Pooling for User-Oriented Evaluation Measures

Gaurav Baruah, Adam Roegiest, Mark D. Smucker

{gbaruah, aroegiest, mark.smucker}@uwaterloo.ca

BACKGROUND: Consider a user model for streaming information access [MSU 2015] (e.g. a stream of news updates filtered from the web).

QUESTION: Can we construct an evaluation pool using updates that have a higher chance of being read?

1) $P(d_i)$: Probability of document $i$ being read

$\text{System A}$

User 1: read $d_1, d_2, d_3, d_4, d_5, d_6, d_7$

User 2: read $d_1, d_2, d_3, d_4$

User 3: read $d_1, d_2, d_3, d_4, d_5$

$\text{System B}$

Number of updates read across all users = 17

Balanced probabilities

Unbalanced probabilities

$P(d_1) = \frac{1}{3}$

$P(d_2) = \frac{1}{3}$

$P(d_3) = \frac{1}{3}$

$P(d_4) = \frac{1}{3}$

$P(d_5) = \frac{1}{3}$

$P(d_6) = \frac{1}{3}$

$P(d_7) = \frac{1}{3}$

$P(d_8) = \frac{1}{3}$

Averaged across all users. Favors users who read more.

3) Alternative pooling: Probability mass cover

$\text{System A}$

All Users

$\text{System B}$

All Users

Local probability mass cover = 0.5 (local to each run)

System A: $P(d_1) + P(d_2) + P(d_3) = 0.176 + 0.176 + 0.176 = 0.528$

System B: $P(d_1) + P(d_2) + P(d_3) = 0.333 + 0.222 = 0.555$

$\text{Pool} = \{d_1, d_2, d_3, d_4\}$

Global probability mass cover = 0.5 (across all runs)

$P(d_1) + P(d_2) + P(d_3) + P(d_4) = 0.176 + 0.176 + 0.176 + 0.176 = 0.528$

$\text{Pool} = \{d_1, d_2, d_3, d_4\}$

OVERVIEW

The frequency and duration of sessions depends on the user’s level of interest and availability of time. The number of updates read in a session depends on the user’s reading speed [TWS 2014].

OBSERVATION: Document score-based pooling helps to identify relevant updates, however, many identified relevant updates are not read by (modeled) users.

We simulated 10,000 users reading updates from the runs submitted to the Temporal Summarization Track [TST] 2013 [TST 2013]. For topic 10, 1616 relevant updates were returned across all runs.

- only 5 were read by all modeled users.
- 25 were read by > 99% of users.
- 546 were read by < 1% of the users.

Other topics at TST 2013 have similar proportions of read relevant updates.

KEY REFERENCES:


CONCLUSIONS: Score-based and probability-based depth pooling result in different pools. Given probabilities of updates being read, we can construct a pool with a specified probability mass cover.

FUTURE WORK: Test-collections based on probability-based pooling; Are they effective for measuring relative system performance? Are they reusable? How might they work for different user models?