**OFFLINE AND ONLINE RANKING MODEL EVALUATION IN INDUSTRY**

What happens in the industry, where real users interact with the system, business interests affect the concept of relevance and pre-defined relevance judgments are not available?

### STEP 1. DATA COLLECTION

**IMPLICIT FEEDBACK**
- **User Interactions**
  1. Collection of users' interactions: REST API, SaaS SOLUTION
  2. Model interactions as JSON objects
  3. Relevance label estimation using interactions aggregation: Click-Through Rate, Add-To-Cart Rate, ...
  4. Test set extraction/creation and Kibana dashboard creation.

**EXPLICIT FEEDBACK**
- Team of experts

### STEP 2. EVALUATION APPROACH

**A/B TESTING**
- **Approach choice:** REST API, SaaS SOLUTION
  - Design parametric search-API
  - Assign users to a population through cookies
  - Tag interactions with the test group

**INTERLEAVING**
- Team Draft available from: Solr version 8.8.0
- Winner estimation process needed (Python script):
  - \( \Delta_{\text{AB}} = \frac{\text{wins}(A) + \frac{1}{2}\text{tie}(A, B)}{\text{wins}(A) + \text{wins}(B) + \text{tie}(A, B)} - 0.5 \)
  - Query distribution analysis (Long tail vs Uniform)

**FOR BOTH**
- **Statistical Significance**
  - One-way ANOVA test
  - Tukey test
  - Kruskal-Wallis test
  - Dunn test

### STEP 3. EXPERIMENT DESIGN

**Control**
- Offline evaluation
  - Winner - Ranking model v1.0
- Online evaluation
  - Winner - Control

**Ranking model v1.0**
- Offline evaluation
  - Compare
- Online evaluation
  - Winner - Ranking model v2.0

**Ranking model v2.0**
- Offline evaluation
  - Compare
- Online evaluation
  - Winner - Ranking model v3.0

**Ranking model v3.0**
- Offline evaluation
  - Compare
- Online evaluation
  - Winner - Ranking model v3.0

Baby steps between experiments: each experiment compares similar models (few features more, different normalization). One experiment per platform (desktop, mobile, ...). Evaluate 2-3 models at the time.

### STEP 4. PITFALLS

**QUERY**
- **Query id generation:** too-specific vs too-generic
  - Too short/too long ranked lists per query → balanced is needed
  - Users tend to click on top-ranking results
- **Number of results:** large result set query vs small result set query
  - Small result set queries → expected small ranking model impact
  - Many small result set queries → cause noise in the evaluation

**INTERACTIONS**
- **Noise:** position bias - source pages - errors during collection
  - Users tend to click on top-ranking results
  - Online evaluation → select only interactions from pages that use rank models

**METRIC**
- **Choose metrics:** industry's interests
  - Estimate offline relevance label → business objective (clicks, add-to-cart, downloads, ...)
  - Offline metrics need support by Online metrics

**TEST SET**
- **Per query data and relevance distribution:** unbalanced
  - Queries with a single sample
  - Queries with a single relevance type

### ALTERNATIVE

**FOR BOTH**
- **Explicit Feedback**
  - Team of experts

**FOR BOTH**
- **Implicit Feedback**
  - Collection of users’ interactions: REST API, SaaS SOLUTION
  - Model interactions as JSON objects
  - Tag interactions with the test group

### WHEN DO WE STOP?

1. Statistical significance
2. Number of users
3. Time = development iteration length

---

Alessandro Benedetti - Director and R&D Software Engineer
Anna Ruggero - R&D Software Engineer and Search Consultant