An Experimental Study of Multi-stage Retrieval Systems
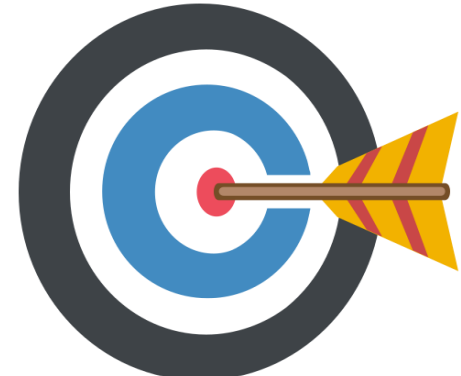
Mohammed Yusuf Ansari
Advisor: Mohammad Hammoud

Information Retrieval

- Information Retrieval (IR) is concerned with *search* over large *unstructured data* like web pages, emails, and image libraries, among others

- Any IR system has two major objectives:

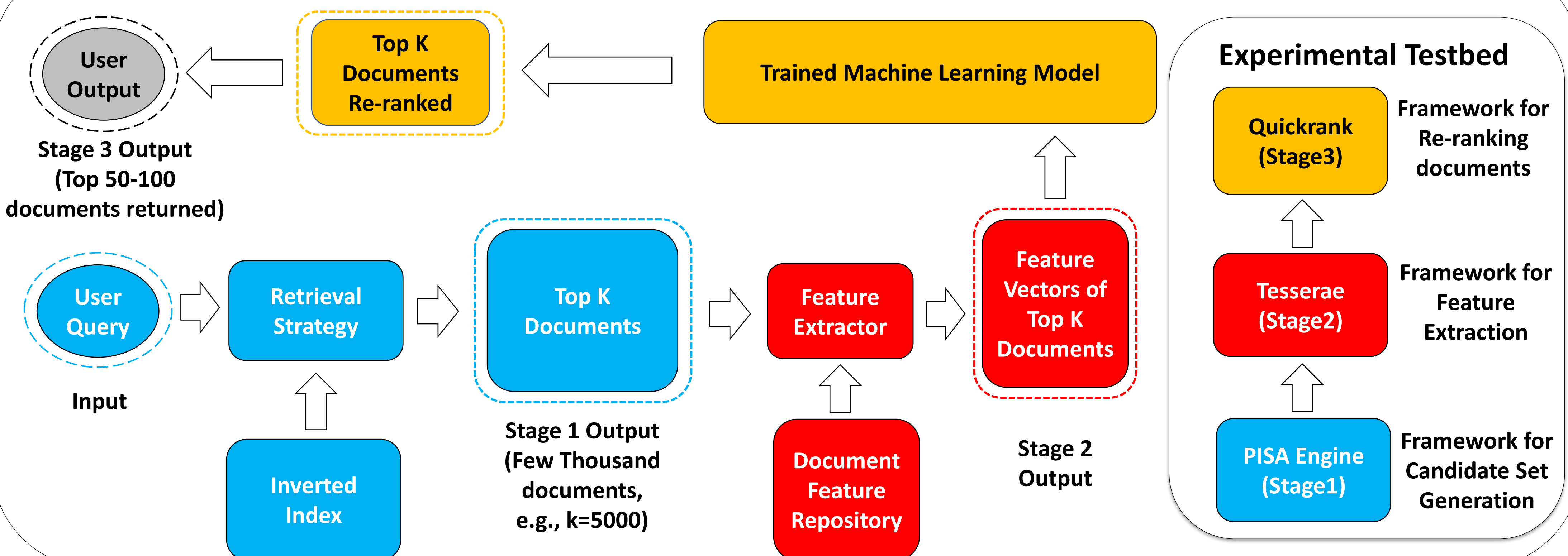**Speed (or *Efficiency*)**   **Accuracy (or *Effectiveness*)**

Problem Description

Observation: Past work aimed at optimizing a particular configuration parameter of a single stage in standard three-stage retrieval systems
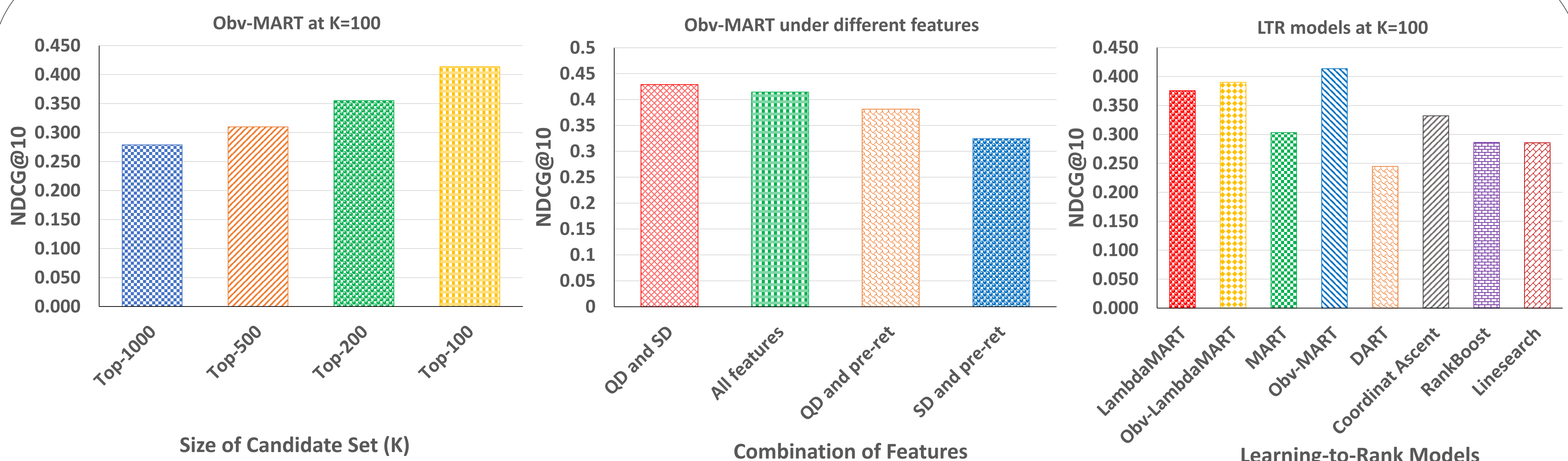
Problem: Learning-to-rank models in stage 3 were developed in obliviousness to the configurations of the first two stages

Solution: A comprehensive empirical study to examine and leverage the performance correlation between the three different stages of IR systems

Three Stages of Multi-stage Retrieval Systems

User Output

Top K Documents Re-ranked

Stage 3 Output (Top 50-100 documents returned)

Trained Machine Learning Model

User Query → **Retrieval Strategy** → **Top K Documents** → **Feature Extractor** → **Feature Vectors of Top K Documents**

Input

Inverted Index

Stage 1 Output (Few Thousand documents, e.g., k=5000)

Document Feature Repository

Stage 2 Output

Experimental Testbed

Quickrank (Stage3) — Framework for Re-ranking documents

Tesserae (Stage2) — Framework for Feature Extraction

PISA Engine (Stage1) — Framework for Candidate Set Generation

Experimental Evaluation

Obv-MART at K=100

NDCG@10 — Size of Candidate Set (K): Top-1000, Top-500, Top-200, Top-100

Obv-MART under different features

NDCG@10 — Combination of Features: QD and SD, All features, QD and pre-ret, SD and pre-ret

LTR models at K=100

NDCG@10 — Learning-to-Rank Models: LambdaMART, Obv-LambdaMART, MART, Obv-MART, DART, Coordinat Ascent, RankBoost, Linesearch

⇒ For candidate set generation stage, we recommend candidate set size (K) equal to 100

⇒ For feature extraction stage, we recommend the combination of query-dependent and static-document features

⇒ For candidate set re-ranking stage, we recommend Obv-LamdaMART or Obv-MART model