Textual Content Moderation in C2C Marketplace

Yusuke Shido
Mercari, inc.
shidoy@mercari.com

Hsien-Chi Toby Liu *
tobbymailbox@gmail.com

Keisuke Umezawa
Mercari, inc.
k-umezawa@mercari.com

Abstract

Automatic monitoring systems for inappropriate user-generated messages have been found to be effective in reducing human operation costs in Consumer to Consumer (C2C) marketplace services, in which customers send messages directly to other customers. We propose a lightweight neural network that takes a conversation as input, which we deployed to a production service. Our results show that the system reduced the human operation costs to less than one-sixth compared to the conventional rule-based monitoring at Mercari.

1 Introduction

Mercari is a C2C marketplace app available in Japan and the United States. The application operates along the lines of an online flea market in which any customer can list items for sale and purchase others' listings. To prevent unpleasant customer experiences and unnecessary difficulties, Mercari defines some guidelines for using the platform. If a customer violates the guidelines, moderators from the customer support of Mercari may give them a warning to protect the safety and trustworthiness of the market.

To prevent abuse, platforms try to screen and monitor user-generated content. This is a human-intense task known as content moderation, and some companies like Appen, Facebook, and Pinterest introduced content-based deep learning approaches (Pavlopoulos et al., 2017a; Risch and Krestel, 2020a) to content platforms in recent years. Likewise, Mercari operates such moderation systems for several domains to maintain a safe and enjoyable marketplace.

We observed that most difficulties are encountered while sellers and buyers process with their transactions. In contrast to a normal B2C e-commerce pattern, sellers and buyers need to send messages to complete transactions, even though they are often both inexperienced customers. For example, customers might send messages to each other to provide specialized delivery information. Moderation of user messages is thus a crucial measure to prevent any forms of abuses, e.g. the transmission of offensive messages to others, inducing others to conduct an offline transaction, or even urging buyers to pay upfront via external services. These abuses are commonly detected by keyword-based monitoring or patrol (random inspections made by moderators of the customer support), which has been noted as being extremely limited in terms of scale and efficiency.

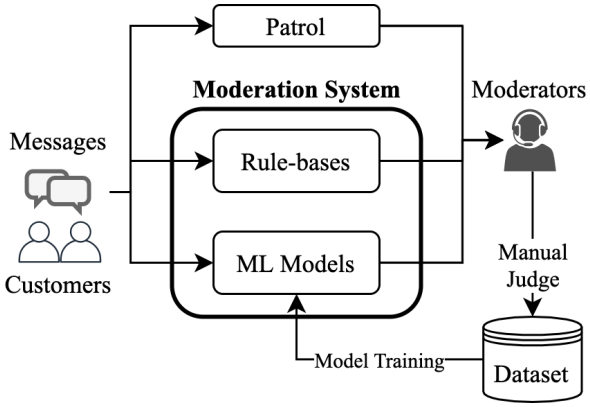

Figure 1: The overview of the moderation system for messages under transaction.

In this work, we introduce a scalable textual content moderation system based on machine learning models to perform the message moderation task. [1] Figure 1 shows an overview of our moderation system. In the proposed moderation system, messages with abusive intent or content that violate the terms and conditions could be reported from these three channels: patrol (random inspec-

*Work performed while at Mercari, inc.

[1] We analyzed messages between customers and introduced the machine learning-based system as fraud countermeasures in a legally permissible manner, with the consent and awareness of customers.

Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5), pages 58 - 62

tions), keyword/rule-based monitoring, and predictions made by machine learning (ML) models. In this work, we found that ML models successfully detected potentially-violated messages with a much higher precision compared to the conventional keyword/rule-based monitoring. However, because the type of violations may evolve over time, (e.g., a local government prohibiting the reselling of medical masks during COVID-19) updating ML-based methods in a timely manner following the fluctuations of the marketplace is notably difficult. We preserve the rule-based approach to accommodate the latest types of violations, so that moderators can still quickly revise the monitoring set of keywords and rules, which helps the system detect new domains of violations. In addition, a moderation system cannot tolerate false positives of any misjudged violation which might negatively effect the customer experience. All of the messages considered as potential violations by our moderation system are thus checked manually by moderators. This ensures that the customer experience on the marketplace is not compromised, and also creates a perfect human-in-the-loop cycle. To better detect offending messages and improve on the accuracy of existing ML models, we used these human-checked messages as an accurately-labelled dataset for model re-training.

The contributions of the present work are summarized as follows.

- A scalable machine learning-based violation detection system for content moderation of user-generated conversation is proposed.
- A resilient microservice architecture with high availability to serve violation domain models in an one-vs-rest manner was implemented.
- A human-in-the-loop framework was used successfully to reduce the workloads of moderators in customer support with ML-driven approaches.

2 Related Work

In contrast to typical B2C or B2B2C platforms, in which every item for sale can often be accessed with a single click, communication is inevitable on a C2C platform. Therefore, commercial content moderation plays a vital role in the process of communication to complete transactions successfully (Roberts, 2016).

Because manual content moderation does not scale to large platforms, it is natural to introduce automated content moderation systems. Pavlopoulos et al. (2017b); Risch and Krestel (2020b) investigated the applicability of machine learning-based approaches to the facilitation of content moderation by detecting textual violations using language models.

In the same context, to relieve human resources in the task at Mercari, Ueta et al. (2020) introduced a machine learning-driven item screening system that detects banned listings such as weapons, money, and medicine.

3 Method

3.1 Dataset

Some intents and behaviors, usually stated explicitly by service providers in terms and conditions, are prohibited in our marketplace to prevent unexpected difficulty between sellers and buyers. In this work, we focus on the messages freely exchanged between sellers and buyers until a transaction is completed.[2] The in-house dataset used in this work included the following three features. A **Message**, a text body sent by either the buyer or seller to the other. A positive example of a message that violates the terms and conditions is "Can we continue trading on this site to avoid fees?" This message is prohibited because the writer is trying to induce another customer to conduct the transaction with an external service, which is a violation of the terms of service. To comprehend the context of a conversation, we monitor the most recent five messages. The **Writer** is a label of either the seller or buyer, which indicates who wrote the message. Every message has one writer attribute. **Status** indicates the status of the transaction, e.g. "waiting for payment" or "waiting for shipping". This is an useful feature because exchanging contacts (e.g. SNS accounts) during the transaction have a higher chance to be made with abusive intentions, whereas it is usually normal to do so after the transaction has been completed.

During the creation of the dataset, we deliberately sampled positive and negative data with different sampling ratios, because the majority of messages exchanged on the platform do not vio-

[2]We use data from Japanese version of the Mercari app. We mask user's personal identifiable information, e.g. full names and addresses in the messages, at the pre-processing phase to use for analysis and the machine learning system.

late policy. We acquired raw data of one-year in-transaction conversations and divided them into the first 10 months, as the training set; the following month as the validation set; and the final month for the testing set so that the training results would be robust over time.

For a C2C marketplace like Mercari, we argue that a uniform language model for toxic speech may not fulfill our scenario in which violations may appear in a variety of forms. Violations also occur when users try to make deals offline by sending messages or by threatening other users to make payments upfront, in addition to hate speech. These intents and behaviors are considered as among the targets to be moderated. Considering the variety of violations as our targeted labels, we take the advantage of one-vs-rest classifiers to leverage this multi-label issue (Gunasekara and Nejadgholi, 2018). Thus, the proposed system comprises an assemblage cluster of sub-language models, and is designed to perform inference against a range of specific violation domains in a one-vs-rest fashion.

3.2 Metrics

As noted above, we had a quote for the number of alerts in actual use case. Therefore, we aimed to maximize the precision @K of the model, where K is the capable number of alert for moderators. We monitor the precision @K of running models to detect deterioration of model performance and concept drift (Zliobaite et al., 2016) in actual operation. However, K may vary due to various factors and the policy itself may change depending on the social situation, which causes concept drift. To simplify the experiment, we use the area under the receiver operating characteristic curve (AUROC) and average precision (a.k.a. AUPRC) to compare the models in this work. Also, the extent to which the ML model obviates the need for human labor is the important metric in our scenario. To monitor this, we measured the number of alerts from the ML model required to detect the same number of positives as the rule-base.

3.3 Model Architecture

Our neural network model shown in figure 2 receives three features described in 3.1 and outputs the probability of a violated message. The model extracts textual features using a convolutional neural network (CNN) and flattens the grasped sentence-level features into a 1D feature representation as an input to the subsequent recurrent neural network (RNN). First, we parse the "Message" input using the fast Japanese morphological analyzer MeCab (Kudo, 2005) with the large dictionary (Toshinori, 2015). We transform the messages to sequences of numeric tokens, and embed them to matrices $\boldsymbol{x^{(M)}} = \left(x_1^{(M)}, \ldots, x_{N_d}^{(M)}\right)$, where $x_i^{(M)} \in \mathbb{R}^{N_s \times d_e^{(M)}}$ is a sequence of word vectors of i-th sentence, N_d, N_s, and d_e are the number of sentences, the length of sentences, and the dimension of embedding space, respectively. We use the word vectors pretrained with word2vec (Mikolov et al., 2013) to perform embedding and optimize the word vectors in the same manner as other model parameters during training. We adopt the convolutional neural network proposed in Kim (2014) (a.k.a. KimCNN) for textual feature extraction because it is lightweight and can reach relatively good accuracy even with a small number of parameters. The d-dimensional textual feature vectors $f_i^{(M)} \in \mathbb{R}^d$ is computed from each $x_i^{(M)}$ using a single KimCNN. The "Writer" feature of i-th sentence and the single "Status" feature are also embedded as d-dimensional vectors $f_i^{(W)}$ and $f^{(S)}$, respectively. The feature of i-th sentence is computed as the sum of the "Message" feature and "Writer" features as $f_i^{(M+W)} = f_i^{(M)} + f_i^{(W)}$. The gated recurrent units (GRUs) (Ballakur and Arya, 2020) compute the feature vector representing the entire conversation $g^{(M+W)} \in \mathbb{R}^d$ from sentence features $\boldsymbol{f^{(M+W)}} = \left(f_1^{(M+W)}, \ldots, f_{N_d}^{(M+W)}\right)$. The GRU is expected to understand the conversation as a sequence of sentences. We use two GRUs, one to enter in forward order and another to enter in reverse order. Finally, the model output $p \in [0, 1]$ is computed from combined feature vector $f^{(M+W+S)} = g^{(M+W)} + f^{(S)}$ using a multi-layer perceptron (MLP). The model parameters

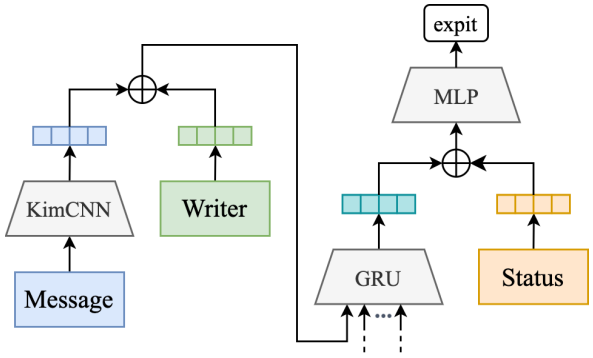

Figure 2: Neural network model architecture.

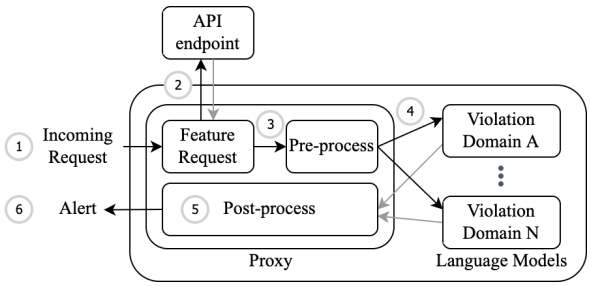

Figure 3: Core components of our ML moderation system.

including embedding vectors are trained by minimizing entropy as $-y \log p - (1-y) \log (1-p)$, where $y \in (0, 1)$ is the true label for the prediction p.

3.4 System Deployment

As one of the largest online marketplace in Japan, Mercari deals with thousands of transaction messages every minute. The system must be robust with high short term capacity to accommodate a variety of language models for each violation domain. Also, it must be easy to add new models and update existing models to keep up with the market situation. Hence, we trained machine learning models in one-vs-rest manner for easy model addition and update, and designed the whole system to serve the models in a horizontal asynchronized pattern (Yusuke, 2020) as shown in Figure 3 to overcome peak traffic. The proxy and each language model components are able to be scaled separately.

Starting from an incoming request sent to the proxy component with a specific message id, the system sends a feature request to the internal API endpoint to fetch the dialogue content as a second step. The proxy component generates feature representations by doing preprocessing over sentences in a given dialogue and sends them to each language model as Steps three and four. Predictions against each violation domain are inferred and returned by the components serving language models as a proxy component again to perform post-processing in Step five. In that step, we send alerts to moderators if the inferred probability surpasses thresholds pre-defined to control the amount of alerts for each violation domain. We also record prediction results for the later human-in-the-loop evaluation.

3.5 Experiments

We conducted experiments to examine whether the data enrichment described in 3.1 improved the performance of the proposed model. Table 1 shows the result of our experiments. We can confirm that using multi-round messages and their additional metadata was effective for the detection of violating messages. In the experiments, we adopted the AdaBound optimizer with a learning rate of 0.001 and decaying learning rate according to the cosine curve. We trained the model for 10 epochs and adopted the weights at epoch, which yielded the best performance on the validation loss. The final performance was evaluated with the testing set.

#M	W	S	AUROC	AUPRC
1			99.4068	97.1052
5			99.6919	98.3702
5		✓	99.6996	98.3793
5	✓		99.6947	98.3733
5	✓	✓	**99.7274**	**98.4194**

Table 1: AUROC and AUPRC score with various features. Here, #M represents the number of message to input to the model and the check mark in column W or S indicates that Writer or Status feature was used. We trained the model three times with different initial weights for each, and adopted average value for the model performance.

In online evaluation, we observe that our model was able to find the same number of positive messages as existing rule-base search methods with $\mathbf{16.05}\%$ of the number of alerts from the rule-base in this violation domain. This means that by replacing the rule-based method with the ML model, the human resources requirements of the system can theoretically be reduced by more than half. However, we preserved rule-bases for the reasons mentioned in Section 1.

4 Conclusion

In this work, we have proposed a scalable ML-based violation detection system designed to accept multi-round user conversations and some categorical features. A variety of violation domains are served as individual components in an one-vs-rest fashion to retain scalability to high volumes of requests and flexibility on targeting domains. In comparison with conventional rule-based monitoring, we have demonstrated that the proposed ML-driven approach successfully reduced the workloads of moderators in customer support to less than one-sixth of their previous levels by automatically detecting abusive messages.

References

Appen. Leveraging ai and machine learning for content moderation [online].

Amulya Arun Ballakur and Arti Arya. 2020. Empirical evaluation of gated recurrent neural network architectures in aviation delay prediction. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pages 1–7. IEEE.

Facebook. How we review content [online].

Isuru Gunasekara and Isar Nejadgholi. 2018. A review of standard text classification practices for multi-label toxicity identification of online content. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*, pages 21–25.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

T. Kudo. 2005. Mecab : Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.net/*.

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*, volume 26. Curran Associates, Inc.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017a. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1125–1135.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017b. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1125–1135.

Pinterest. Getting better at helping people feel better [online].

Julian Risch and Ralf Krestel. 2020a. Toxic comment detection in online discussions. In *Deep Learning-Based Approaches for Sentiment Analysis*, pages 85–109. Springer.

Julian Risch and Ralf Krestel. 2020b. Toxic comment detection in online discussions. In *Deep Learning-Based Approaches for Sentiment Analysis*, pages 85–109. Springer.

Sarah T Roberts. 2016. Commercial content moderation: Digital laborers' dirty work.

Sato Toshinori. 2015. Neologism dictionary based on the language resources on the web for mecab.

Shunya Ueta, Suganprabu Nagaraja, and Mizuki Sango. 2020. Auto content moderation in c2c e-commerce. In *2020 USENIX Conference on Operational Machine Learning (OpML 20)*. USENIX Association.

Shibui Yusuke. 2020. Machine learning system design pattern.

Indre Zliobaite, Mykola Pechenizkiy, and Joao Gama. 2016. *An Overview of Concept Drift Applications*, Studies in Big Data, pages 91–114. Springer International Publishing AG, Switzerland.