Unsupervised Ensemble of Ranking Models for News Comments Using Pseudo Answers

Soichiro Fujita[1], Hayato Kobayashi[2], and Manabu Okumura[1]

[1] Tokyo Institute of Technology, {fujiso@lr.,oku@}pi.titech.ac.jp
[2] Yahoo Japan Corporation / RIKEN AIP, hakobaya@yahoo-corp.jp

Abstract. Ranking comments on an online news service is a practically important task, and thus there have been many studies on this task. Although ensemble techniques are widely known to improve the performance of models, there is little types of research on ensemble neural-ranking models. In this paper, we investigate how to improve the performance on the comment-ranking task by using unsupervised ensemble methods. We propose a new hybrid method composed of an output selection method and a typical averaging method. Our method uses a pseudo answer represented by the average of multiple model outputs. The pseudo answer is used to evaluate multiple model outputs via ranking evaluation metrics, and the results are used to select and weight the models. Experimental results on the comment-ranking task show that our proposed method outperforms several ensemble baselines, including supervised one.

1 Introduction

User comments on online news services can be regarded as a useful content since users can read other users' opinions related to each news article. Many online news service sites rank comments in the order of the number of positive user-feedback for a comment, such as "Like"-button clicks, and preferentially display popular comments to readers. However, this type of user-feedback is not suitable to assess the comment quality, because this type of measurement is biased by where a comment appears [7]; Earlier comments tend to receive more feedback since they will be displayed at the top of the page. In attempt of solving this problem, several studies introduce some aspects of the comment quality to focus on, e.g., constructiveness [13, 7] or persuasiveness [22]. In particular, Fujita et al. [7] proposed a new dataset to rank comments directly according to comment quality. This is a difficult task because we have various situations of judging whether a comment is good. For example, comments can indicate rare user experiences, provide new ideas, or cause discussions. Ranking models often fail to capture such information.

According to recent studies [15, 2, 12], ensemble techniques are widely known to improve the accuracy of machine learning models. These ensemble techniques can be roughly divided into two types: averaging and selecting. Averaging methods such as Naftaly et al. [17] simply average multiple model outputs. Selecting methods such as majority vote [15] select the most frequent label from the predicted labels of multiple classifiers in post-processing. These methods assist models to make up for other models' mistakes and to improve the results. Recently, Kobayashi [12] proposed an unsupervised ensemble method, post-ensemble, based on kernel density estimation, which was an extension of the majority vote to text generation models. He showed that this method outperformed averaging methods in a text summarization task.

In this paper, we propose a new unsupervised ensemble method, HPA, which is a hybrid of an output selection and a typical averaging method. In typical averaging methods, a lower accuracy model could merely be noise. A simple denoising method is to statically remove such lower accuracy models [19]. However, there is basically no model that fails for every inputs, particularly in neural models with the same architecture. In general, each model has its own strengths and weaknesses. Therefore, our method adopts dynamic denoising of outputs via a provisional averaging result. We use the provisional averaging result as a pseudo answer. Each predicted ranking is compared to the pseudo answer via a similarity function, and the similarity scores are used for selecting and weighting models. We adopt evaluation metrics as a kind of similarity to specialize in the ranking task. In experiments on a task of ranking constructive news comments, our proposed method HPA outperformed both previous unsupervised ensemble methods and a simple supervised ensemble method. Furthermore, we found that one of the evaluation metrics is useful as a similarity measure for the ensemble process.

2 Proposed method

2.1 Problem Statement

Comment Ranking Task: Let an article be associated with comments $C = (c_1, ..., c_n)$. Each comment has a manually annotated score $S = (s_1, ..., s_n)$, such as the degree of comment quality. A ranking model m learns a scoring function $\tilde{s_i} = m(c_i)$. We consider a predicted score sequence as a ranking of the comments $r = (\tilde{s_1}, ..., \tilde{s_n})$, because we can generate a ranked comment sequence using this score sequence.
Ensemble Problem: We prepare N rankings $R = (r_1, ..., r_N)$ from ranking models $M = (m_1, ..., m_N)$. The goal of the ensemble is to combine the ranking models to produce a better ranking than any of the individual ranking functions. A simple averaging method calculates the average of the comment scores, like $r^* = \sum_{r \in R} \frac{r}{|R|}$.

2.2 Post-Ensemble

We introduce PostNDCG which applies the post-ensemble method [12] to the ranking task. Post-ensemble is an unsupervised ensemble method based on kernel density estimation for sequence generation. This method compares the similarity between model outputs and selects the majority-like output which is similar to the other outputs. This selection is equivalent to selecting the output whose estimated density is the highest in the outputs. PostNDCG calculates this scoring function: $f(r) = \frac{1}{|R|}\sum_{r' \in R} sim(r, r')$, where $sim(r, r')$ represents the similarity between r and r'. The final ranking of PostNDCG is defined as $r^* = \mathrm{argmax}_{r \in R} f(r)$. We used the normalized discounted cumulative gain (NDCG@k) [1] as the similarity function $sim(\cdot)$ to compare each ranker.

2.3 HPA Ensemble

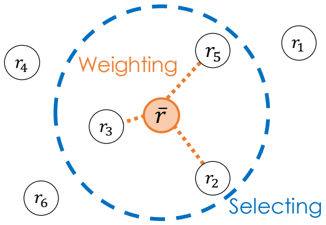

Fig. 1. Example of HPA.

We propose a **H**ybrid method using the **P**seudo **A**nswer (HPA). Fig. 1 illustrates an example of HPA. Here, HPA selects the top three rankings $\{r_2, r_3, r_5\}$ that are nearest to the pseudo answer. After that, it weights each selected ranking via a scoring function based on the pseudo answer. The concept of HPA is to denoise outputs via a pseudo answer $\bar{r}$, which is represented by the average of each model output after the L2 normalization: $\bar{r} = \frac{1}{|R|}\sum_{r \in R} \frac{r}{||r||}$. The scoring function g is calculated as the similarity between the pseudo answer and the predicted ranking: $g(r) = sim(\bar{r}, r)$. Then, HPA selects the top k models with the highest scores. The final ranking r^* is represented as, $r^* = \sum_{r \in \bar{R}} g(r) \cdot r$, where $\bar{R}$ is the set of selected models (rankings).

3 Experiments

3.1 Experimental Settings

Dataset: We used a dataset for ranking constructive comments on Japanese articles in Yahoo! News[3], which was prepared in Fujita et al. [7]. The dataset consists of triplets of an article title, comment, and constructiveness score. The constructiveness score (C-score) is defined as the number of crowdsourced workers, out of 40, who have judged a comment to be constructive. Therefore, the C-score is an integer ranging from 0 to 40. In this research, 130,000 comments from 1,300 articles were used as training data, 11,300 comments from 113 articles were used as validation data, and 42,436 comments from 200 articles were used as test data. In the training and validation data, 100 comments were randomly extracted in each article, whereas in the test data, all the comments were extracted assuming an actual service environment.

Preprocessing: We used a morphological analyzer MeCab[4] [14] with a neologism dictionary, NEologd[5] [20], for splitting Japanese texts into words. We

[3] https://research-lab.yahoo.co.jp/en/software/
[4] http://taku910.github.io/mecab/
[5] https://github.com/neologd/mecab-ipadic-neologd

replaced numbers with a special token and standardized the letter types by halfwidth to fullwidth[6]. We did not remove stop-words because function words will affect the performance in our task. We cutoff low-frequency words that appeared only three times or less in each dataset.

Model and Training: We used RankNet [1], a well-known pairwise ranking algorithm based on neural networks. Given a pair of two comments c_1 and c_2 on an article q, RankNet solves a binary classification problem of whether or not c_1 has a higher score than c_2. The score indicates the comment has high quality or not. We adopted the encoder-scorer structure for RankNet. The encoder consisted of two long short-term memory (LSTM) instances with 300 units to separately encode a comment and its title. The scorer predicted the ranking score of the comment via a fully-connected layer after concatenating the two encoded (comment and title) vectors. We used pre-trained word representations as the encoder input. They were obtained from a skip-gram model [16] trained with 1.5 million unlabeled news comments. 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. Both the dimensions of the hidden states of the encoders of article titles and comments were 300. In the experiments, we trained 100 different models by random initialization for the ensemble methods.

Evaluation: We used normalized discounted cumulative gain (NDCG@k) [1]. The NDCG@k is typically calculated in the top-k comments ranked by the ranking model and denoted by NDCG@$k = Z_k \sum_{i=1}^{k} \frac{\text{score}_i}{\log_2 (i+1)}$, where score$_i$ represents the true ranking score of the i-th comment ranked by the model, and Z_k is the normalization constant to scale the value between 0 and 1. In addition to NDCG@k, we use Precision@k as the second evaluation metrics. Precision@k is defined as the ratio of the correctly included comments in the inferred top-k comments to the true top-k comments. In the experiment, we evaluated the case of $k \in \{1, 5, 10\}$. Note that a well-known paper [10] in the information retrieval field determined NDCG to be more appropriate than Precision@k for graded-scores settings like ours.

3.2 Compared Methods

Ensemble Baselines: We prepared the following methods as baselines. `RankSVM` and `RankNet` are baselines of a single model. `ScoreAvg`, `RankAvg`, `TopkAvg`, and `NormAvg` are commonly used ensemble methods that combine multiple models in post-processing without training. `SupWeight` is the popular supervised ensemble method based on weighting.

- `RankSVM`: The best single RankSVM model proposed in Fujita et al. [7].
- `RankNet`: The best single RankNet model in 100 models for ensemble.
- `ScoreAvg`: Average output scores of the models for each comment.
- `RankAvg`: Average rank orders of each comment.
- `TopkAvg`: Select comments with higher scores than a threshold from each ranking and average their scores [5].

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

- `NormAvg`: Average normalized output scores of the model outputs, as typified by [2]. We used L2 normalization to each ranking as $r' = r/||r||$.
- `SupWeight`: Average weighted scores of the model outputs [19]. Scores are weighted on the basis of NDCG@k on the validation dataset. Note that their weights are constant values per model.
- `PostNDCG`: Select the best single model per article introduced in Sec. 2.2.

Our Methods: We show proposed methods as following:

- `HPA`: Hybrid the output selection method and a typical averaging method proposed in Sec. 2.3. We set $k = 50$, which obtained the highest accuracy in $k = \{5 \times n, n = 1, ..., 20\}$ on the validation dataset.
- `SPA`: Select models using the pseudo answer and average them (equal to `HPA` without the weighting). We set $k = 50$ which is the same setting of `HPA`.
- `WPA`: Average weighted model outputs using the pseudo answer (equal to `HPA` without the selecting).

3.3 Experimental Results

Our experimental results are shown in Tab. 1. As a result of the ensemble, we confirmed that all ensemble methods perform better than when using a single model. In particular, the proposed method `HPA` has achieved the highest NDCG@k. `PostNDCG` achieved higher accuracy than `RankNet`. This implies that the method of calculating the similarity between models using evaluation metrics for each article is effective. However, it was less accurate than the common averaging ensemble method such as `NormAvg`. Since models were originally trained by a relative comparison of rankings, preserving the diversity of models is more effective for improving performance than selecting models with high confidence by using `PostNDCG`. The unsupervised method `HPA` outperformed the supervised method `SupWeight`. Therefore, we confirmed that it is better to determine the important model from the similarity between the predicted rankings rather than learning it in advance using the labeled data.

	NDCG			Precision		
	@1	@5	@10	@1	@5	@10
`RankSVM`	73.38	74.59	76.01	15.5	30.20	38.95
`RankNet`	76.35	77.97	79.52	15.0	33.20	42.99
`ScoreAvg`	76.91	79.11	80.48	16.08	33.67	44.32
`RankAvg`	79.19	80.53	81.81	13.57	36.18	46.08
`TopkAvg`	78.38	80.52	81.57	14.07	35.38	46.08
`NormAvg`	79.83	80.77	82.16	**17.08**	37.18	46.48
`SupWeight`	78.64	80.33	81.94	16.28	35.47	46.58
`PostNDCG`	77.18	80.09	81.24	14.57	35.58	45.78
`HPA`	**79.87**	**81.43**	**82.33**	**17.08**	37.39	**47.34**
`SPA`	79.68	80.96	82.19	**17.08**	35.87	46.68
`WPA`	**79.87**	81.39	82.17	**17.08**	**37.88**	46.63

Table 1. NDCG@k and Precision@k scores (%) on ranking comment task ($k \in \{1, 5, 10\}$).

Furthermore, we verified the effectiveness of NDCG@k as a similarity function to calculate `HPA`, compared to other similarity functions. We selected Precision, cosine

	NDCG			Precision		
	@1	@5	@10	@1	@5	@10
NDCG@k	**79.87**	**81.43**	**82.33**	**17.08**	**37.39**	**47.34**
Precision@k	79.47	80.54	81.57	17.00	36.80	46.25
cos	77.80	80.21	81.82	14.07	35.90	46.93
kendall	78.10	80.44	81.61	16.28	36.88	46.85
spearman	78.70	80.52	81.62	15.50	37.18	46.58

Table 2. Comparison of similarity functions for `HPA`.

similarity, Kendall rank correlation coefficient [11], and Spearman rank correlation coefficient [21] as compared methods. Tab. 2 shows the results of `HPA` when the similarity function is changed. The NDCG@k functions outperformed other similarity functions. Furthermore, Precision@k performed better than cos. Note that Precision@k equals top-k cosine similarity. It indicates top-k focused measurement, evaluation metrics, is useful for the ensemble.

4 Related Work

Analyzing comments on online forums, including news comments, has been widely studied in recent years. This line of research has included many studies on ranking comments according to user feedback [9, 6, 22]. On the other hand, there has also been much research on analyzing news comments in terms of "constructiveness"[13, 18, 7]. The most related research is Fujita et al. [7]. They ranked comments by using the C-score to evaluate the quality, instead of relying on user feedback. They created a news comment ranking dataset and improved the model performance from the viewpoint of the dataset structure. In our research, we further improve the the performance from the viewpoint of the model structure.

In the ensemble methods for ranking task, there are methods to average model outputs [2, 5], as mentioned in Sec. 3.2. Our method expands those methods by denoising through the relationships between predicted rankings. There is also research on learning the query-dependent weights with semi-supervised ensemble learning in an information retrieval task [8]. This method focused on selecting documents that are highly relevant to a query (article). It is effective for information retrieval tasks but not for ranking news comments task, because almost all such comments would be associated with a news article.

There are also approaches that improve the ranking model according to evaluation metrics: NDCG@k, LambdaRank [3], and LambdaMART [4]. These methods handled model training by calculating NDCG@k between a gold ranking and a predicted one. It means NDCG@k was not used in inference. That fundamentally differs from our method which calculates NDCG@k between predicted rankings during inference.

5 Conclusion and Future Work

We proposed a hybrid unsupervised method of an output selection method and a typical averaging method. Our experiments showed that comparing predicted rankings using the evaluation metrics is effective for selecting and weighting models. For future work, we would like to compare the proposed method with the supervised ensemble method in terms of performance and speed. We also plan to combine various types of networks instead of using the same network structure.

Bibliography

[1] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to Rank Using Gradient Descent. In: Proceedings of the 22nd International Conference on Machine Learning (ICML 2005). pp. 89–96. ACM (2005), https://dl.acm.org/doi/abs/10.1145/1102351.1102363

[2] Burges, C., Svore, K., Bennett, P., Pastusiak, A., Wu, Q.: Learning to Rank Using an Ensemble of Lambda-gradient Models. In: Proceedings of the Learning to Rank Challenge. pp. 25–35. PMLR (2011), http://proceedings.mlr.press/v14/burges11a

[3] Burges, C.J., Ragno, R., Le, Q.V.: Learning to Rank with Nonsmooth Cost Functions. In: Advances in Neural Information Processing Systems 19 (NIPS 2007). pp. 193–200 (2007), https://papers.nips.cc/paper/2971-learning-to-rank-with-nonsmooth-cost-functions.pdf

[4] Burges, C.J.: From Ranknet to Lambdarank to Lambdamart: An Overview. Learning 11(23-581), 81 (2010), https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/MSR-TR-2010-82.pdf

[5] Cormack, G.V., Clarke, C.L., Buettcher, S.: 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). pp. 758–759. ACM (2009), https://dl.acm.org/doi/10.1145/1571941.1572114

[6] Das Sarma, A., Das Sarma, A., Gollapudi, S., Panigrahy, R.: Ranking Mechanisms in Twitter-like Forums. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010). pp. 21–30. ACM (2010), http://doi.org/10.1145/1718487.1718491

[7] Fujita, S., Kobayashi, H., Okumura, M.: Dataset Creation for Ranking Constructive News Comments. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019). pp. 2619–2626. Association for Computational Linguistics (2019), https://www.aclweb.org/anthology/P19-1250

[8] Hoi, S.C., Jin, R.: Semi-Supervised Ensemble Ranking. In: Proceedings of the 23rd National Conference on Artificial Intelligence-Volume 2 (AAAI 2008). pp. 634–639. AAAI Press (2008), https://www.aaai.org/Papers/AAAI/2008/AAAI08-101.pdf

[9] Hsu, C.F., Khabiri, E., Caverlee, J.: Ranking Comments on the Social Web. In: Proceedings of the 2009 International Conference on Computational Science and Engineering (CSE 2009). vol. 4, pp. 90–97. IEEE (2009), https://doi.org/10.1109/CSE.2009.109

[10] Järvelin, K., Kekäläinen, J.: Cumulated Gain-based Evaluation of IR Techniques. ACM Transactions on Information Systems (TOIS) 20(4), 422–446 (2002), http://doi.org/10.1145/582415.582418

[11] Kendall, M.G.: A New Measure of Rank Correlation. Biometrika 30(1/2), 81–93 (1938), https://www.jstor.org/stable/pdf/2332226.pdf

[12] Kobayashi, H.: Frustratingly Easy Model Ensemble for Abstractive Summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). pp. 4165–4176. Association for Computational Linguistics (2018), https://www.aclweb.org/anthology/D18-1449

[13] Kolhatkar, V., Taboada, M.: Constructive Language in News Comments. In: Proceedings of the First Workshop on Abusive Language Online. pp. 11–17. Association for Computational Linguistics (2017), http://www.aclweb.org/anthology/W17-3002

[14] Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004). pp. 230–237. Association for Computational Linguistics (2004), http://aclweb.org/anthology/W04-3230

[15] Littlestone, N., Warmuth, M.K.: The Weighted Majority Algorithm. Information and computation 108(2), 212–261 (1994), https://www.sciencedirect.com/science/article/pii/S0890540184710091

[16] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality. In: Advances in Neural Information Processing Systems 26 (NIPS 2013). pp. 3111–3119 (2013), https://arxiv.org/abs/1310.4546

[17] Naftaly, U., Intrator, N., Horn, D.: Optimal ensemble averaging of neural networks. Network: Computation in Neural Systems 8(3), 283–296 (1997), https://www.tandfonline.com/doi/abs/10.1088/0954-898X_8_3_004

[18] Napoles, C., Pappu, A., Tetreault, J.R.: Automatically Identifying Good Conversations Online (Yes, They Do Exist!). In: Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017). pp. 628–631. AAAI Press (2017), https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15673

[19] Opitz, D.W., Shavlik, J.W.: Actively Searching for an Effective Neural Network Ensemble. Connection Science 8(3-4), 337–354 (1996), https://research.cs.wisc.edu/machine-learning/shavlik-group/opitz.consci96.pdf

[20] Sato, T., Hashimoto, T., Okumura, M.: 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. pp. NLP2017–B6–1. The Association for Natural Language Processing (2017)

[21] Spearman, C.: The Proof and Measurement of Association between Two Things. The American Journal of Psychology 15(1), 72–101 (1904), http://digamoo.free.fr/spearman1904a.pdf

[22] Wei, Z., Liu, Y., Li, Y.: 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). vol. 2, pp. 195–200. Association for Computational Linguistics (2016), https://www.aclweb.org/anthology/P16-2032