Merge-Tie-Judge: Low-Cost Preference Judgments with Ties

Kai Hui, Klaus Berberich
Max Planck Institute for Informatics

Background

Two rankings from system $S_1$ and $S_2$:

$S_1: d_3, d_1, d_2$

$S_2: d_5, d_4, d_2, d_1$

Graded judgments on five documents:

$d_1: 0, d_2: 1, d_3: 1, d_4: 1, d_5: 2$

Preference judgments:

$d_1 < d_2, d_1 < d_3, \ldots, d_3 < d_4, d_3 < d_5, d_4 < d_5$

A ground-truth document ranking:

$d_1 < d_2 < d_3 < d_4 < d_5$

Represented as tie partitions:

$\{d_1\} < \{d_2, d_3, d_4\} < \{d_5\}$

Motivation

- Preference judgments achieve better quality;
- Require $O(N_d \log N_d)$ judgments, which are too expensive for practical usage;
- Theoretically, employing ties could reduce the number of judgments to $O(2 N_t \log N_t + N_d)$ [1] over $N_d$ documents and $N_t$ tie partitions;
- This work attempts to implement the finding.

Merge-Tie-Judge

- $O(2 N_t \log N_t + N_d)$ is achievable only when tie partitions are recognized beforehand
- Henceforth, prioritize the pairs that are more likely to be tied during judgments

Algorithm

0.1) regard individual documents as tie partitions $c_1, c_2, \ldots, c_{N_d}$;
0.2) initialization of the probability for two partitions being tied, denoted as $Pr_{tie}(c_i, c_j)$, randomly or with ActiveSVM using document content;

While there exist unjudged pairs of partitions:

1) pick up $c_i$ and $c_j$ that are most likely to be tied based on $Pr_{tie}(c_i, c_j)$;
2) manually judge the selected pairs;
If the judged pair is tied:

3) merge them to generate new partition;
4) update $Pr_{tie}(c_i, c_j)$ based on the manual judgments;

Dataset: graded judgments from TREC Web Track 2011–2014 for ad-hoc task including 200 queries.

Number of judgments required to obtain the same ground truth.

- Merge-Tie-Judge significantly reduces the number of judgments ($\&$);
- The usage of ActiveSVM further improve the robustness ($\cdot$).