YJRS at the NTCIR-14 OpenLiveQ-2 Task

Tomohiro Manabe Sumio Fujita Akiomi Nishida
Yahoo Japan Corporation
{tomanabe, sufujita, anishida}@yahoo-corp.jp

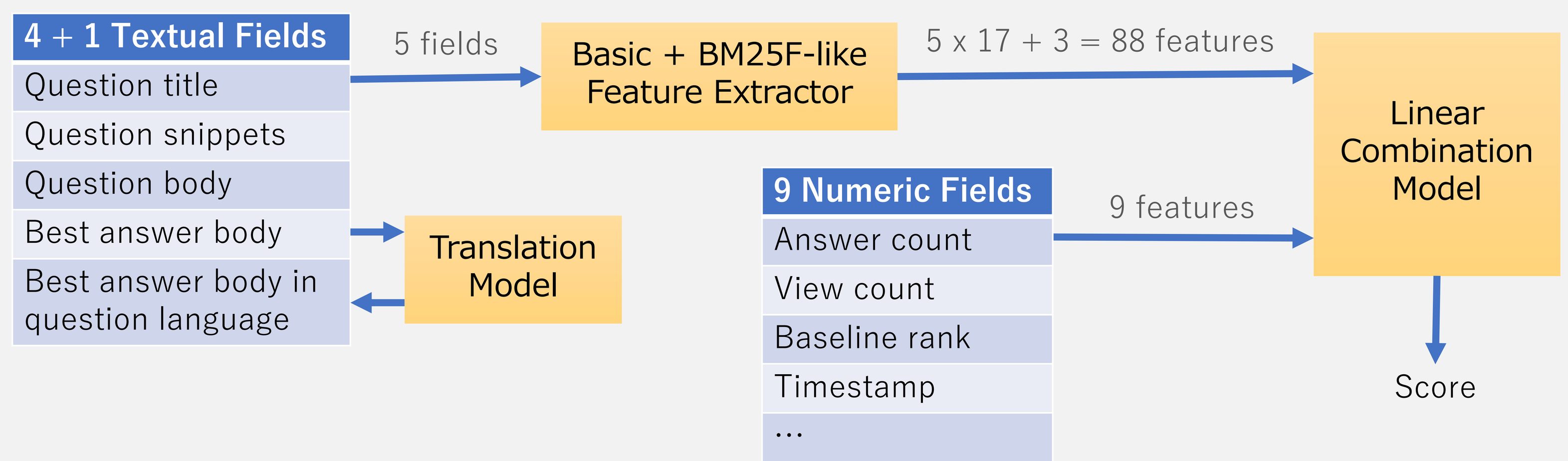
- OpenLiveQ is an information retrieval task for community QA sites.
- "*The task is simply defined as follows: Given a query and a set of questions with answers, return a ranked list of questions.*"
- Data: 2,000 queries x 1,000 questions (with additional information, e.g., answers and snippets) and CTR as a relevance measure

Our Approaches

We started with a linear combination of 77 basic features trained with the Coordinate Ascent (CA) method. This table summarizes our approaches.

Run IDs	Ranking Model	Training Method	Feature Set	Feature Count
91, 95	Linear combination	CA	Basic	77
92			BM25F	80
93			Translated	94
136			All	97
100, 113	Neural ranking	ListNet	Basic	77
144, 148	Ensemble trees	LightGBM	Basic	77

Below is the system pipeline of our CA + All method.



CA + Basic Features [1] (IDs: 91, 95)

- Linear combination of 77 basic features
  - 4 fields x 17 textual features, e.g. TF, TFIDF, BM25, language models, …
  - 9 numeric features, e.g. answer count, view count, timestamp …
- Weights are optimized by CA.
- ID: 95 is just a retry (Note: CA is probabilistic).

CA + BM25F Features (ID: 92)

- The best performer in the previous round of the task [2]
- 77 basic features + 3 BM25F-like features
- Five-fold cross validation, nDCG@10 as the objective function

CA + Translated Features (ID: 93)

- Inspired by the other best performer in the previous round of the task [3]
- The vocabulary of queries must be more similar to questions than to answers.
- This method translates the best answer body texts into the "question language" with the GIZA++ toolkit and a public QA corpus.
- 77 basic features + 17 textual features of the translated field

CA + All Features (ID: 136)

- 77 basic features + 3 BM25F-like features + 17 translated features

ListNet + Basic Features (IDs: 100, 113)

- 77 basic features
- 3-layer fully-connected feed-forward neural network with 200 hidden nodes
- Trained with ListNet [4]
- ID: 113 is with five-fold cross validation.

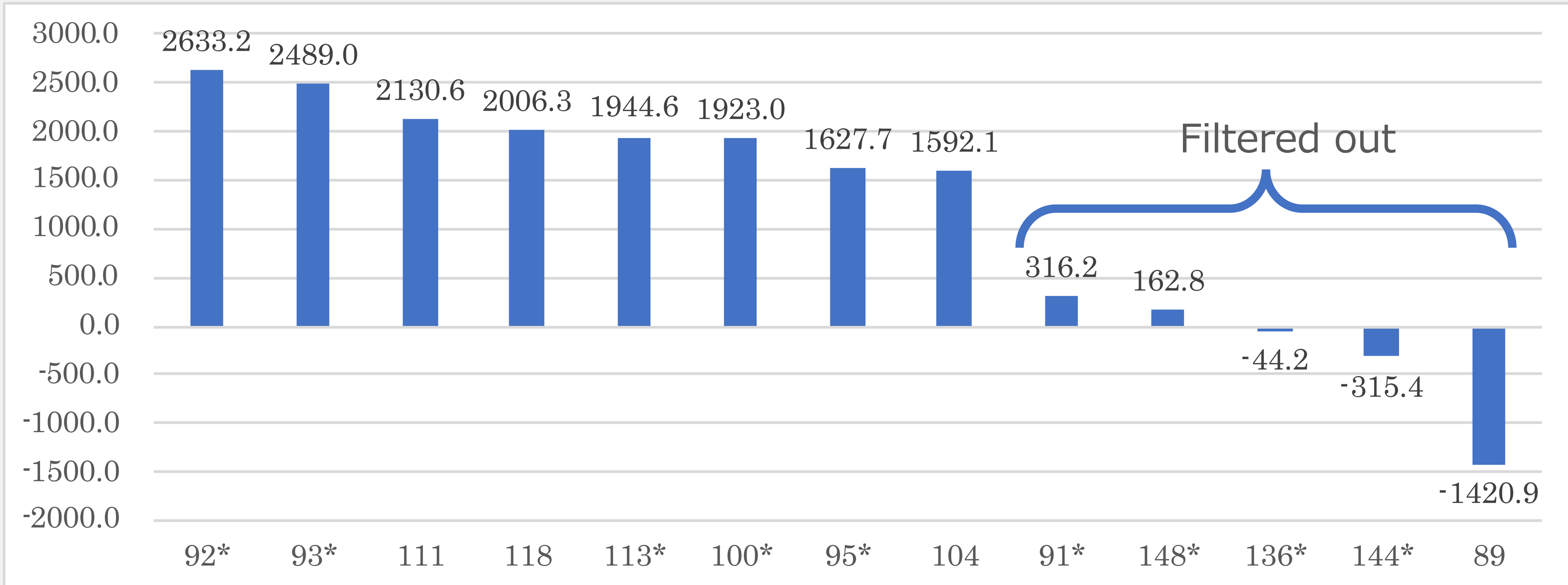
LightGBM + Basic Features (IDs: 144, 148)

- 77 basic features
- Linear combination of 100 regression trees with 15 leaves each
- Trained with LightGBM [5]
- ID: 148 is roughly tuned.

Evaluations

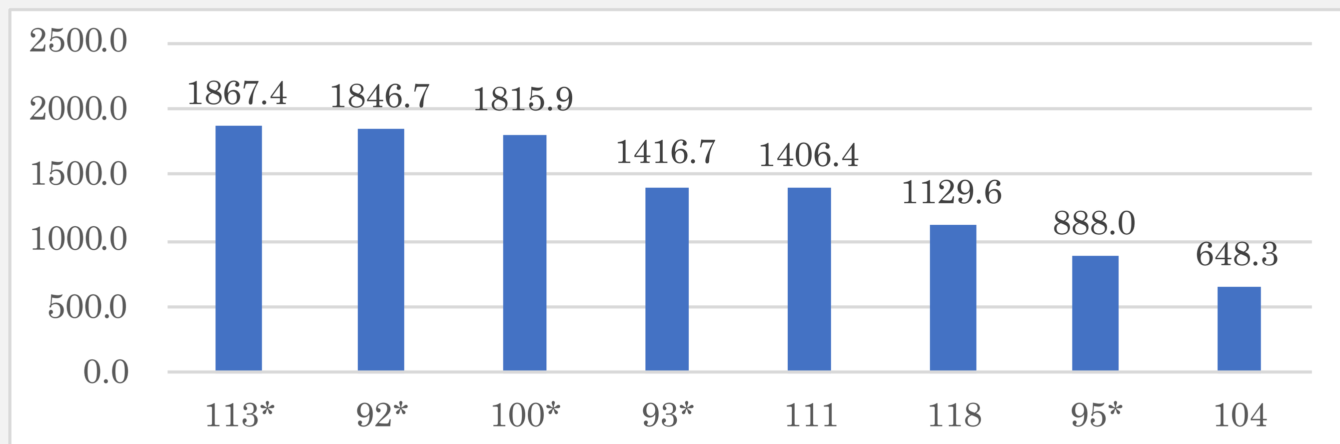
- In this task, all runs were evaluated online with the Pairwise Preference Multileaving (PPM) method.
- When a user submits a query, PPM mixes multiple rankings for the query into a list and show it to the user. Clicks on some questions indicate the user's preference in questions. The PPM gives credits to rankings which accord with the preference.
- The entire evaluation process spanned the first round (61 runs, 164k PVs) and the second round (30 runs, 149k PVs).
  - Note: We list only our runs (with *) and other team's best runs due to space limitation.

First Round of Multileaved Comparisons



- CA + BM25F (ID: 92), CA + Translated (ID: 93) > ListNet (IDs: 113, 100) > CA + Basic (IDs: 95, 91), LightGBM (IDs: 148, 144), CA + All (ID: 136)
- The performance of our linear combination runs were not stable.
  - Performance of our CA + Basic runs were quite different although they share the same feature set.
  - Performance of our CA + All run was quite worse than ones of our CA + BM25F and our CA + Translated runs which use only subsets of features.
- Our LightGBM runs did not work. This may be because of small training data.

Second Round of Multileaved Comparisons



- ListNet (IDs: 113, 100), CA + BM25F (ID: 92) > CA + translated (ID: 93) > CA + Basic (ID: 95)
- The ListNet runs occupied better positions than in the first round.
- The other tendencies were same as the ones in the first round.

Conclusions

- Among our runs, some combinations of a relatively simple learning-to-rank method and a reasonable feature set performed well.
- Namely, our ListNet (+ Basic) and CA + BM25F methods performed well
- Our linear combination runs resulted in unstable performance whereas ListNet runs, which used only basic features, were quite promising.

References

- [1] https://github.com/mpkato/openliveq/blob/master/README.md
- [2] Manabe *et al*. YJRS at the NTCIR-13 OpenLiveQ Task, NTCIR 2017.
- [3] Chen *et al*. Erler at the NTCIR-13 OpenLiveQ task, NTCIR 2017.
- [4] Cao *et al*. Learning to rank: from pairwise approach to listwise approach, ICML 2007.
- [5] https://github.com/microsoft/LightGBM

NTCIR-14, June 10-13, 2019, NII, Tokyo, Japan