Dataset Creation for Ranking Constructive News Comments

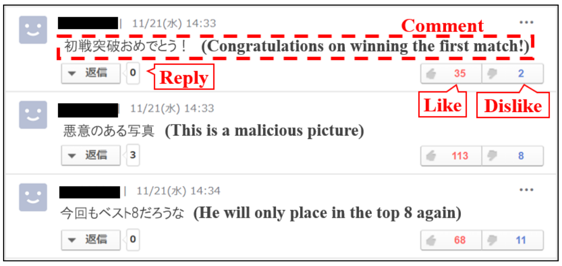
Soichiro Fujita,[†] **Hayato Kobayashi,**[‡] **and Manabu Okumura**[†]
[†] Tokyo Institute of Technology
[‡] Yahoo Japan Corporation / RIKEN AIP
{fujiso@lr.,oku@}pi.titech.ac.jp, hakobaya@yahoo-corp.jp

Abstract

Ranking comments on an online news service is a practically important task for the service provider, and thus there have been many studies on this task. However, most of them considered users' positive feedback, such as "Like"-button clicks, as a quality measure. In this paper, we address directly evaluating the quality of comments on the basis of "constructiveness," separately from user feedback. To this end, we create a new dataset including 100K+ Japanese comments with constructiveness scores (C-scores). Our experiments clarify (a) C-scores are not always related to users' positive feedback and (b) the performance of pairwise ranking models tends to be more enhanced by the variation in comments than that in articles.

1 Introduction

Users' comments on an online news service can be regarded as beneficial content (often called *user-generated content*[1]) for service providers because users can obtain supplementary information about news articles through other users' opinions. Given that comment visibility is a part of the user experience, ranking comments is practically important. For example, Figure 1 shows a page displaying comments on a Japanese news portal, Yahoo! News.[2] The page has a list of comments (displayed below articles), and each comment has buttons for user feedback ("Like," "Dislike," and "Reply").


[redacted] | 11/21(水) 14:33
初戦突破おめでとう！ (Congratulations on winning the first match!)

[redacted] | 11/21(水) 14:33
悪意のある写真 (This is a malicious picture)

[redacted] | 11/21(水) 14:34
今回もベスト8だろうな (He will only place in the top 8 again)

Figure 1: Examples of comments on Yahoo! News.

There have been many comment ranking studies (Hsu et al., 2009; Das Sarma et al., 2010; Brand and Van Der Merwe, 2014; Wei et al., 2016) with users' positive feedback for a comment (e.g., "Like"- or "Upvote"-button clicks) serving as the quality measure. However, this type of measurement has two drawbacks: (a) user feedback does not always satisfy the service provider's needs, such as to create a fair place, and (b) user feedback will be biased by where comments appear in a comment thread. A typical situation for (a) can be seen in political comments, where the "goodness" of the comment will be decided on the basis of the political views of the majority of the users rather than its quality. The situation for (b) can be illustrated by a case where earlier comments tend to receive more feedback since they will be displayed at the top of the page, which implies later comments will be ignored irrespective of their quality.

In this paper, we directly evaluate the quality of comments separately from user feedback, focusing on their "constructiveness," as studied in (Napoles et al., 2017; Kolhatkar and Taboada, 2017). This quality measure is reasonable for services in that displaying constructive comments can stimulate discussion on a news article, which makes the user-generated content richer. We use the definition of constructiveness as in the previous studies, but a clear difference from them is that we address a ranking task, whereas the aforementioned sources addressed classification tasks. In a ranking task, we need to rank comments for each article. That is, when we label 1,000 comments, there are many choices, e.g., 200 articles with 5

[1] https://en.wikipedia.org/wiki/User-generated_content

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

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2619–2626
Florence, Italy, July 28 - August 2, 2019.

comments or 10 articles with 100 comments. We investigate which choice is better for widely used ranking algorithms.

Our contributions are as follows.

- We create a dataset for ranking constructive comments including 100K+ Japanese comments with constructiveness scores, in collaboration with Yahoo! News. Our dataset will be publicly available.[3]
- We show empirical evidence that constructiveness scores are not always related to positive user feedback such as "Like"-button clicks.
- We investigate how to label comments for ranking and clarify that the performance of pairwise ranking models tends to be more enhanced by the variation in comments than that in articles.

2 Dataset Creation

2.1 Definition for "Constructiveness"

According to the dictionary,[4] "constructive" means "*having or intended to have a useful or beneficial purpose.*" Therefore, we expect constructive comments to provide insight and encourage healthy discussion. However, this dictionary definition is a bit too generic for deciding if a comment is constructive. To avoid individual variation as much as possible, we need to prepare a more specific definition before annotation. We follow a previous study (Kolhatkar and Taboada, 2017) on constructiveness, where a questionnaire given to 100 people clarified detailed conditions for constructive comments. We digested it into several simple conditions, shown in Table 1, so that crowdsourced workers could systematically judge comments. Our conditions consist of a 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 satisfying the precondition and at least one of the main condition in Table 1.

2.2 Crowdsourcing Task

Our purpose is to label each comment with a graded numeric score that represents the level of constructiveness for ranking comments. We refer to this score as the **constructiveness score (C-score)**. We defined the C-score as the number of crowdsourcing workers who judged a comment to be constructive as an answer to a yes-or-no (binary) question because it is more difficult for workers to answer other types of questions such as a numerical selection question (like "How constructive is the comment?") or a comparison question (like "Which comment is the most constructive?"). This definition realizes a graded numeric score that harnesses the individual variation due to subjective judgements in the conditions, such as "new idea" and "rare experience." 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.

We used Yahoo! Crowdsourcing[5] to label comments. We prepared a task with questions that reference a news article and its comments extracted from Yahoo! News. After the workers read the definition of constructiveness, we asked them to judge whether each comment was constructive (see Appendix A for detailed instructions). To ensure reliability, we extracted only serious workers who correctly answered quality control questions with obvious answers that were randomly included in each task. We used 10 (or 40) workers for each comment for a training (test) dataset. For example, a C-score of 8 means that 8 workers judged a comment as constructive.

Pre cond.	• Related to article and not slander
Main cond.	• Intent to cause discussions • Objective and supported by fact • New idea, solution, or insight • User's rare experience

Table 1: Conditions for constructive comments. Constructive comment is defined as one satisfying the precondition and at least one of main conditions.

	#A	#C	#C/#A	Score
Shallow	8,000	40,000	5	0 ∼ 10
Deep	400	40,000	100	0 ∼ 10
Test	200	42,436	212	0 ∼ 40

Table 2: Details on created datasets. #A and #C mean numbers of articles and comments in each dataset, respectively.

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

[4] https://en.oxforddictionaries.com/definition/constructive

[5] https://crowdsourcing.yahoo.co.jp/

Comment	Score
Ex.1) We should build a society where people do not drink and smoke since both can lead to bad health or accidents.	9
Ex.2) If giving freedom, punishment should also be strictly given.	6
Ex.3) They are fools because they smoke, or they smoke because they are fools.	0

Table 3: Examples of comments and scores for article "Lifting the ban on drinking and smoking at 18."

2.3 Training and Test Datasets

We created three datasets: Shallow, Deep, and Test, as shown in Table 2. Shallow and Deep are training datasets made from 8K articles with 5 comments and 400 articles with 100 comments respectively, as extreme cases with the same cost. The comments in each setting were randomly chosen after we extracted news articles with more than 100 comments and were 10 to 125 Japanese characters long. Test is the test dataset we made from 200 articles with an average of 212 comments. We used 40 workers for each comment only for Test to evaluate the ranking results in as much detail as possible, where the setting of 40 was chosen to avoid the top-ranked comments that frequently had the same score. Note that we did not use such a costly setting for training since training data tends to increase over time. None of the datasets overlapped.

We calculated an agreement score by using Krippendorff's alpha (Krippendorff, 2004; Antoine et al., 2014) and by regarding the ranking task as a classification task of whether one comment is more constructive than the other for any pair of two comments, in a similar manner as RankSVM in Section 3. The agreement scores of Shallow and Deep were 0.5282 and 0.5495, respectively, which mean "moderate agreement" (Landis and Koch, 1977). Note that directly applying such an agreement measure is not appropriate for our task since we assume individual variations in workers making graded scores.

Table 3 shows examples of scored comments. Ex. (1) has a high score since it includes a constructive opinion with some reasoning. Ex. (2) has a middle score since 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.

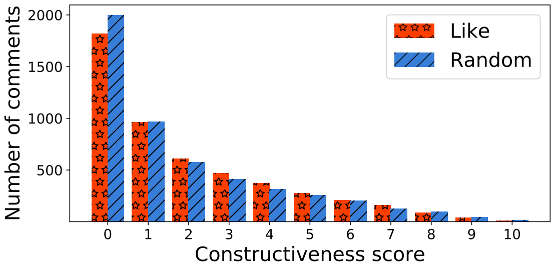


Figure 2: Frequency distribution of C-scores for comment group selected in descending order of user feedback (Like) and one randomly selected (Random).

2.4 Comparison with User Feedback

We investigated the relationship between constructiveness and user feedback by comparing 5K comments randomly extracted in the same way as for Shallow and 5K comments extracted in descending order of user feedback score. The user feedback score of a comment was calculated as the number of "Likes" minus 5 times the number of "Dislikes." This definition is determined on the basis of the fact that the ratio of "Likes" and "Dislikes" was about 1:5 on average, and in fact, a similar definition is used as a basic sorting feature in this news service. All of the comments in the above two groups were labeled with C-scores in the same way as for Shallow/Deep.

Figure 2 shows the frequency distributions of the two groups over C-scores. Surprisingly, both distributions form almost the same shape even though we expected that the comments ordered with the user feedback would have high C-scores. In fact, the correlation coefficient between the user feedback scores and the C-scores was nearly zero, i.e., -0.0036. This means that constructiveness is completely different from user feedback, and using user feedback is not a promising way to show constructive comments in the service.

3 Ranking Constructive News Comments

3.1 Compared Methods

We compared the following methods for understanding the characteristics of our datasets. Here, we selected simple SVM-based methods since we can easily interpret the results, although we included the results of neural ranking models in Appendix B.

- Like ranks with the user feedback score.
- Random ranks randomly.
- Length ranks in descending order on the basis

of the comment length.

- RankSVM ranks via a rankSVM model (Lee and Lin, 2014) trained to infer relative constructiveness between two comments. Roughly speaking, we solve a binary classification problem of whether or not a comment is more constructive than another one, like SVM.
- SVR ranks via a support vector regression model (Vapnik et al., 1997) trained to directly infer the C-score.

We used liblinear-ranksvm[6] for RankSVM and SVR. The cost parameter was determined from $\{2^0, \ldots, 2^{-13}\}$ with a validation dataset, where we prepared another 5K comments for each setting for Shallow/Deep. The features for training RankSVM and SVR were made from a comment and the corresponding article. See the next section for the details on preprocessing and the features.

3.2 Preprocessing and Features

The preprocessing for training RankSVM and SVR is as follows. We used a morphological analyzer MeCab[7] (Kudo et al., 2004), with a neologism dictionary, NEologd[8] (Toshinori Sato and Okumura, 2017), for splitting Japanese text into words. We replaced numbers with a special token and standardized letter types, i.e., decapitalization and halfwidth-to-fullwidth.[9] We did not remove stop-words because function words would affect the performance in our task, especially for decency. We cut low-frequency words off that appeared only three times or less in each dataset. The dictionary size was about 50,000.

The features for a comment (with the corresponding news article) used for RankSVM and SVR are the bag-of-words of the comment, the number of unique words in the comment, the cosine similarity (based on bag-of-words vectors) between the comment and the title, and the bag-of-words co-occurring in the comment and the title, which are distinguished from the normal bag-of-words. Note that we used only titles for features to avoid extra labeling and training costs for lengthy article bodies, assuming that a title can be regarded as a summary of the corresponding article.

3.3 Evaluation

We used normalized discounted cumulative gain (NDCG) (Burges et al., 2005a) as our primary evaluation measure, which is widely used for evaluating ranking models in information retrieval tasks. The NDCG is typically calculated for the top-k comments ranked by a ranking model and denoted by NDCG@$k = Z_k \sum_{i=1}^{k} \frac{r_i}{\log_2(i+1)}$, where r_i represents the true C-score of the i-th ranked comment, and Z_k is a normalization constant to scale the value between 0 and 1. This equation means that the value becomes higher (better) as the inferred ranking becomes closer to the correct ranking, especially for top ranked comments. In addition, we used precision@k as our secondary evaluation measure, which is defined as the ratio of correctly included comments in the inferred top-k comments with respect to the true top-k comments. Note that a well-known paper (Järvelin and Kekäläinen, 2002) in the information retrieval field determined NDCG to be more appropriate than precision for graded scores like our setting.

3.4 Results

Table 4 shows the results of NDCG@k and precision@k (for $k \in \{1, 5, 10\}$) for Test for the compared models, where RankSVM and SVR have two variations trained with Shallow and Deep. Random was averaged over 10 trials. Note that all values are represented as percentages.

The results of Like and Random show that neither of them performed well, which is consistent with our finding that Like has a similar tendency to Random, as described in Section 2. However, Length performed better than Like and Random. This implies that long comments tend to be constructive, but of course, the length of comments is not enough to accurately infer the C-score, compared with RankSVM.

Among all variations of RankSVM and SVR, RankSVM with Deep consistently performed the best for our primary evaluation measure NDCG. The differences between NDCGs of RankSVM with Deep and SVR with Shallow were statistically significant in a paired t-test ($p < 0.05$). As for precision, it was beaten by SVR with Shallow for @1 and @5. This means that RankSVM sometimes failed to find the best solutions (the most constructive comment) but obtained better solutions (fairly constructive ones).

[6] https://github.com/FurongPeng/liblinear-ranksvm

[7] http://taku910.github.io/mecab/

[8] https://github.com/neologd/mecab-ipadic-neologd

[9] https://en.wikipedia.org/wiki/Halfwidth_and_fullwidth_forms

	Dataset	NDCG@1	NDCG@5	NDCG@10	Prec@1	Prec@5	Prec@10
`Like`	-	29.93	31.84	34.99	2.00	6.20	8.70
`Random`	-	25.85	27.90	29.06	1.10	4.60	6.50
`Length`	-	60.28	64.93	67.72	6.00	20.80	30.04
`RankSVM`	`Shallow`	72.24	74.63	76.79	14.50	29.40	41.24
`RankSVM`	`Deep`	**74.15**	**76.44**	**78.25**	13.00	31.60	**42.20**
`SVR`	`Shallow`	73.87	75.48	76.97	**16.50**	**32.70**	41.00
`SVR`	`Deep`	69.68	71.99	74.26	11.00	27.20	36.35

Table 4: Results (%) of NDCG@k and precision@k for task of ranking constructive comments.

Comparing `Shallow` and `Deep` for `RankSVM`, we can see that `RankSVM` performed better with `Deep` than with `Shallow` because the number of training examples for pairwise ranking models was 2-combinations from n, i.e., $\binom{n}{2} = \frac{n(n-1)}{2}$, given n comments. This means that the number of pairwise examples increases in $O(n^2)$. Conversely, `SVR` performed well with `Shallow`. Features based on articles can be useful for directly inferring the C-scores without comparing comments in such cases. Similar findings were observed in the results of neural ranking models (see Appendix B), but we omitted them because of space limitations.

4 Related Work

Analyzing comments on online news services or discussion forums has been extensively studied (Wanas et al., 2008; Ma et al., 2012; Brand and Van Der Merwe, 2014; Llewellyn et al., 2016; Shi and Lam, 2018). In this line of research, there have been many studies on ranking comments (Hsu et al., 2009; Das Sarma et al., 2010; Brand and Van Der Merwe, 2014; Wei et al., 2016). However, their approaches were based on user feedback, which is completely different from constructiveness, as explained in Section 2.

Constructiveness has sometimes been introduced in argument analysis frameworks. Napoles et al. (2017) created a dataset for argument analysis on the basis of reply threads, each of which has a label as a constructiveness flag and consists of child comments replying to the parent comment. Kolhatkar and Taboada (2017) proposed a classification model that determines constructiveness for a comment by regarding all comments in a constructive thread as constructive and evaluated it with a dataset of 1K manually annotated comments, which is much smaller than our datasets. Our task is a ranking task based on graded numeric scores and different from their task. If training a regression model with binary labels, the results will be similar to `SVR`.

There are mainly two approaches to analyzing the quality of comments on the basis of their content without using constructiveness. One is hate speech detection (Kwok and Wang, 2013; Nobata et al., 2016; Davidson et al., 2017) and the other is sentiment analysis (Fan and Sun, 2010; Siersdorfer et al., 2014). Although these approaches are useful for other tasks, they do not directly solve our task, i.e., 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.

Learning-to-rank methods are often used for information retrieval tasks (Liu, 2009). There are several datasets for ranking documents on search engines, such as Microsoft LETOR (Qin et al., 2010; Qin and Liu, 2013) and Yahoo! LTRC (Chapelle and Chang, 2011). Because it is not feasible to label all documents for each query, “possibly” relevant documents are typically sampled by using a simple ranking algorithm such as BM25 (Robertson and Zaragoza, 2009). However, we cannot use such a strategy since comments are basically relevant to an article, and there are many relevant but non-constructive comments.

5 Conclusion

We created a new labeled dataset for ranking constructive comments. Experimental results suggested that pairwise ranking models work well with the variation of comments rather than articles. Our future work will include efficiently labeling promising comments via active learning.

Acknowledgements

We would like to thank anonymous reviewers for their constructive comments.

References

Jean-Yves Antoine, Jeanne Villaneau, and Anaïs Lefeuvre. 2014. Weighted Krippendorff's alpha is a more reliable metrics for multi-coders ordinal annotations: experimental studies on emotion, opinion and coreference annotation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 550–559. Association for Computational Linguistics.

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.

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

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

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.

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.

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.

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

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology (second edition)*. Sage Publications.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 230–237. 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.

J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.

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

Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

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.

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 (NIPS 2013)*, pages 3111–3119. Curran Associates, Inc.

Courtney Napoles, Aasish Pappu, and Joel R Tetreault. 2017. 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.

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.

Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR*, abs/1306.2597.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. 2010. LETOR: A Benchmark Collection for Research on Learning to Rank for Information Retrieval. *Information Retrieval*, 13(4):346–374.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

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.

Taiichi Hashimoto Toshinori Sato and Manabu Okumura. 2017. Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval (in japanese). In *Proceedings of the Twenty-three Annual Meeting of the Association for Natural Language Processing*, pages NLP2017–B6–1. The Association for Natural Language Processing.

Vladimir Vapnik, Steven E. Golowich, and Alex J. Smola. 1997. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. In *Advances in Neural Information Processing Systems 9 (NIPS 1997)*, pages 281–287. MIT Press.

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.

A Details on Instructions for Crowdsourced Workers

Detailed instructions (translated in English) on our crowdsourcing task are as follows. We included five comments of the same article in each task to reduce workers' annotation cost.

> Instruction: Given five comments for an article, please select all comments that satisfy the following precondition and at least one main condition.
>
> - Pre-condition: The comment is related to the article and is not an unpleasant one, including slander.
>
> - Main-condition 1: The comment intends to cause discussions on the basis of the author's opinion.
>
> - Main-condition 2: The comment is objective and supported by fact or reason.
>
> - Main-condition 3: The comment gives a new idea, solution, or insight.
>
> - Main-condition 4: The comment is a user's rare experience related to the article.

B Results of Neural Models

We confirmed that the results of neural models have a similar tendency to those of SVM-based models, although we omitted these results due to space limitations. We compared a neural pairwise ranking model, `RankNet`, and a neural regression model, `LSTMReg`, as follows.

- `RankNet` ranks via a neural pairwise ranking model, RankNet (Burges et al., 2005b). The key concept of this model is similar to that of `RankSVM`, i.e., solving the ranking problem as a classification problem of whether a comment is more constructive than another one. Specifically, the model is constructed to predict the ranking score of a comment and trained so that, given two comments, the magnitude relation of the predicted scores corresponds to that of the true constructiveness scores, via cross entropy loss.
- `LSTMReg` ranks via an LSTM-based regression model. The basic structure is the same as `RankNet`, but the training is performed so that, given a comment, the predicted score corresponds to the true constructiveness score, via mean squared error loss.

The experimental settings were as follows. The preprocessing was the same as in `RankSVM`, except that cutoff tokens were replaced with a special token "`<unk>`". We used 300 dimensional embeddings of a skip-gram model (Mikolov

	Dataset	NDCG@1	NDCG@5	NDCG@10	Prec@1	Prec@5	Prec@10
RankNet	Shallow	73.42	73.91	75.11	**13.67**	27.40	37.81
RankNet	Deep	**75.19**	**77.17**	**78.62**	13.17	**31.72**	**41.68**
LSTMReg	Shallow	71.71	73.96	75.74	12.68	28.48	38.99
LSTMReg	Deep	69.40	72.51	74.21	10.55	26.75	36.28

Table 5: Results (%) of NDCG@k and precision@k for task of ranking constructive comments for RankNet and LSTMReg.

et al., 2013) trained with 1.5 million unlabeled news comments by using an open source software, gensim,[10] with the default parameters. Both RankNet and LSTMReg had the same structure, i.e., an encoder-scorer. The encoder consisted of two LSTMs with 300 units to separately encode a comment and its title, and the scorer predicted the ranking score of the comment via a full-connected layer after concatenating the two encoded (comment and title) vectors. We used the Adam optimizer ($\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$) to train these models. The batch size was 10 (pairs sampled from each article when training RankNet), and the number of iterations of batches was 10,000.

The formal definition of the loss function of RankNet is the same as in the original paper. Given two comments c_1 and c_2, we define the probability of c_1 being more constructive than c_2 as $p = \sigma(f(c_1) - f(c_2))$, where $\sigma(\cdot)$ is a sigmoid function, and $f(c)$ is the predicted score of c. The cross entropy loss is calculated as $-\overline{p} \log p - (1 - \overline{p}) \log(1 - p)$, where $\overline{p}$ is 1 if the true constructive score of c_1 is higher than that of c_2, 0 if lower, and 0.5 if otherwise.

Figure 5 shows the results of RankNet and LSTMReg. Looking at our primary measure NDCG, we can see that RankNet with Deep clearly performed the best. Furthermore, comparing the results with Shallow and Deep, RankNet with Deep performed better than RankNet with Shallow, while LSTMReg with Shallow performed better than LSTMReg with Deep. These findings are consistent with the results of SVM-based models.

[10] https://radimrehurek.com/gensim/