Mind News Recommendation Technical Report

Qin Ruan

University College Dublin, Ireland

Task description

Everyday, online news services publish massive news articles. In order to improve user browsing experience and avoid information overload, recommender systems have been widely adopted by these online news platforms. Based on the large-scale news datasets MIND from Microsoft, in this competition, I exploited different latest news recommendation algorithms and explored to enhance these algorithms by expanding the features and harnessing ensembling methods.

Data analysis

I identified the features of mind news dataset into three parts: users, news items,interactions.

User_id is the only feature I can observe directly from users. This potentially implies that the current dataset is more suited for constructing a content-based news recommender system.

Every news item has news_id, category, subcategory, title, abstract, url, title entities and abstract entities. Statistical information presented in Table 1 shows that there are some missing values in the attributes abstract, url, title_entity and abstract_entity. Figure 1 describes the length distribution of title, abstract on train, dev and test respectively.

	news_id	category	sub_category	title	abstract	url	title_entity	abstract_entity
train set								
count	101527	101527	101527	101527	96112	101527	101524	101521
unique	101527	18	285	98388	91654	101526	66863	72168
dev set								
count	72023	72023	72023	72023	68400	72023	72021	72018
unique	72023	17	269	70316	65871	72022	48107	51034
test set								
count	120959	120959	120959	120959	114256	120958	120953	120950
unique	120959	18	290	117306	109113	120956	79337	85928

Table 1. Description of news on train set, dev set, test set respectively.

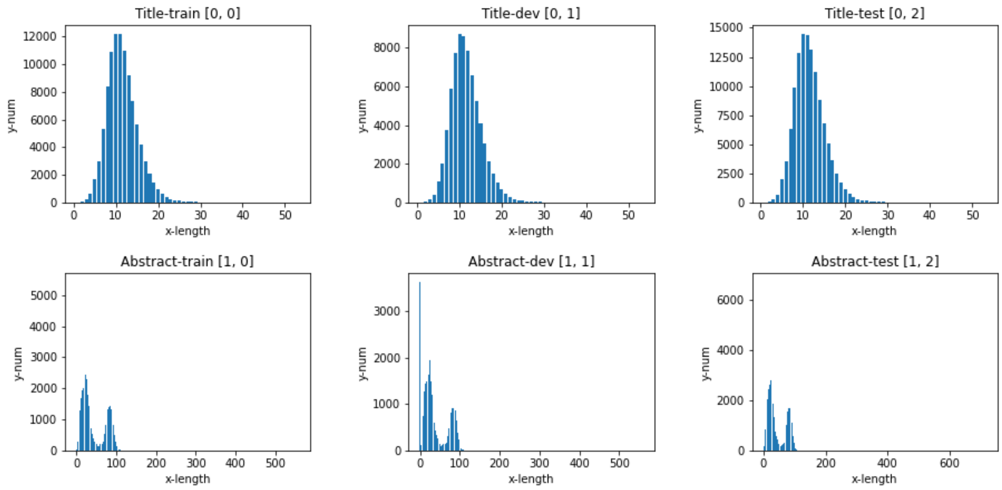

Figure 1. The length distribution of title and abstract on train, dev and test set.

As for interactive information on news and users, I counted the number of positive and negative samples for the train set, dev set and test set respectively. The results are presented in Table 2.

	#clicks	#non-clicks
train	3383656	80123718
dev	574845	13510712

Table 2. The number of positive and negative clicks on the train and dev set.

Submitted solution

1. Dataset selection

During experiments, I found that evaluating the model does not require the full dev set. Also, according to Wu et al., 2020, the train set of MIND is the first six-day record of week 5, and the dev set is the last day of week 5. Therefore,I sorted the dev set by ascending timestamp. I then divided the first 50% dev set into train set, and kept the remaining 50% as evaluated. Finally, the new training set and dev set were shuffled respectively. For each candidate items generated from the impression, positive clicks of each impression are retained, and negative clicks are filtered by the ratio of positive and negative 1:4.

2.Features selection

The attribute title, abstract, category, subcategory and user_id were chosen as features. Based on previous data analysis, I realized that there are a lot of missing values in abstract, so I tried to modify them by crawling the news content based on the news url, and I then used an open summarization library called Pegasus (Zhang et al., 2019) to generate a summary for each news item.

3.Models selection

Our base models modified from LSTUR (An et al., 2019), NAML(Wu et al., 2019a), NRMS(Wu et al., 2019b), TARN(Wu et al., 2019c), I did some experiments to find useful knowledge from these models. I listed the modified schemas as follow:

3.1. LSTUR+NAML: use an attentive multi-view learning way to encode news, the way LSTUR learns user behaviors is retained.

3.2. LSTUR+NAML+(summary): replace original abstract to summary, which generated from summarization library pegasus.

3.3.LSTUR+NAML+(uid): use maximum entropy strategy to select learnable user_id throughout total user list.

3.4.NRMS+NAML: use an attentive multi-view learning way to encode news, the way NRMS learns user behaviors is retained.

3.5.LSTUR+NAML+Category: learn rank task and category prediction task together.

4.Ensembles

In the field of machine learning, ensemble algorithms integrate multiple learning methods and usually achieve better prediction results than a single algorithm. Here, a simple average ensemble method is used to perform a secondary fusion of the results of the base models.

5.Training details

Prediction results are evaluated by four metrics in the recommendation field, they are auc, mrr, ndcg@5, ndcg@10 respectively, the auc is the main metric to be considered. Hence, the model selected during training based on the performance of auc on the dev set.

6.Framework

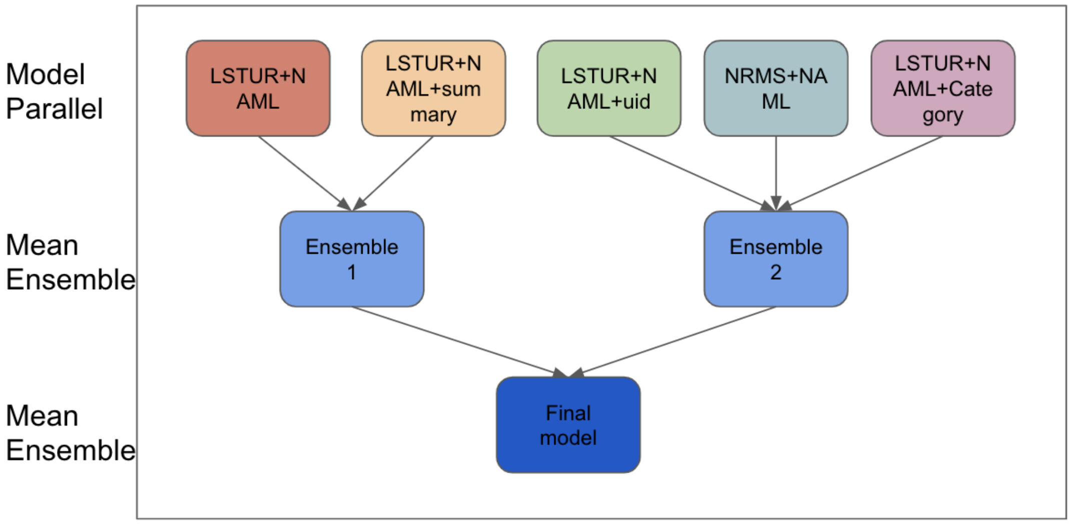

Figure 2. The Training architecture for the task of ranking candidates.

7.Experimental results on dev and 10%test

	AUC	MRR	nDCG@5	nDCG@10
Eval@dev				
LSTUR+NAML	0.7001	0.3416	0.4426	0.3809
LSTUR+NAML+(summary)	0.7056	0.3469	0.4471	0.3852
LSTUR+NAML+(uid)	0.7013	0.3406	0.4411	0.3798
NRMS+NAML	0.7051	0.3457	0.4477	0.3863
LSTUR+NAML+Category	0.7067	0.345	0.446	0.3845
Eval@test10%				
LSTUR+NAML	0.6967	0.3444	0.3767	0.4338
LSTUR+NAML+(summary)	0.6965	0.3455	0.3778	0.4344
LSTUR+NAML+(uid)	0.6945	0.3427	0.3747	0.4318
NRMS+NAML	0.6912	0.3429	0.3754	0.4324
LSTUR+NAML+Category	0.6954	0.3433	0.3749	0.4319
Ensembles	0.7028	0.3499	0.3833	0.4399
Eval@test				
Submitted to sys.	0.7032	0.3496	0.383	0.4397

Table 3. The performance of evaluation metrics on dev set and 10% test set.

Acknowledgements

As a newbie in the field of news recommender systems, I would like to express my gratitude to Microsoft for providing this opportunity. Due to this competition, I got deep insight into the baseline models mentioned in Wu et al., 2020. I also did exploratory experiments on the top of the open platform —— Microsoft recommenders. These are valuable resources for researching news recommender systems.

Second, I would like to express my warm appreciation to my supervisor Dr. Ruihai Dong, who led me into the world of the recommender system, pushed me forward and encouraged me not to give up.

Last my thanks would give my parents, who supported me and always believed in me.

References

1.An, M., Wu, F., Wu, C., Zhang, K., Liu, Z. and Xie, X., 2019, July. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 336-345)

2.Microsoft. GitHub. 2020. *Microsoft/Recommenders.* [online] Available at: <https://github.com/microsoft/recommenders> [Accessed 23 September 2020].

3.Wu, C., Wu, F., An, M., Huang, J., Huang, Y. and Xie, X., 2019a. Neural news recommendation with attentive multi-view learning. In *IJCAI-19, pages 3863–3869*.

4.Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y. and Xie, X., 2019b, November. Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 6390-6395).

5.Wu, C., Wu, F., An, M., Huang, Y. and Xie, X., 2019c, July. Neural news recommendation with topic-aware news representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1154-1159).

6.Wu, F., Qiao, Y., Chen, J.H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W. and Zhou, M., 2020, July. MIND: A Large-scale Dataset for News Recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 3597-3606).

7.Zhang, J., Zhao, Y., Saleh, M. and Liu, P.J., 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*.