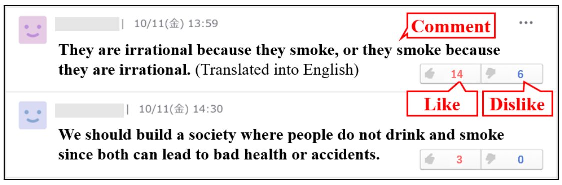
A Case Study of In-House Competition for Ranking Constructive Comments in a News Service

Hayato Kobayashi[1*] **Hiroaki Taguchi**[1*] **Yoshimune Tabuchi**[1] **Chahine Koleejan**[1]
Ken Kobayashi[1] **Soichiro Fujita**[2] **Kazuma Murao**[3] **Takeshi Masuyama**[1]
Taichi Yatsuka[1] **Manabu Okumura**[2] **Satoshi Sekine**[4]
[1]Yahoo Japan Corporation [2]Tokyo Institute of Technology [3]VISITS Technologies Inc. [4]RIKEN
{hakobaya, htaguchi, yotabuch, ckoleeja, kenkoba, tamasuya, tyatsuka}@yahoo-corp.jp
{fujiso@lr., oku@}pi.titech.ac.jp murao@vis-its.com satoshi.sekine@riken.jp

Abstract

Ranking the user comments posted on a news article is important for online news services because comment visibility directly affects the user experience. Research on ranking comments with different metrics to measure the comment quality has shown "constructiveness" used in argument analysis is promising from a practical standpoint. In this paper, we report a case study in which this constructiveness is examined in the real world. Specifically, we examine an in-house competition to improve the performance of ranking constructive comments and demonstrate the effectiveness of the best obtained model for a commercial service.



[redacted] | 10/11(金) 13:59

They are irrational because they smoke, or they smoke because they are irrational. (Translated into English)

[redacted] | 10/11(金) 14:30

We should build a society where people do not drink and smoke since both can lead to bad health or accidents.

Figure 1: Comments on Yahoo! JAPAN News for article "Lifting the ban on drinking/smoking at 18."

1 Introduction

In online news services, the user comments posted on news articles function as a type of useful content known as user-generated content (UGC). Figure 1 shows examples of comments posted on Yahoo! JAPAN News, a Japanese news portal.[1] By reading these comments along with the article, users can obtain supplementary information such as other users' opinions, experiences, and simplified explanations of the article. There is a limit, however, on the number of comments that can be displayed on a page, and as users typically do not have the time or inclination to read through all the comments, ideally they should be ranked in some way. Prioritizing the comments for display is directly linked to user satisfaction, so improving this ranking is an important issue for such services.

There have already been multiple studies on comment ranking in online news services and discussion forums (Hsu et al., 2009; Das Sarma et al., 2010; Brand and Van Der Merwe, 2014; Wei et al., 2016). All of these studies have utilized user feedback (e.g., "Like"-button clicks in Figure 1) as their ranking metrics. Although such user feedback is easy to obtain, this type of measurement has two drawbacks: (i) user feedback does not always satisfy the service provider's needs, such as to create a fair place (i.e., a news space that is neutral), and (ii) user feedback will be biased by where comments appear in a comment thread (also known as "position bias" (Craswell et al., 2008)). A typical example for (i) can be seen in political comments, where the "goodness" of the comment tends to be decided on the basis of the political views of the majority of the users rather than on its quality. A typical example of (ii) can be illustrated by a case where earlier comments tend to receive more feedback since they are displayed at the top of the page, which implies later comments will be ignored irrespective of their quality. To resolve this issue, Fujita et al. (2019) introduced a metric representing a comment's constructiveness (see Section 2 for details), which has also been studied in argument analysis (Kolhatkar and Taboada, 2017a; Napoles et al., 2017a). Interestingly, they found empirical evidence that the constructiveness has no correlation with the user feedback, which has been commonly used for ranking comments. This implies that we need to consider the constructiveness rather than the user feedback to avoid unfavorable situations (i) and (ii) in real services.

In this paper, we take their study one step further towards practical application. Specifically, in collaboration with Yahoo! JAPAN News, we report a case study of deploying a model that ranks constructive comments in a commercial service. The

[*] Equal contribution.

[1] https://news.yahoo.co.jp/

Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media, pages 24–35
Online Workshop, June 10, 2021. ©2021 Association for Computational Linguistics

characteristic unique point of our study is that we aim to improve the ranking quality through an in-house competition. As represented by Kaggle (Kaggle, 2020), the machine learning competition platform, it has become common to improve a model's performance through a competition format. This kind of experiment has also been conducted in various research areas through shared-task workshops, with the WMT translation task (Barrault et al., 2019), TAC text analysis task (Demner-Fushman et al., 2018), and NTCIR information retrieval task (Kato and Liu, 2019) being well-known examples. Following this trend, we also aim to improve the ranking performance through a competition format. As this kind of work conducted within a company towards a commercial service is rarely released in the form of an academic paper, we expect our findings to become valuable knowledge for practitioners in the field. We clarify the novelty of our study against other previous studies in Section 7.

Our main contributions are as follows:

- We report the details of the in-house competition (i.e., constructive comment ranking task) conducted in a commercial news service, Yahoo! JAPAN News, where we obtained a new model with a 2.73% improvement in performance (NDCG) compared to the baseline (Section 3). We also administer a participant survey and discuss positive and negative opinions relating to this competition (Section 6).
- We consider several ensembles of the submitted models and show that the best one performed better than the best single model (Section 4). Nevertheless, the service does not find it reasonable for practical use considering the need for maintainability and low latency against the performance increase (0.62%). This suggests that while an ensemble of various models submitted in the competition is promising in an academic sense, it still has challenges in an industrial sense. We believe that this will open a new direction for the ensemble research field to solve such challenges.
- We demonstrate that the high-performance models in the competition are practically useful in the real world with a service perspective evaluation (Section 5), and in fact, the service decided to introduce the best single model.
- We will release the 59K labeled dataset and the models submitted in the competition for future research.[2]

Precondition	• Related to article and not libelous
Main conditions	• Intended to stimulate discussions • Objective and supported by fact • New idea, solution, or insight • User's unique experience

Table 1: Conditions for constructive comments.

2 Preliminaries

Constructiveness: We use the concept of constructiveness to prioritize comments that provide insight and encourage healthy discussion. According to the dictionary (Oxford, 2020), the term "constructive" is defined as "*having or intended to have a useful or beneficial purpose*." However, this dictionary definition is a bit too generic to determine whether a comment is constructive or not. To avoid individual variation as much as possible, we need a more specific definition for our task. Thus, we follow a previous study (Kolhatkar and Taboada, 2017a) on constructiveness, where a questionnaire administered to 100 people clarified the detailed conditions for constructive comments. Table 1 shows a summarized version of the conditions, which was also used by Fujita et al. (2019). The conditions consist of one precondition for maintaining decency and relevance and four main conditions for representing typical cases of being constructive. Specifically, a constructive comment is defined as one that satisfies the precondition and at least one of the main conditions.

YJCCR Dataset: We use (part of) the YJ Constructive Comment Ranking (YJCCR) Dataset, which was created by Fujita et al. (2019). The YJCCR dataset consists of more than 100K Japanese comments labelled with a **constructiveness score (C-score)**, which is a graded numeric score representing the level of constructiveness for ranking comments. The C-score was defined as the number of crowdsourced workers who judged a comment as constructive in response to a yes-or-no (binary) question. As a consequence, the C-score indicates how many people think that a comment is constructive with the goal of sufficiently satisfying as many users as possible.

The detailed settings of the crowdsourcing were as follows. The task was prepared with questions referencing a news article and its comments extracted from Yahoo! JAPAN News and conducted on a crowdsourcing service. The workers were asked to read the definition of constructiveness and then judge whether each comment was con-

[2] https://research-lab.yahoo.co.jp/en/software/

structive. To ensure reliability, only the results of serious workers who correctly answered quality-control questions that were randomly included in each task were kept. Ten workers were used for each comment in the dataset, so a C-score of 8, for example, means that eight workers judged a comment as constructive. The reliability of this annotation was confirmed with Krippendorff's alpha, which was "moderate agreement."

The comments in Figure 1 are actual ones in the YJCCR dataset. The lower comment has a high score (9) because it includes a constructive opinion with some reasoning, whereas the upper comment has a low score (0) since it includes offensive content (see Appendix C for more examples).

3 In-House Competition

Task: The competition task consisted of ranking comments based on their degree of constructiveness, that is, the **C-score** defined in Section 2. Specifically, given that we have training data with triples $\{(a, x, y)\}$ consisting of a news article a, a comment x on the article, and its corresponding C-score y, the task is to predict the ranking of comments for every article in the test dataset $\{(a, x)\}$, where the C-scores are unknown. The goal of this task is to create a model that predicts the correct ranking from the training data as closely as possible.

The competition was held for about six weeks (Dec. 13, 2018 – Jan. 23, 2019), and a dozen employees related to the comment ranking service were made aware of it. The information shared among them included not only the dataset but also sample code consisting of a simple feature extraction, model creation, and evaluation pipeline in order to reduce the burden on the participants. We also prepared a leaderboard to display the latest evaluation results for submitted models. The participants reported their evaluation results on the leaderboard and were able to update them any number of times during the competition period.

Dataset: The training dataset consisted of a combination of the above-mentioned public dataset YJCCR and a new dataset of long comments created for this study. We used 49,215 comments (9,845 articles with five comments each) from the YJCCR dataset, each comment having a C-score assigned by crowdsourcing. While this dataset only contained comments up to 125 characters in length, we noticed in our preliminary experiments that long comments tended to be incorrectly determined as constructive despite having a bigger impact on visibility than short ones. For that reason, we additionally extracted long comments (from 126 up to the maximum of 400 characters) posted to the articles in YJCCR and created a long comment dataset with C-scores assigned by crowdsourcing in the same way as for YJCCR, as described in Section 2. The resulting combination of the above two datasets yielded 59,120 comments (9,845 articles with an average of six comments). We split it into 80% training data, 10% validation data, and 10% test data to form the competition dataset.

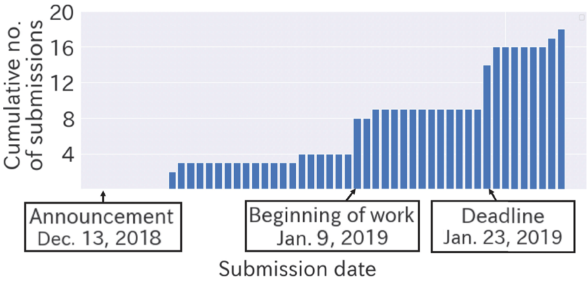


Figure 2: Cumulative number of submissions over the competition period.

Evaluation: We used Normalized Discounted Cumulative Gain (NDCG) (Burges et al., 2005), which is a widely used evaluation measure for ranking tasks. In this competition, we adopted a variant defined as NDCG@$k = Z_k \sum_{i=1}^{k} \frac{2^{r_i}-1}{\log_2(i+1)}$, which was also used in the Yahoo! Learning to Rank Challenge (Chapelle and Chang, 2011). This NDCG@k computes how close the top k comments predicted by a model are to the correct ranking, where r_i is the true C-score of the comment with predicted rank i, and Z_k is a normalization term.

To simplify the evaluation process, we set the average value of NDCG@k, i.e., $\frac{1}{K} \sum_{k=1}^{K}$ NDCG@k, as the main measure in the competition, where K is the number of comments included in the article. Furthermore, to particularly encourage the performance improvement for long comments, we extracted a dataset consisting of only long comments (305 articles, 917 comments) from the test data and used its NDCG@k value as a supplementary measure. This was meant to reduce the effect of submitting sloppy methods that merely determined long comments to be constructive. From here on we call the normal measure NDCG and the one for long comments NDCG-L.

Submitted Models: Eight individuals participated in the competition and submitted 14 models during the competition period (before the deadline). Figure 2 shows the total number of submissions across the competition period. We can see that the number of submissions was low during the initial period of the competition but increased significantly at the start of the year (beginning of work), a period where time is relatively more available (Jan. 9, 2019), and on the day of the deadline (Jan. 23, 2019). Moreover, after the submission deadline had passed, several participants continued to work on the task and created an additional four models. We included these additional models when carrying out our analysis, although only the models submitted before the deadline were eligible for internal awards. We obtained a wide variety of models created by the participants' trials and errors, but due to space limitations, we only discuss in detail the four highest-performing models, which were `Model-4`, `Model-11`, `Model-14`, and `Model-17`. The following list includes the summary of each model with its detailed settings and features (see Appendix A for their hyperparameter settings).

- `Model-4`: The model with the highest NDCG (before the deadline). It is a gradient boosting model (pairwise learning) with features based on pretrained word embeddings.
  Model: The model was a LambdaMART model (Burges, 2010), which is a boosted tree variant of LambdaRank (Burges et al., 2007) extended from RankNet (Burges et al., 2005). It was trained using RankLib (ver. 2.1) (Lemur Project, 2020), a library of "learning to rank" algorithms.
  Features: The features were based on pretrained word embeddings trained with fastText (ver. 0.2.0) (Facebook, 2020), an open-source library,that includes a subword-based extension (Bojanowski et al., 2017) of the skip-gram model (Mikolov et al., 2013). The training dataset consisted of 100M news articles in the service, and they were split into words using MeCab (ver. 0.996), a Japanese morphological analyzer (Kudo et al., 2004; Kudo, 2020a), with IPADIC (ver. 2.7.0). Finally, the features of each comment were set to the average vector of the pretrained word embeddings for the words in the comment.
- `Model-11`: The model with the highest sum of NDCG and NDCG-L. It is a linear rankSVM (Lee and Lin, 2014) model (pairwise learning) with features based on C-score prediction and the distance between an article and its comment, where this setting is a kind of stacking ensemble.
  Model: The model was an L2-regularized L2-loss linear rankSVM model that was implemented as an instance of the well-known SVM tool LIBLINEAR (ver. 2.1.1) (Lin, 2020). The cost parameter C was determined from $\{2^{-13}, \dots, 2^{1}\}$ on the basis of the performance on the validation set.
  Features: The features consisted of two factors. The first was the expected C-score, which was determined by first computing the probabilities of C-scores (considered as classes) using the open-source library fastText (ver. 0.2.0) (Joulin et al., 2017; Facebook, 2020)[2] with word embeddings trained on news articles and then calculating their expected value. The second feature was the Euclidean distance between the comment and title vectors, each of which consisted of the frequencies of words.
- `Model-14`: The model with the highest NDCG-L. It is a gradient boosting model (pointwise learning) with features based on maximal substrings and words.
  Model: The model was based on LightGBM (ver. 2.2.1) (Microsoft, 2020; Ke et al., 2017), a tree-based gradient boosting framework. The parameters were hand-tuned with a tuning guide (LightGBM Doc., 2020).
  Features: The features were based on a combination of maximal substrings and words, where a maximal substring is a substring s whose superstring never occurs at the same frequency as s. The features of the maximal substrings were the number of unique substrings, the frequencies of substrings, and the tf-idf values of substrings in the character-based maximal substrings in each comment (see Appendix A for how to extract maximal substrings). The features of words were the frequencies of words, which were extracted by MeCab (ver. 0.996), a Japanese morphological analyzer, with IPADIC (ver. 2.7.0). Finally, those two kinds of feature were combined and scaled to the range of $[-1, 1]$ using svm-scale in LIBLINEAR (ver. 2.1.1), a feature-scaling library.
- `Model-17`: The model with the highest NDCG (after the deadline). It is a variant of the RankNet model (pointwise and listwise learning) with features based on subwords.
  Model: The model was a variant of RankNet,

which has an encoder-scorer structure consisting of BiLSTMs and Gated CNNs (see Appendix A for the detailed model structure). C-score was predicted by (a) extracting the representations of the input subwords, (b) obtaining one vector averaging their representations, (c) estimating the classification probabilities, regarding the prediction problem of the C-score (0–10) as an 11-class classification problem, and (d) calculating the expected C-score with the probabilities. The loss was a combination of a pointwise loss, i.e., cross entropy loss for C-score probabilities, and a listwise loss, i.e., permutation probability loss for comment lists (Cao et al., 2007). The optimizer was Adam (Kingma and Ba, 2015) with parameters ($\alpha = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$), and the training was done in ten epochs with early stopping after random initialization in the range of $[-0.01, 0.01]$, where the batch size was 32 and the dropout rate was 0.3.

Features: The features (input) were a sequence of subwords based on SentencePiece (ver. 0.1.8) (Kudo and Richardson, 2018; Kudo, 2020b), where the subword model was trained with the training data using the unigram language model algorithm with the vocabulary size of 5,000.

Comparison with Baseline: We analyzed how well the submitted models performed compared to the baseline described below.

- `Baseline`: A linear rankSVM model (pairwise learning) with features based on term-frequency vectors. It was almost the same as the model in the previous study (Fujita et al., 2019) but was tuned for this competition.

  Model: The model was an L2-regularized L2-loss linear rankSVM model, which was implemented in LIBLINEAR (ver. 2.1.1). The cost parameter C was determined from $\{2^{-13}, \ldots, 2^{1}\}$ on the basis of the performance on the validation set.

  Features: The features consisted of the frequencies of words in each comment. Note that this setting performed better than the one-hot representations, the fractions (normalized frequencies) of the words, the number of distinct words, the tf-idf values, and any combinations thereof. They were scaled to the range of $[-1, 1]$ by using svm-scale in LIBLINEAR.

Figure 3 shows the performance increase (%) in NDCG and NDCG-L for the submitted models compared to `Baseline`. Note that decreases are not shown. As we can see, many models performed better than `Baseline`. Interestingly, a high NDCG score did not necessarily correspond to a high NDCG-L score, and in fact, `Model-4` with a high NDCG in particular had a lower NDCG-L than `Baseline`. The use of the leaderboard had a positive effect for participants submitting high-performance models for both measures in the latter half of the competition (right sides of the graphs). In the end, the highest performance increase was 2.73% by `Model-17` for NDCG and 2.34% by `Model-14` for NDCG-L.

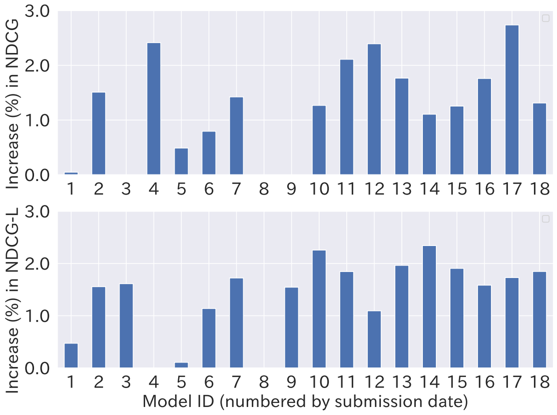


Figure 3: Increase (%) in NDCG (top) and NDCG-L (bottom) for each model compared to `Baseline`.

4 Model Ensemble

To further improve the performance, we considered using an ensemble of the models submitted in the competition. For ease of implementation, we focused on unsupervised ensemble methods that combine predicted scores. Assuming practical use, we only used the models that could accurately (or stably) reproduce their leaderboard performance, resulting in ensembles of 12 models.

Ensemble Methods: We prepared various ensemble methods covering both commonly used and recently proposed ones as follows.

- `ScoreAve`: Use the average of the predicted scores of all models as an ensemble score.
- `NormAve`: Use `ScoreAve` after normalizing the scores (Burges et al., 2011). We treated the predicted scores for all comments in each article as a vector v and applied the L2 norm, i.e., $v/||v||_2$.
- `RankAve`: Use the inverse of the averaged rank after ranking all comments with each model.
- `TopkAve`: Use `ScoreAve` only for the top-k ranked comments by each model (Cormack et al.,

	NDCG	NDCG-L	NDCG@3	Prec@3
Baseline	81.63	86.74	81.09	73.30
Model-4	83.60	82.15	82.79	73.98
Model-11	83.35	88.34	82.93	73.20
Model-14	82.53	**88.77**	81.83	72.86
Model-17	83.86	88.24	83.27	72.01
ScoreAve	83.85	86.66	83.20	73.40
NormAve	84.33	88.41	84.01	**74.11**
RankAve	83.46	88.25	82.92	73.30
TopkAve	84.35	88.35	83.31	73.54
PostEval	84.32	88.64	83.88	73.91
WeightEval	**84.38**	88.30	**84.18**	74.04

Table 2: NDCG variants (%) and precision (%) for (a part of) the submitted models and their ensembles.

2009), where k was chosen with the validation set.

- PostEval: Select the most promising output (or model) per article with a continuous version of majority voting (Kobayashi, 2018), where the similarity of two outputs was calculated with NDCG.
- WeightEval: Use the weighted average of the top-k promising outputs (Fujita et al., 2020), where k was chosen with the validation set. This method is a hybrid of output selection (PostEval) and output average (NormAve), where NDCG was used as a similarity function for selecting and weighting.

Evaluation Measures: Along with NDCG and NDCG-L, we used NDCG@3 and Prec@3 as supplementary measures, since only the top three comments are displayed first on each article page in the actual service, although users can read all comments on the next comment list page. Prec@3 is defined as the proportion of the predicted top-3 comments being in the correct top-3. Note that Järvelin and Kekäläinen (2002) reported that NDCG is more suitable than precision for graded scores like in our setting.

Results: Table 2 lists the results of the four high-performance models in Section 3 and the six ensembles of submitted models. Looking at the ensemble models, we can see that the recently proposed WeightEval performed the best for the main measure NDCG, and NormAve also performed competitively despite its simplicity. ScoreAve and RankAve did not perform as well as NormAve, as ScoreAve did not adjust outputs with different scales and RankAve failed when trying to adjust them, ignoring score shapes. These results imply that score adjustment (NormAve, TopkAve) and model selection (PostEval, WeightEval) contributed to the performance improvement. As a whole, NormAve is the most promising for practical use, since TopkAve and WeightEval need parameter tuning. Looking at single models, all the models performed better than Baseline for the main measure NDCG, and Model-17 performed the best overall. The differences between Baseline and Model-17 and between Model-17 and NormAve for the main measure NDCG were statistically significant in a Wilcoxon signed-rank test ($p < 0.05$). The high NDCG-L of Model-14 seems to be related to how to make the features. Model-14 used maximal substrings, including longer text spans than ordinary words. This implies that Model-14 can successfully characterize long comments, even if it might be harmful for short ones. We may need to consider this effect for other tasks including only long texts, although it was not effective for the main measure NDCG of our task since most comments are short.

5 Towards Practical Use

To determine if the submitted models can be used in the running service, we carried out a qualitative evaluation from the perspective of service, not just constructiveness. Specifically, we prepared the comment lists ranked by candidate models for each news article and asked three experts in the comment service to rank them. We instructed the experts to evaluate them on the basis of “which list should be provided as a service” rather than “which list is constructive,” as the goal of this evaluation was to improve the service quality. As an evaluation measure, we calculated the micro-average of the ranks by the experts over the evaluation data prepared separately from the competition data. We used 104 articles (each having 3,406 comments on average) for the first evaluation and 66 articles (each having 3,888 comments on average) for the second evaluation.[3]

Baseline vs. Naive Methods: We first examined whether the constructiveness ranking model Baseline is useful compared to other naive methods, which was confirmed by Fujita et al. (2019) in terms of automatic evaluation (NDCG) only. Specifically, we compared the four models described below in terms of human evaluation.

- Feedback: A model ranking basically in descending/ascending order of the number of

[3] We reduced the number of articles in the second round because the evaluation cost was too high.

	Average Rank
Feedback	2.61
Latest	3.42
Length	2.20
Baseline (C-score)	**1.77**

Table 3: Qualitative evaluation results of Baseline and naive methods (lower ranks are better).

	Average Rank
Baseline	3.86
Model-4	3.64
Model-11	3.63
Model-14	3.41
Model-17	**3.11**

Table 4: Qualitative evaluation results of submitted models and Baseline (lower ranks are better).

Likes/Dislikes. This model has been used in the service.

- Latest: A model ranking in descending order of comment date. This model is a naive method used when user feedback and constructiveness scores are not available.
- Length: A model ranking in descending order of comment length. This model is a naive method based on the rule of thumb that long comments tend to be constructive.
- Baseline: A model ranking in descending order of predicted C-score, which is almost the same as the model in the previous study (Fujita et al., 2019) but has been tuned to this competition.

Table 3 shows the results of the qualitative evaluation. We can see that Baseline clearly performed better than the other models. The differences between Baseline and Feedback and between Baseline and Length were statistically significant in a Wilcoxon signed-rank test ($p < 0.05$). These results mean that the finding in the previous paper holds true even in human evaluation.

Baseline vs. Submitted Models: We prepared the four high-performance single models in Table 2 (excluding ensemble models) for comparison with Baseline. We also suggested introducing the most promising ensemble model, NormAve, but the service preferred not to because it would be unreasonable to maintain 12 different models and to re-normalize the scores every time a comment was posted, where static scores must be stored in the DB due to the low latency constraint.

Table 4 lists the results of the qualitative evaluation. As shown, the best single model for NDCG, Model-17, also had the best (lowest) average rank. The difference between Baseline and Model-17 was statistically significant in a Wilcoxon signed-rank test ($p < 0.05$). This implies that a competition format is effective in terms of obtaining an improved model even when we consider service-level judgment. As a result, the service introduced Model-17 into its comment ranking module.

One of the reasons Model-17 performed better than the others seems to be related to the fact that it had a full neural structure (as explained in Section 3), which implies "robustness" (or expressiveness of the model) thanks to a lot of parameters, as in Neyshabur et al. (2017)'s study. In fact, the evaluators reported that Model-17 had few critical errors compared to the other models. Although Model-4 and Model-11 performed well in Table 2 (automatic evaluation), we will have to consider the robustness (or the number of critical errors) from a practical point of view. Note that the detailed investigation of these factors is beyond the scope of this study.

6 Participant Survey and Future Issues

After the competition, we collected opinions from the participants through an optional survey. We discuss certain positive and negative opinions in detail below (see Appendix B for other opinions).

Positive Opinions: The most popular opinion was that the number of model submissions was greater than initially expected. According to the participants, this was mainly due to the game element of the competition, i.e., publicly competing against other participants. In other words, the fun of the task was an implicit incentive to encourage submissions. As a result, we were able to use a wide variety of models for the ensemble experiment (Section 4), which seems to have contributed to the performance improvement. Another interesting opinion was about disclosure of the modeling methods. In this competition, the participants were encouraged to include model descriptions such as structures and features when reporting their evaluation results on the leaderboard. This information helped the participants make improved models, which contributed to the best performance of single models (Section 3). Other positive opinions were related to the improved knowledge and skills acquired by the participants.

Negative Opinions: One major negative opinion was about the leaderboard system, where the participants individually posted their own results pertaining to the evaluation tool and test data. This setting allowed the participants to purposefully design models effective only on the test data, although we confirmed that they actually used the validation data for fine-tuning. To hold a competition on a larger scale, we should prepare an automatic evaluation system with private test data. Such a setting is relatively common in strict competitions such as Kaggle, while most test datasets tend to be publicly available in research communities (under research ethics). Another insightful opinion was to make an incentive for exploring new directions, since it is valuable to obtain findings in unknown/rare directions, even if the results are not superior. In addition, model diversity can contribute to the ensemble performance, as discussed above. We suggest preparing a special prize for novelty in order to encourage exploring different directions.

7 Related Work

Constructiveness: Analyzing the comments on online news services or discussion forums has been extensively studied (Wanas et al., 2008; Ma et al., 2012; Llewellyn et al., 2016; Shi and Lam, 2018). In this line of research, many studies have focused on ranking comments (Hsu et al., 2009; Das Sarma et al., 2010; Brand and Van Der Merwe, 2014; Wei et al., 2016). However, the prior approaches have been based on user feedback, which is completely different from constructiveness.

Constructiveness has been introduced in argument analysis frameworks (Napoles et al., 2017a,b; Kolhatkar and Taboada, 2017a,b; Kolhatkar et al., 2020). The purpose of these studies was to classify constructive comments, whereas Fujita et al. (2019) recently expanded their tasks to a ranking one. They created a new dataset for ranking constructive comments on a news service and showed that the commonly used method that ranks comments by user feedback does not contribute to constructiveness in terms of automatic evaluation (NDCG). Our study has value as a deployment report of their approach, and we also confirmed that constructiveness performed better than user feedback for ranking comments in terms of human evaluation by experts.

Aside from constructiveness and user feedback, we may consider hate speech detection (Kwok and Wang, 2013; Nobata et al., 2016; Davidson et al., 2017) and sentiment analysis (Fan and Sun, 2010; Siersdorfer et al., 2014) as alternative approaches for analyzing the quality of comments on the basis of their content. Although these approaches are useful for other tasks, they do not directly solve our task, namely, ranking constructive comments. For example, the simple comment “Great!” is positive and is not hate speech, but it is not suitable as a top-ranked comment in our task.

Shared Tasks and Competitions: There have been many competitions in various research areas through shared-task workshops, such as the WMT translation task (Barrault et al., 2019), TAC text analysis task (Demner-Fushman et al., 2018), and NTCIR information retrieval task (Kato and Liu, 2019). Their purpose to find good models for a specific task is almost the same as ours, and the main difference (ignoring the task) is that the competition in our work was conducted within a company. As this kind of work towards a commercial service is rarely released in the form of an academic paper, we expect that our findings will become valuable knowledge for practitioners in this field.

As for “learning to rank” tasks, there have also been several competitions such as the Internet Mathematics 2009 (Yandex, 2020), the Yahoo! Learning to Rank Challenge (Chapelle and Chang, 2011), and the Personalized Web Search Challenge (Kharitonov and Serdyukov, 2020). Their tasks are basically to rank pages in terms of relevance to a search query, which is common in the information retrieval field. In contrast, our task is to rank comments in terms of constructiveness. It has value in the sense of applying the concept of argument analysis in the real world.

A unique aspect of our work is the ensemble of submitted models in the competition. Although there have been many studies on model ensembles (Hoi and Jin, 2008; Cormack et al., 2009; Burges et al., 2011), the models for prior ensemble experiments were basically prepared by either random initialization or a researcher’s preference, which is different from our competition setting. The most closely related study involves the concept of “Resource by Collaborative Contribution (RbCC)” (Sekine et al., 2019), which collaboratively creates a large-scale dataset for named entity recognition by using the predicted labels of submitted models in a shared task, although their purpose and task were completely different from ours. We believe our findings in a commercial service will be useful for future ensemble studies.

8 Conclusion

We reported a case study of an in-house competition for ranking constructive comments. Our experimental results showed that the competition format is effective for testing various model structures, and that ensembling submitted models can further improve the ranking performance. Moreover, we confirmed that the submitted models were practically useful in a service perspective evaluation.

Acknowledgements

We would like to thank the comment service team in Yahoo! JAPAN News for their continued support of our research. We would also like to thank the anonymous reviewers for their constructive comments.

References

Alfred V. Aho and Margaret J. Corasick. 1975. Efficient String Matching: An Aid to Bibliographic Search. *Communications of the ACM*, 18(6):333–340.

Shunsuke Aihara. 2020. pykwic. https://github.com/shunsukeaihara/pykwic. Accessed: Apr. 1, 2020.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pages 1–61. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Dirk Brand and Brink Van Der Merwe. 2014. Comment Classification for an Online News Domain. In *Proceedings of the First International Conference on the Use of Mobile Informations and Communication Technology in Africa*, pages 50–55. Stellenbosch University.

Christopher J. C. Burges. 2010. From RankNet to LambdaRank to LambdaMART: An Overview. Technical Report MSR-TR-2010-82, Microsoft.

Christopher J. C. Burges, Robert Ragno, and Quoc V. Le. 2007. Learning to Rank with Nonsmooth Cost Functions. In *Advances in Neural Information Processing Systems 19 (NIPS 2007)*, pages 193–200. MIT Press.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to Rank Using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 89–96. ACM.

Christopher J. C. Burges, Krysta Svore, Paul Bennett, Andrzej Pastusiak, and Qiang Wu. 2011. Learning to Rank Using an Ensemble of Lambda-Gradient Models. In *Proceedings of the Learning to Rank Challenge*, pages 25–35.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pages 129–136. ACM.

Olivier Chapelle and Yi Chang. 2011. Yahoo! Learning to Rank Challenge Overview. In *Proceedings of the Learning to Rank Challenge*, pages 1–24. PMLR.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 758–759. ACM.

Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-Bias Models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM 2008)*, pages 87–94. Association for Computing Machinery.

Anish Das Sarma, Atish Das Sarma, Sreenivas Gollapudi, and Rina Panigrahy. 2010. Ranking Mechanisms in Twitter-like Forums. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*, pages 21–30. ACM.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)*, pages 512–515. AAAI Press.

Dina Demner-Fushman, Kin Wah Fung, Phong Do, Richard D. Boyce, and Travis Goodwin. 2018. Overview of the TAC 2018 Drug-Drug Interaction Extraction from Drug Labels Track. In *Proceedings of the 2018 Text Analysis Conference (TAC 2018)*.

Facebook. 2020. fastText. https://github.com/facebookresearch/fastText. Accessed: Apr. 1, 2020.

Wen Fan and Shutao Sun. 2010. Sentiment classification for online comments on Chinese news. In *Proceedings of the 2010 International Conference* *on Computer Application and System Modeling (ICCASM 2010)*, volume 4, pages V4–740–V4–745. IEEE.

Jerome H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29:1189–1232.

Soichiro Fujita, Hayato Kobayashi, and Manabu Okumura. 2019. Dataset Creation for Ranking Constructive News Comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 2619–2626. Association for Computational Linguistics.

Soichiro Fujita, Hayato Kobayashi, and Manabu Okumura. 2020. Unsupervised Ensemble of Ranking Models for News Comments Using Pseudo Answers. In *Proceedings of the 42nd European Conference on Information Retrieval (ECIR 2020)*, pages 133–140. Springer International Publishing.

Steven C. H. Hoi and Rong Jin. 2008. Semi-supervised Ensemble Ranking. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI 2008)*, pages 634–639. AAAI Press.

Chiao-Fang Hsu, Elham Khabiri, and James Caverlee. 2009. Ranking Comments on the Social Web. In *Proceedings of the 2009 International Conference on Computational Science and Engineering (CSE 2009)*, volume 4, pages 90–97. IEEE.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 427–431. Association for Computational Linguistics.

Kaggle. 2020. Kaggle: Your Home for Data Science. https://www.kaggle.com/. Accessed: Apr. 1, 2020.

Makoto P. Kato and Yiqun Liu. 2019. Overview of NTCIR-14. In *Proceedings of the 14th NTCIR Conference (NTCIR 2019)*.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 3146–3154. Curran Associates, Inc.

Eugene Kharitonov and Pavel Serdyukov. 2020. Personalized Web Search Challenge. https://www.kaggle.com/c/yandex-personalized-web-search-challenge/overview/description. Accessed: Apr. 1, 2020.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.

Hayato Kobayashi. 2018. Frustratingly Easy Model Ensemble for Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 4165–4176. Association for Computational Linguistics.

Varada Kolhatkar and Maite Taboada. 2017a. Constructive Language in News Comments. In *Proceedings of the First Workshop on Abusive Language Online*, pages 11–17. Association for Computational Linguistics.

Varada Kolhatkar and Maite Taboada. 2017b. Using New York Times Picks to Identify Constructive Comments. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 100–105. Association for Computational Linguistics.

Varada Kolhatkar, Nithum Thain, Jeffrey Scott Sorensen, Lucas Dixon, and Maite Taboada. 2020. Classifying Constructive Comments. *arXiv*, abs/2004.05476.

Taku Kudo. 2020a. MeCab: Yet Another Part-of-Speech and Morphological Analyzer . https://taku910.github.io/mecab/. Accessed: Apr. 1, 2020.

Taku Kudo. 2020b. SentencePiece. https://github.com/google/sentencepiece. Accessed: Apr. 1, 2020.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 66–71. Association for Computational Linguistics.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Proceedings of EMNLP 2004*, pages 230–237, Barcelona, Spain. Association for Computational Linguistics.

Irene Kwok and Yuzhou Wang. 2013. Locate the Hate: Detecting Tweets Against Blacks. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2013)*, pages 1621–1622. AAAI Press.

Ching-Pei Lee and Chih-Jen Lin. 2014. Large-scale Linear RankSVM. *Neural Computation*, 26(4):781–817.

Lemur Project. 2020. RankLib. https://sourceforge.net/p/lemur/wiki/RankLib/. Accessed: Apr. 1, 2020.

LightGBM Doc. 2020. LightGBM Parameters Tuning. https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html. Accessed: Apr. 1, 2020.

Chih-Jen Lin. 2020. LIBLINEAR. https://www.csie.ntu.edu.tw/~cjlin/liblinear/. Accessed: Apr. 1, 2020.

Clare Llewellyn, Claire Grover, and Jon Oberlander. 2016. Improving Topic Model Clustering of Newspaper Comments for Summarisation. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 43–50. Association for Computational Linguistics.

Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2012. Topic-driven Reader Comments Summarization. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012)*, pages 265–274. ACM.

Microsoft. 2020. LightGBM. https://github.com/microsoft/LightGBM. Accessed: Apr. 1, 2020.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Wojciech Muła. 2020. pyachocorasick. https://pypi.org/project/pyahocorasick/. Accessed: Apr. 1, 2020.

Courtney Napoles, Aasish Pappu, and Joel R Tetreault. 2017a. Automatically Identifying Good Conversations Online (Yes, They Do Exist!). In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)*, pages 628–631. AAAI Press.

Courtney Napoles, Joel Tetreault, Aasish Pappu, Enrica Rosato, and Brian Provenzale. 2017b. Finding Good Conversations Online: The Yahoo News Annotated Comments Corpus. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 13–23. Association for Computational Linguistics.

Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. 2017. Exploring Generalization in Deep Learning. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5947–5956. Curran Associates, Inc.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive Language Detection in Online User Content. In *Proceedings of the 25th International Conference on World Wide Web (WWW 2016)*, pages 145–153. International World Wide Web Conferences Steering Committee.

Daisuke Okanohara and Jun'ichi Tsujii. 2009. Text Categorization with All Substring Features. In *Proceedings of the SIAM International Conference on Data Mining (SDM 2009)*, pages 838–846. SIAM.

Oxford. 2020. Definition of Constructive by Oxford Dictionary. https://www.lexico.com/definition/constructive. Accessed: Jun. 17, 2020.

Satoshi Sekine, Akio Kobayashi, and Kouta Nakayama. 2019. SHINRA: Structuring Wikipedia by Collaborative Contribution. In *Proceedings of the 1st International Conference on Automated Knowledge Base Construction (AKBC 2019)*.

Bei Shi and Wai Lam. 2018. Reader Comment Digest Through Latent Event Facets and News Specificity. *IEEE Transactions on Knowledge and Data Engineering*.

Stefan Siersdorfer, Sergiu Chelaru, Jose San Pedro, Ismail Sengor Altingovde, and Wolfgang Nejdl. 2014. Analyzing and Mining Comments and Comment Ratings on the Social Web. *ACM Transactions on the Web (TWEB)*, 8(3):17:1–17:39.

Nayer Wanas, Motaz El-Saban, Heba Ashour, and Waleed Ammar. 2008. Automatic Scoring of Online Discussion Posts. In *Proceedings of the Second ACM Workshop on Information Credibility on the Web*, pages 19–26. ACM.

Zhongyu Wei, Yang Liu, and Yi Li. 2016. Is This Post Persuasive? Ranking Argumentative Comments in Online Forum. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 195–200. Association for Computational Linguistics.

Yandex. 2020. Internet Mathematics 2009. https://academy.yandex.ru/events/data_analysis/grant2009. Accessed: Apr. 1, 2020.

A Details of Model Settings

The following list shows the detailed settings for the submitted models. Figure 4 shows the model structure of `Model-17`.

- Parameters of LambdaMART for `Model-4`: number of trees ('tree') = 1000, number of leaves for each tree ('leaf') = 10, learning rate ('shrinkage') = 0.1, number of threshold candidates for tree splitting ('tc') = 256, minimum number of samples each leaf has to contain ('mls') = 1, number of rounds for early stopping ('estop') = 100 (stopping early when no improvement is observed on the validation set over 100 rounds), and metric to optimize on the training data ('metric2t') = NDCG@100.
- Parameters of fastText for `Model-4` and `Model-11`: learning rate ('lr') = 0.1, update rate for the learning rate ('lrUpdateRate') = 100, dimension size of word embeddings ('dim') = 100, size of the context window ('ws') = 5, number of epochs ('epoch') = 5, number of negative

samples (‘neg’) = 5, and loss function (‘loss’) = ‘softmax’.

- Parameters of LightGBM for `Model-14`: boosting type (‘boosting_type’) = Gradient Boosting Decision Tree (Friedman, 2000) (‘gbdt’), objective function (‘objective’) = L2-loss (‘regression’), evaluation metric (‘metric’) = L2-loss (‘l2’), maximum number of leaves in one tree (‘num_leaves’) = 128, learning rate (‘learning_rate’) = 0.1, fraction to randomly select part of features on each iteration or tree (‘feature_fraction’) = 0.9, fraction to randomly select part of data without resampling (‘bagging_fraction’) = 0.8, frequency for bagging (‘bagging_freq’) = 5 (every 5 iterations), maximum number of bins that feature values are bucketed in (‘max_bin’) = 1000, number of iterations (‘num_iteration’) = 1000, and number of rounds for early stopping (‘early_stopping_rounds’) = 10 (stop if a validation metric does not improve in last 10 rounds).
- Feature construction for NDCG-L. The substrings were extracted by making a dictionary of maximal substrings (whose frequencies were more than 2) from all the comments by using a suffix tree-based extraction algorithm (Okanohara and Tsujii, 2009) with pykwic (ver. 0.1.5), a Python library (Aihara, 2020), and searching for maximal substrings in each comment by using the Eho-Chorasic dictionary-matching algorithm (Aho and Corasick, 1975) with pyachocorasick (ver. 1.4.0), another Python library (Muła, 2020).

B Details of Participant Survey

Table 5 shows the details of the participant survey (translated fromJapanese to English).

C Examples of Scored Comments

Table 6 shows examples of scored comments (translated into English) in the YJCCR dataset. Ex. 1 has a high score because it includes a constructive opinion with some reasoning. Ex. 2 has a middle score because the judgement, e.g., whether the comment is a new idea, depends on each worker’s background knowledge. Ex. 3 has a low score since it includes offensive content.

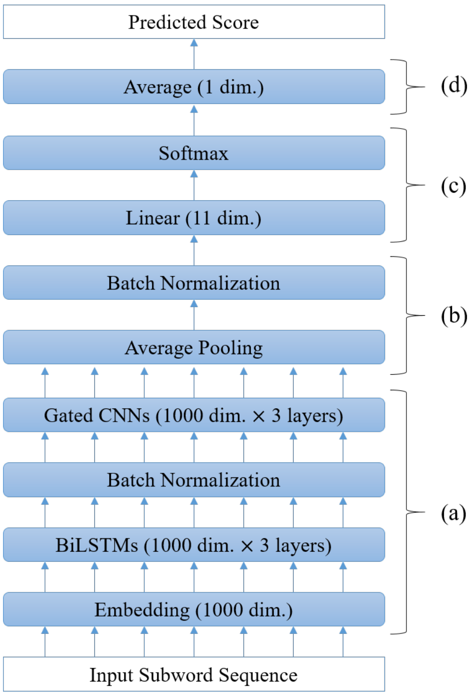


Figure 4: Structure of `Model-17`.

	Opinion
+	There were more participants than initially expected and a wide variety of models were submitted, so it turned out to be a good competition.
+	Since the participants disclosed their modeling methods, there were cases where one participant adopted the methods of other participants, which had a positive effect on improving the model’s performance.
+	Although I did not understand much about the work I was not in charge of, my participation in this competition deepened my understanding of the task and made it easier to participate in discussions during meetings.
+	I managed to learn a lot through trial and error in the competition.
–	It would be better to have a system that automatically evaluates predictions upon submission.
–	It would be better to not publicly disclose the test data.
–	When we were able to create a model with a high performance, we could not share detailed knowledge such as what kind of library was used, so it seems like there is room for improvement in the knowledge sharing system.
–	It would be good to have a system that rewards not only an increase in performance but also trying out new methods.

Table 5: Summary of the survey results (translated from Japanese to English).

Comment	Score
We should build a society where people do not drink and smoke since both can lead to bad health or accidents.	9
If we give freedom, punishment should also be strictly given.	6
They are irrational because they smoke, or they smoke because they are irrational.	0

Table 6: Examples of comments and scores for article “Lifting the ban on drinking and smoking at 18.”