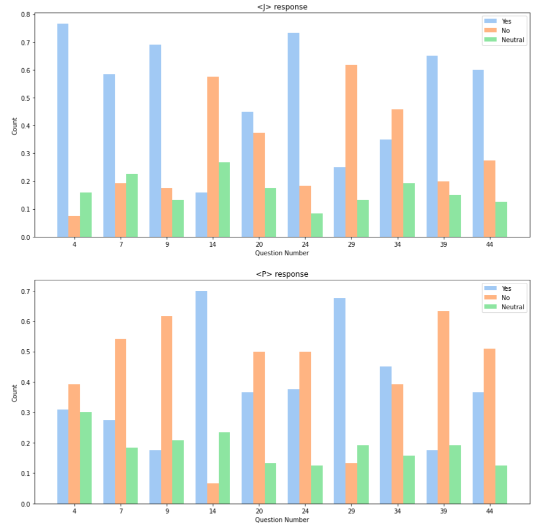


Figure 4. Question distribution on J vs. P (Judging vs Perceiving)

Dimension	No. of Questions
I/E	13
S/N	16
T/F	17
J/P	14

Table 1. Number of questions that pertain to each dimension of MBTI

3. Training Scheme

For hyperparameters, we have utilized Adam as optimizer with learning rate of 1e-5, batch size of 16, with

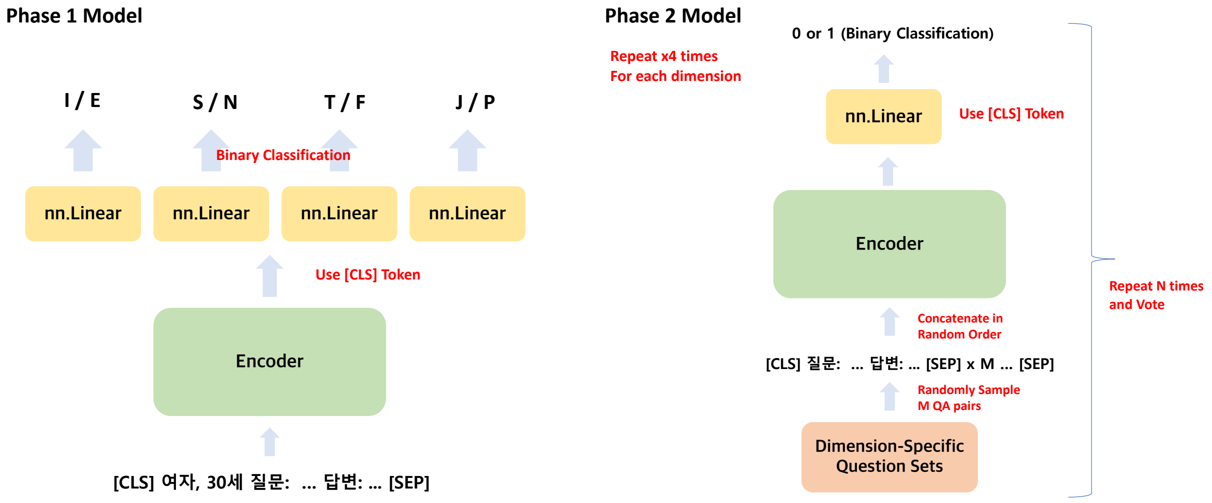


Figure 5. Framework used for MBTI prediction of Phase 1 (Left) and Phase 2 (Right). We use gender and age information only for question-wise prediction as the users are the same for train and test time. For phase 2, we sample 5 questions from dimension specific question subsets (Shown in previous figures) and concatenate them to feed into encoder. Also, for phase 2 we perform dimension-specific models due to input length constraint.

LambdaLR where learning rate is reduced by power of 0.95 each epoch. We also apply dropout of 0.1 for all linear layers. For phase 1 and 2, training was done for 10 epochs and 15 epochs, with max input token length of 144 and 512 respectively. We select the model checkpoint with the best validation accuracy, where training and validation set are divided in 5-fold.

On phase 1, we further utilized **soft-voting** scheme with 5-fold validation to produce a floating point score. On phase 2, our dataloader length was determined by number of unique users, thus making only $120 * 0.8$ (training set ratio) $= 96$ instances available for each epoch. To increase sample size, we repeat this selection process for every user for 10 times (non-consecutively, of course) within a single epoch. Then, when validating and testing, for every user we repeat this selection for 15 times, and average out the resulting score (from 0-1) and round to nearest integer (0 or 1) to produce its final prediction.

4. Results

4.1. Phase 1

For Phase 1, the accuracy was pretty high. Single train-valid split yielded public test accuracy of about **-70%**. Running K-Fold (K=5) cross-validation and using regression approach (predicting a floating-point value) rather than a binary label (0 or 1) increased this value up until **78%**. Even though the prediction first seemed impossible as it is hard for one to predict all the letters, the model seemed as if it were able to somehow predict one's MBTI even with a single question.

However, it is worth to note that removing the gender and age significantly hindered training and led to lower validation accuracy, and we suspect that the model is doing short-cut learning as the users are the same between train and test dataset, thus exploiting 'common' information between these two may facilitate the prediction process.

Also, prior to this, we have tested the model with various baselines, including **klue/bert-base**, **klue/roberta-base**, **klue/roberta-large**, **monologg/kobigbird-bert-base**, and roberta-large seemed to give the best result. We observed that the generalization power increases with scale, thus if we had enough resource to fine-tune larger models (i.e. KoGPT from KakaoBrain), we would have obtained stronger results.

After release of phase 2 data, we also tried to incorporate Phase 2 data for phase 1 training (as it also contains responses to question 48-60, which are the question sets in the test data set), yet using these ironically led to valid. accuracy of **67%** (vs. 70% using only phase 1 data), suggesting there is a certain discrepancy between phase 1 data and phase 2 data, or in the worst case, there may be flaws in the phase 2 dataset, which we will describe further in the next section.

4.2. Phase 2

For Phase 2, this was no different than random prediction. The training could be done, yet the validation consistently remained around 0.45-0.55. Of course, this all depended on the type of dataset selected for validation dataset;

however, public test set accuracy resembled valid accuracy ONLY IF we used **solely phase 2 data** as validation.

Now consider these three different settings: **a)** Using phase 1 data as training (240 users), and phase 2 data as validation. **(b)** Mixing phase 1 and phase 2 data and randomly spliting training (240 users) and validation (120 users). **(c)** Only using phase 2 data and randomly splitting training (96 users) and validation (24 users).

When training with the framework in Figure 5, in setting a), the val. accuracy remains between 0.45-0.55, as aforementioned; in b), val. accuracy goes up to 70%. Yet the subset accuracy of the validation set that belongs to phase 2 data users, is still close to 0.5. In c), it is similar to a), which may be due to lack of data or again, **there may be flaw in the phase 2 dataset in general.**

We have attempted to vary the number of questions sampled, ranging from N=1 to N=10, and also have tested to sample for a user more than 15 times (i.e. T=30, 50, 100). While with bigger values of N and T, the validation accuracy seemed to increase, yet no strong correlation was found in the midst of this phase 2 data inconsistency - and the test leaderboard score. Now that we have confirmed some form of inconsistency exists AT MINIMUM between phase 1 data and phase 2 data, it is possible to inquire whether this phenomena occurs due to the methodology itself. Thus we have used two different approaches to confirm this phenomena.

Using ChatGPT We have tested on the ChatGPT for MBTI prediction, which does not require any training. In light of [3], which argues that using english instruction is better even for tasks in other languages, we utilize following instruction: *”Below you are given a series of responses used for MBTI test questions. Your task is to determine whether the user is more likely to be extroverted (E) or introverted (I). Generate answer only without producing any explanation.* *n*\” Note that we have only tested for I/E of simplicity as I/E dimension yielded highest performance in setting b). When we test the accuracy of ChatGPT for phase 1 data, the accuracy was **0.7375**, while for the phase 2 data, the accuracy was **0.5**, which indicates random prediction.

Using ML Model As many existing testing methods do not require explanation yet make the user to perform a likert scale evaluation of given questions (i.e. *https://www.16personalities.com/*), we also adopt attempt to predict the MBTI with Short Answers. After converting ”No” to 0, ”Neutral” to 0.5, ”yes” to 1, We train 4 models, one for each dimension and we only select the subset of questions that aim to explore each dimension. As phase 1 train dataset only contains question no. from 1-48, we have previously refrained from using 49-60, but later we fill all the responses of those missing data to 0, which gives higher results than the former.

We report the full result on table 2, where we observe similar or worse phenomena as using the NLP-based approach with **klue/roberta-large**. It is hard to explore any correlation. Since we did not optimize the model based on validation accuracy (i.e. using optuna [1]), but directly report validation accuracy after fitting the train data on LGBM model, **this indicates there is too much divergence or few to no correlation among phase 2 dataset (both train and test).** Note that setting (a) is data-balanced, while setting (b) may or may not be data balanced due to random split.

	Setting (a)	Setting (b)	
Dimension	Val Acc.	Val Acc.	P2* Acc.
I/E	0.6125	0.7638	0.3636
S/N	0.4958	0.6667	0.5909
T/F	0.525	0.7083	0.4545
J/P	0.5958	0.7639	0.4545
Average	0.5572	0.7257	0.4659
Public Test Acc.	0.45853		0.5159

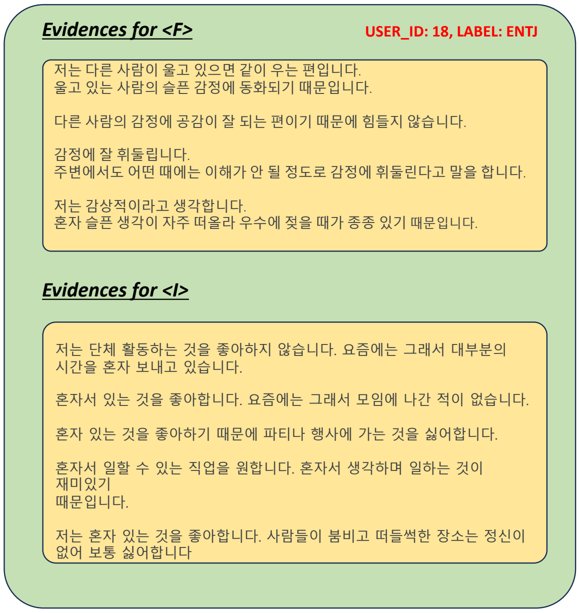
Table 2. Results of validation accuracy and testing accuracy when testing LGBM on setting (a) and setting(b) *P2 refers to phase 2 subset accuracy.

Qualitative Analysis This discrepancy can be explained in two different ways. First interpretation is that it requires much more difficulty to predict the MBTI of the user for phase 2 than for phase 1. However we believe that the case is very unlikely, as both datasets seem to be of same types, only differing in the methodology of spliting train / test. Another one, which is more plausible (even if it can be misleading) is that phase 2 dataset contains errorneous labeling. This is corroborated by manually looking at some of the data. (Figure 6) The sample is from training set of phase 2 data - even if we are fully aware of lack of expertise in the domain and all the questions will need to be considered in bigger scheme to make a final judgment, there seem to be strong evidences to suggest that user may be ”I” and ”F”, contrary to ”E” and ”T” as suggested by the annotated label.

In conclusion, in spite of having reached about 56% of accuracy in *public* test leaderboard using framework introduced in Figure 5, yet there is almost no correlation that can be connected to the training, suggesting there may be flaws in the phase 2 dataset, especially in the presence of annotation error in 1.3.

5. Innovative Parts AND Failures

For description of innovative parts and failures, we combine the two sections as the innovative attempts mostly resulted without success. While we describe our innovative attempts, we also provide an analysis of the possible cause of failure.



Evidences for <F> USER_ID: 18, LABEL: ENTJ

저는 다른 사람이 울고 있으면 같이 우는 편입니다.
울고 있는 사람의 슬픈 감정에 동화되기 때문입니다.

다른 사람의 감정에 공감이 잘 되는 편이기 때문에 힘들지 않습니다.

감정에 잘 휘둘립니다.
주변에서도 어떤 때에는 이해가 안 될 정도로 감정에 휘둘린다고 말을 합니다.

저는 감상적이라고 생각합니다.
혼자 슬픈 생각이 자주 떠올라 우수에 젖을 때가 종종 있기 때문입니다.

Evidences for <I>

저는 단체 활동하는 것을 좋아하지 않습니다. 요즘에는 그래서 대부분의
시간을 혼자 보내고 있습니다.

혼자서 있는 것을 좋아합니다. 요즘에는 그래서 모임에 나간 적이 없습니다.

혼자 있는 것을 좋아하기 때문에 파티나 행사에 가는 것을 싫어합니다.

혼자서 일할 수 있는 직업을 원합니다. 혼자서 생각하며 일하는 것이
재미있기
때문입니다.

저는 혼자 있는 것을 좋아합니다. 사람들이 붐비고 떠들썩한 장소는 정신이
없어 보통 싫어합니다

Figure 6. Qualitative Analysis of data sample. While the label is done in ENTJ, looking at specific questions the user seems to have clear evidences for 'I' and 'F'.

5.1. Phase 1

For phase 1, we initially considered the task to be **retrieval + clustering** task. The reasons are as follows:

Retrieval As predicting all 4 MBTI letters based on a single question response is deemed to be impossible, we rather diverted our focus to exploit the fact people with same (or similar) MBTI would respond to a question in a very similar way. Also, we further hypothesized that since the users are the same between train and test time: there could exist the possibility that for a given user u who responded to question $q_{test,i}, i \in [49, 60]$ with the textual description a_i in the test set, we thought the same user u could have responded to a different question $q_{train,i}, i \in [1, 48]$ with a'_i that would result in the maximum similarity with a_i. Of course we don't know which response was generated by which user on the test set, but if we could retrieve relevant user from the train set for each response, we thought this problem could be solved.

For each response in the test set, we effectively make our retrieval corpus smaller by using three information, question, gender, and age. For a given pair of question and response of interest (q. a) in the test set, we first retrieve K most relevant questions from $q_{train,i}|i = 1, 2, ..48$, and limit our corpus to consist of the responses to these K questions and have been responded by the users that have same age and gender as that who responded to a. We utilize **klue/Roberta-Large** to extract embedding of the questions and responses, and after use cosine similarity to find nearest K relevant questions and M responses. We tried K = 1,3,5 and M = 1, 5, 10, 25; and after retrieving M responses, we perform soft voting of user MBTIs from these responses.

Clustering Also considering that the 240 users are comprised of equal number of MBTIs, We also opted to use clustering for assigning given 240 users into 16 bins of MBTI, where 15 user should be assigned to each bin. For this, we used a varaint of KMeans where we forcefully assign same size of clusters.

Failure However, this resulted in almost random prediction, reaching at most 55% of accuracy.W ithout fine-tuning the encoder model, the model is still able to retrieve similar sentences which we have confirmed through manual inspection. Yet we suspect that the reason lies in that there are so many texts that resembles the text that make the majority voting ambiguous. Also clustering has led to consistent performance drop, even when combined other approaches, such as the framework in Figure 5. We hypothesize that manually forcing labels in the midst of insufficient accuracy leads to worse results. Also, it is disadvantageous to use this method as this method leads to binary classification (1,0) rather than regarding this problem as regression (0-1) which consistently proved to yield higher performance.

Overall, we believe instead of doing manual extraction and matching with Retrieval and Clustering, it is better to let the model automatically capture the subtle textual semantics and nuances in its parameter by using a conventional finetuning method.

5.2. Phase 2

For phase 2, as aforementioned, various attempts were made with ChatGPT, Machine Learning method (LightGBM), and AI model. As the first 2 methods were somewhat covered in Section 4, we focus on the various attempts done on AI model, mainly the variations on that introduced in Figure 5. We do not particularly discuss the efficacy of each attempt (if we do, it's on the setting (b) validation acc.), as we could not truly "distinguish" what is really effective in light of the inconsistency in phase 2 dataset.

Input Form One of the main elements was **what** to train. In other words, as there were much more information to predict MBTI than phase 1, (60 QA pairs vs. single QA pair), the key was to how to effectively utilize these information. At the same time the maximum input length of our backbone encoder was 512, so there was a constraint on how much of the answers we were to going to utilize. (We also tried KoBigbird which has input length of 4096, but we did not see particular performance gain.) While we initially thought both incorporating question and the answer helps overcome overfitting issue as repetitive text spans would appear across different MBTIs, at later stages of the experiment, we instead increased the number of answers to

10, concatenated them with *[SEP]* token, while excluding the question part. Then without the question, ”Yes”, ”No”, ”Neutral” would add more confusion, thus we excluded this short answer part also from the input text. In setting (b), this approach yielded higher performance.

Model Design The reason we thought incorporating [SEP] token was to let model know that the answers are discrete for different questions. Especially we considered this to be crucial when not using the question part. We also opted to put [CLS] token in front of every response, so that using N=10 responses would result in 10 [CLS] tokens. We then use encoded embeddings of these N tokens and apply mean pooling before passing it to Linear layer. However, we did not particularly see performance gain neither. We believe that using the first [CLS] token is enough considering that MHA is done in global scale and the length (512) is not that long. As aforementioned, repeating T times of extracting binary label led to performance increase with larger T in setting (b), and we believe this played similar effect to ensemble voting.

6. Conclusion

In this competition, comprised of phase1 and phase2, we mainly utilized **klue/roberta-large** as our backbone and on top of that apply other techniques that best fit the task characteristic. We reach 4^{th}, and 6^{th} on public test leaderboard using method introduced in Figure 5. We also show the other creative techniques that can be employed for the task, yet needs to be explored further to give better performance than our main method. Our work is limited on the basis that we assume much of the phase 2 dataset is flawed with multiple clear evidences, yet if this is not true, this would indicate that phase 2 dataset is just more challenging and thus requires a more sophisticated modeling method that can appropriately handle edge cases. For future work, it would be beneficial to verify the validity of the phase 2 dataset through expert inspection, and also to explore the application of larger models to the phase 1 dataset, which we couldn’t due to limit of computational budget. (i.e. KoGPT [2])

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019. 4

[2] Ildoo Kim, Gunsoo Han, Jiyeon Ham, and Woonhyuk Baek. Kogpt: Kakaobrain korean(hangul) generative pre-trained transformer. https://github.com/kakaobrain/kogpt, 2021. 6

[3] Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning, 2023. 4

[4] Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Taehwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Lucy Park, Alice Oh, Jung-Woo Ha, and Kyunghyun Cho. Klue: Korean language understanding evaluation, 2021. 2