Reciprocal Rank Fusion

This is a rank aggregation method, which for a given document takes the ranks given to that document by different ranking engines and combines them to get an aggregate rrf score.
Let $r_i(d)$ be the rank given to document d by the i^{th} ranking engine. The rrf score given to document d is as follows:

$$rrf(d) = \sum_i \frac{1}{K + r_i(d)}$$

If the i^{th} ranking engine does not retrieve document d then $r_i(d) = \infty$
For a given Query we compute $rrf(d)$ for all documents and re-rank them according to these scores, the intuition being if many ranking engines predict that a document is highly ranked it is probably relevant and should be ranked higher.
K is one of the hyperparameters of this model and the results for precision values with varying K are as follows:

K	maP	P@5	P@10
10	0.5345	0.4005	0.2966
30	0.5516	0.4138	0.3032
60	0.5536	0.4199	0.3060
100	0.5560	0.4232	0.3069
200	0.5594	0.4255	0.3077
400	0.5592	0.4255	0.3092
500	0.5588	0.4255	0.3088
1000	0.5345	0.4005	0.2966

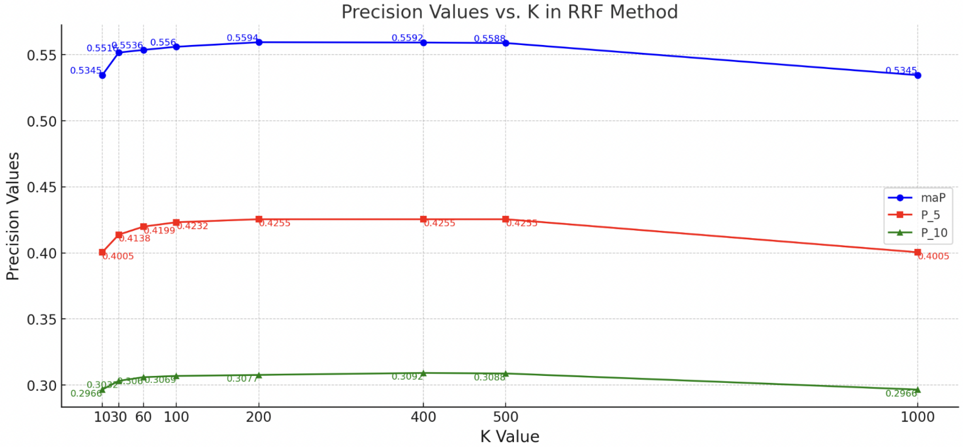

Figure 1: Effect of K on Precision in Reciprocal Rank Fusion

BordaCount Fusion

If the i^{th} Ranking System retrieves N_i documents, then the 1^{st} documents recieves N_i points, the second recieves $N_i - 1$ points and so on the j^{th} ranked document receives $N_i - j + 1$ points from this ranking engine.

The score of a document is the sum of the points received by it from each ranking engine.

For each query we score all the documents by the scheme mentioned above and rerank the documents according to these scores. Ties are resolved arbitrarily.

There are no hyper-parameters for this model.

maP	**P@5**	**P@10**
0.5635	0.4278	0.3108

Condorcet Voting

This scheme is a rank-based aggregation method. The score of a document is the number of other documents it beats in head-to-head ties. D_i beats D_j head-to-head if more ranking systems rank D_i ahead of D_j

This model does not have any hyper-parameters.

Ties are resolved by BordaCounts, and ties in BordaCounts are resolved arbitrarily.

The results of Condorcet Fusion are as follows:

maP	**P@5**	**P@10**
0.5596	0.4258	0.3106

BayesFuse

This is a Bayesian Framework for Rank Fusion. If there are n ranking engines and r_i denotes the ranking provided by the i^{th} ranking engine.

We need to estimate the following for a given document.

$$P_{rel} = Pr[rel|r_1, \ldots r_n]$$

$$P_{irr} = Pr[irr|r_1, \ldots r_n]$$

Where P_{rel} is the probability that the document is relevant given the ranking evidence.

P_{irr} is the probability that the document is irrelevant given the ranking evidence.

We can rank document by their Odds of Relevance.

$$\mathcal{O}_{rel} = \frac{P_{rel}}{P_{irr}}$$

Taking the Logarithm preservers the rankings.

Using the Bayes rule we get

$$P_{rel} = \frac{Pr[r_1, \dots r_n|rel]Pr[rel]}{Pr[r_1, \dots r_n]}$$

$$P_{irr} = \frac{Pr[r_1, \dots r_n|irr]Pr[irr]}{Pr[r_1, \dots r_n]}$$

$$\mathcal{O}_{rel} = \frac{Pr[r_1, \dots r_n|rel]Pr[rel]}{Pr[r_1, \dots r_n|irr]Pr[irr]}$$

Assuming independence of ranking engines.

$$\mathcal{O}_{rel} = \frac{\Pi_{i=1}^{n} Pr[r_i|rel]Pr[rel]}{\Pi_{i=1}^{n} Pr[r_i|irr]Pr[irr]}$$

$$log(\mathcal{O}_{rel}) = \sum_{i=1}^{n} log(\frac{Pr[r_i|rel]}{Pr[r_i|irr]}) + \frac{Pr[rel]}{Pr[irr]}$$

The ranking is thus determined by the following score for each document

$$\sum_{i=1}^{n} log(\frac{Pr[r_i|rel]}{Pr[r_i|irr]})$$

$Pr[r_i|rel]$ can be determined using relevance labels and precision metrics for each ranking system. I have used Gaussian Mixture Models to generate smooth Precision curves and estimate the probabilities.

There are no hyper-parameters in our model and ties are resolved arbitrarily. The results for BayesFuse are as follows:

maP	P@5	P@10
0.5839	0.4441	0.3170

Weighted BordaFuse

This method is very similar to BordaFuse except that instead of simply summing BordaCounts from different ranking systems, we do a weighted average.

The weights of a ranking system are estimated using statistics such as mean precision values.

There are no hyperparameters and ties are resolved arbitrarily.

The results for this model are as follows:

maP	P@5	P@10
0.5675	0.4293	0.3129

Results

Method	maP	P@5	P@10
Reciprocal Rank Fusion	0.5560	0.4232	0.3069
BordaFuse	0.5635	0.4278	0.3108
Condorcet Voting	0.5596	0.4258	0.3106
Bayes Fuse	**0.5839**	**0.4441**	**0.3170**
Weighted BordaFuse	0.5675	0.4293	0.3129