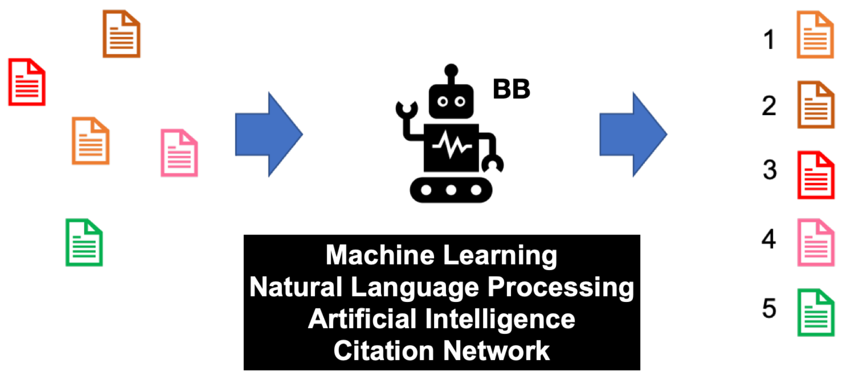
Enhancing Reliability in Automated Literature Screening: A Resample Algorithm

Zhipeng Hou and Elizabeth Tipton
STEPP Center, Northwestern University

Overview

- Automated literature screening
- A resample algorithm
- Questions

Automated Literature Screening



1. Train a predictive model to predict the probability of relevancy for each candidate paper.
2. Only screen the papers with high probability of relevance.

Example 1

- Task: Screening 10,000 candidate papers.

- Example Workflow 1 (Supervised Learning):
    1. **Initial Screening**: Human reviewers screen 4,000 candidate papers randomly.
    2. **Model Training**: Utilize the screened papers as the training set to train a predictive model, denoted as M.
    3. **Final Screening**: Subsequently, screen the remaining 6,000 papers based on the predictive relevancy determined by the model.

Example 2

- Task: Screening 10,000 candidate papers.

- Example Workflow 2 (Learning with Model Updates):
  1. **Initial Screening**: Human reviewers randomly screen 2,000 candidate papers.
  2. **Model Training**: Utilize the screened papers as the training set to train the initial predictive model, denoted as M_1.
  3. **Secondary Screening**: Subsequently, select 2,000 candidate records with the highest predicted relevancy from the pool of 8,000 unscreened candidates.
  4. **Model Update**: Use the screened 4,000 papers to train an updated model.
  5. **Final Screening**: Screen the remaining 6,000 papers based on the predictive relevancy determined by the updated model.

Example 3

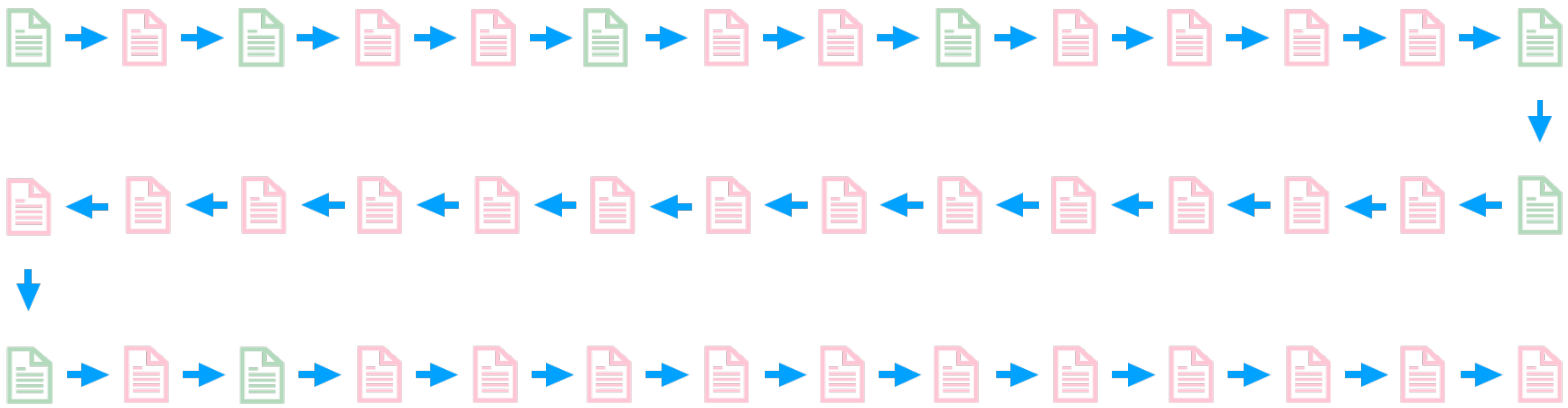
- Task: Screening 10,000 candidate papers.

- Example Workflow 3 (Active Learning):
  1. **Initial Screening**: Human reviewers randomly screen 1 candidate paper.
  2. **Model Training**: Utilize the screened papers as the training set to train the initial predictive model, denoted as M_1.
  3. **Secondary Screening**: Subsequently, select 1 candidate record with the highest predicted relevancy from the pool of 9,999 unscreened candidates.
  4. **Model Update:** $M_2 \to M_3 \to M_4 \to \cdots$

abstrackr

BETA

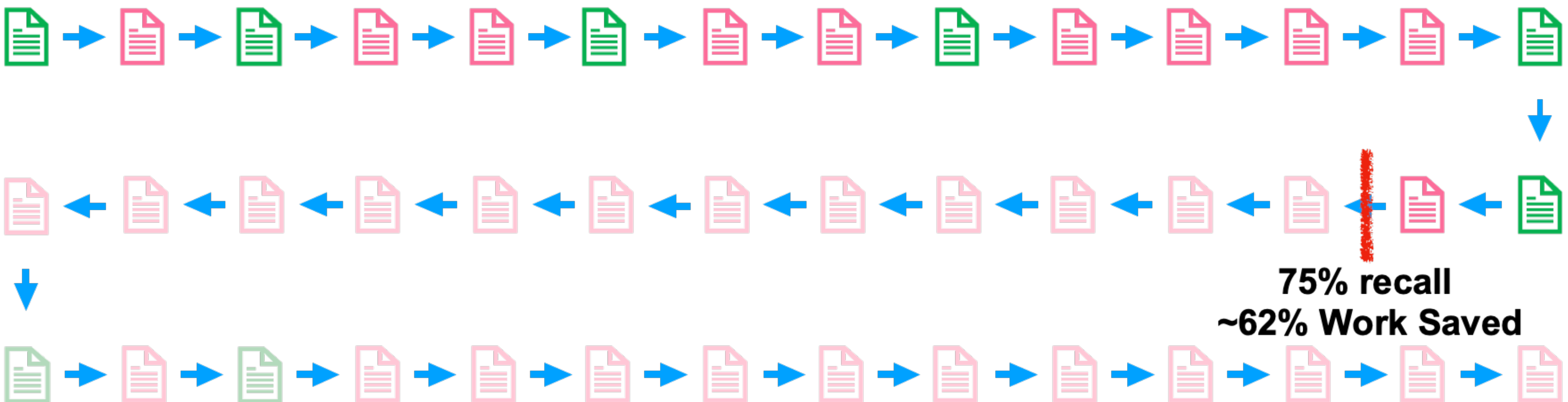
When to stop?



 Relevant Paper (Unscreened)

 Irrelevant Paper (Unscreened)

When to stop?



Relevant Paper (Unscreened)

Relevant Paper (Screened)

Irrelevant Paper (Unscreened)

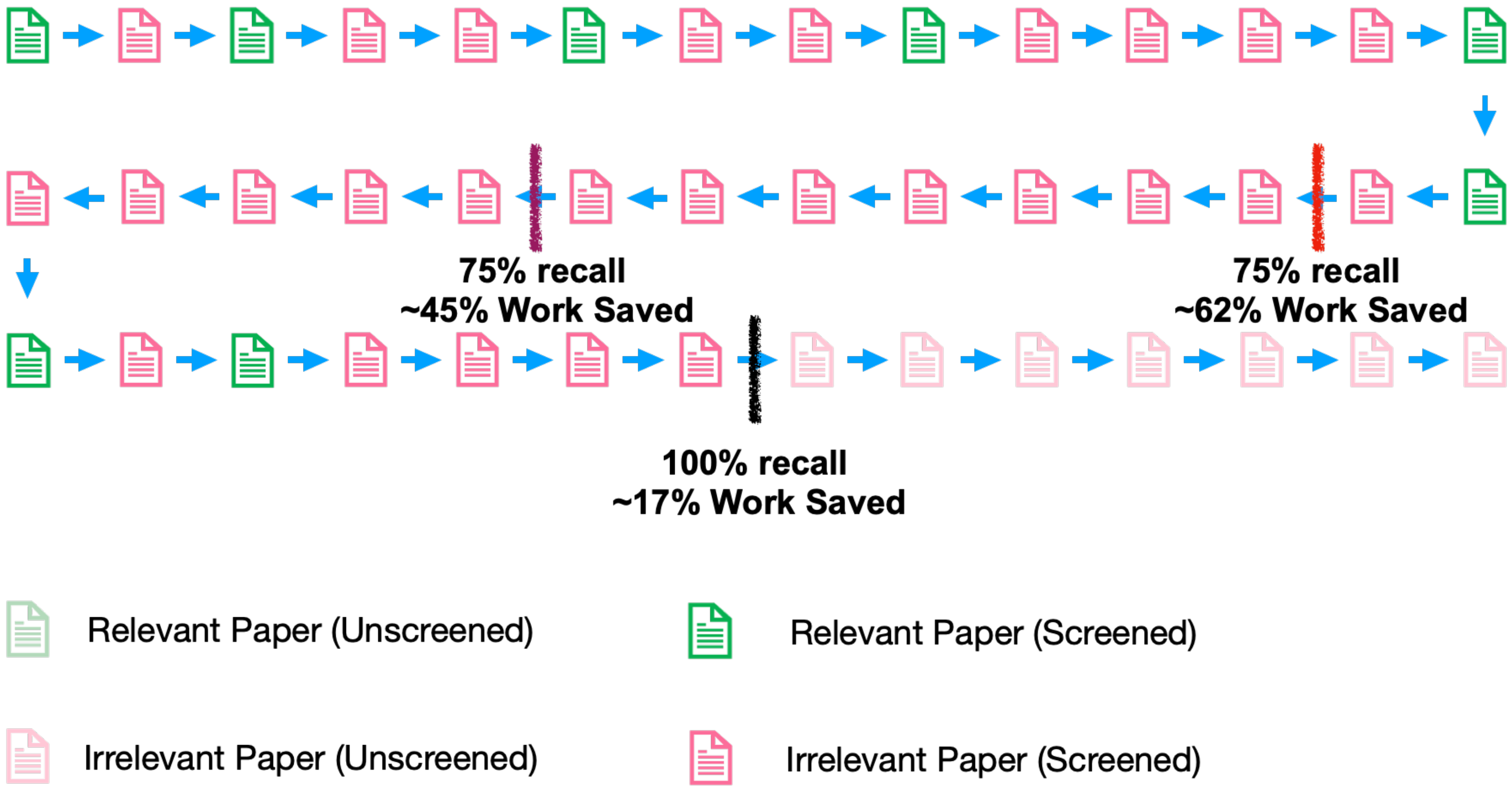
Irrelevant Paper (Screened)

When to stop?

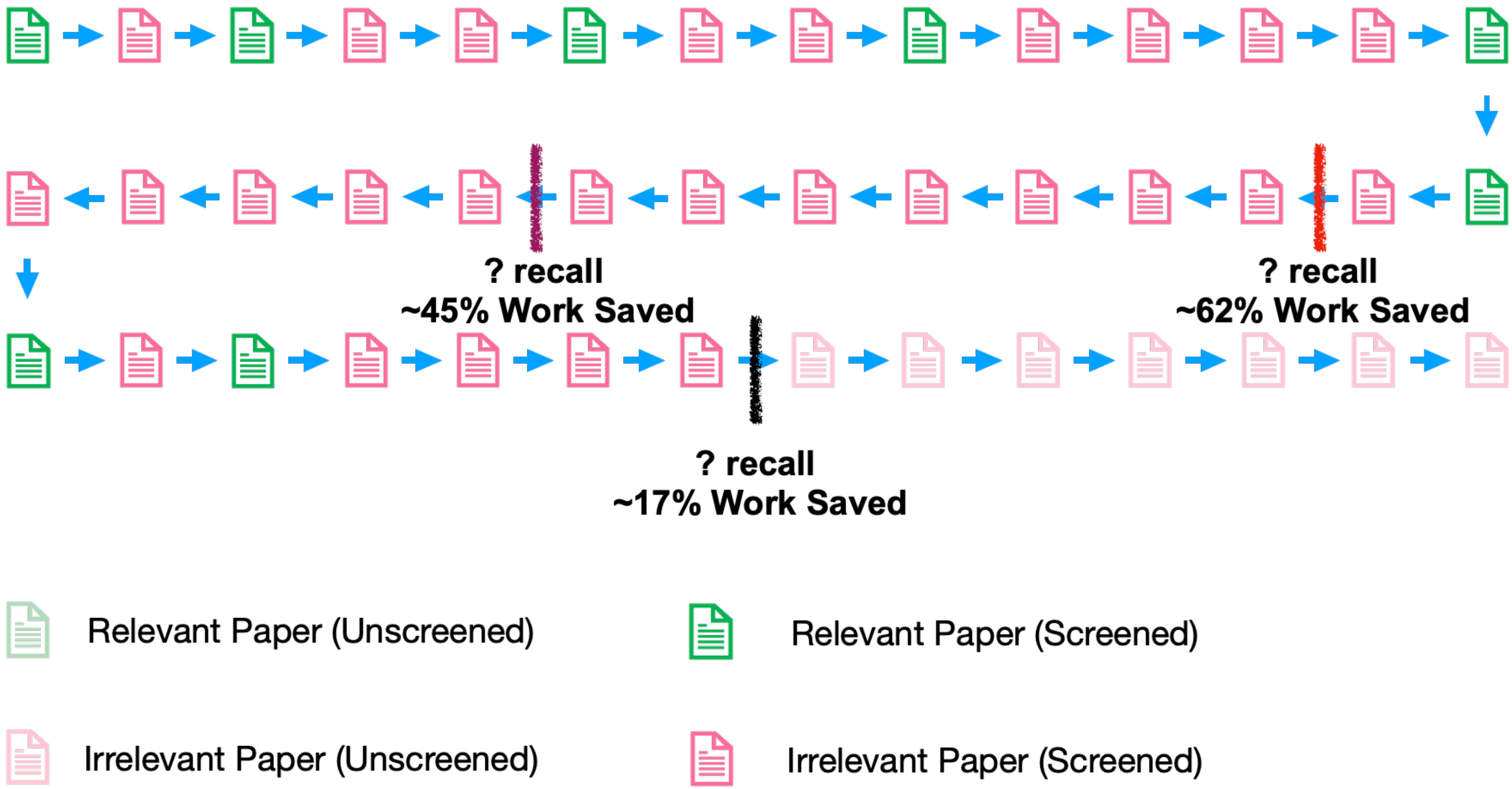

75% recall
~45% Work Saved
75% recall
~62% Work Saved


Relevant Paper (Unscreened)
Relevant Paper (Screened)
Irrelevant Paper (Unscreened)
Irrelevant Paper (Screened)

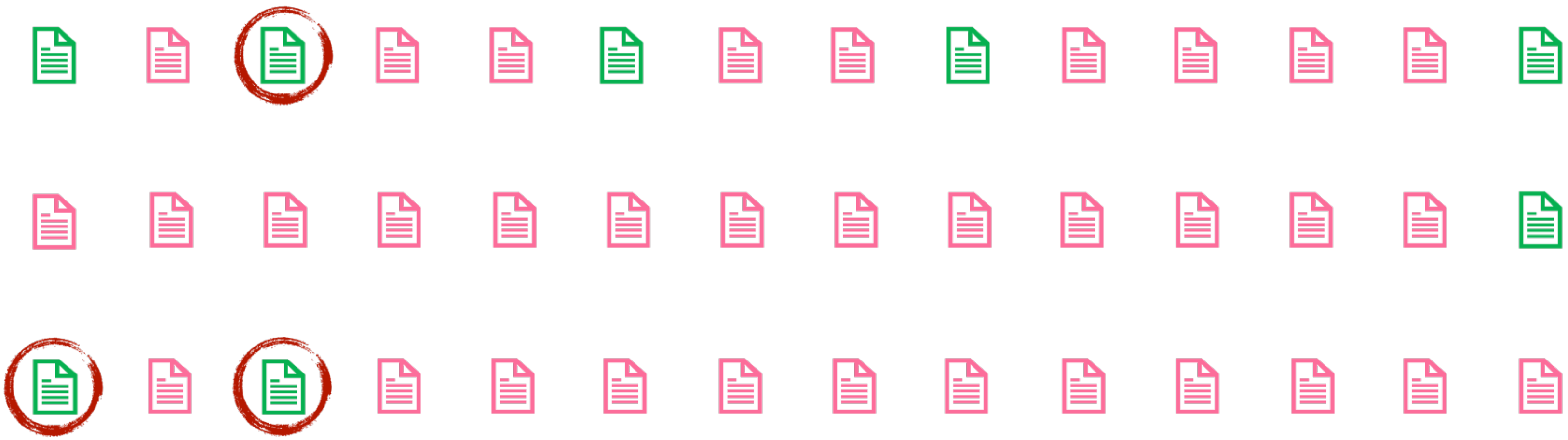
When to stop?


75% recall
~45% Work Saved
75% recall
~62% Work Saved
100% recall
~17% Work Saved
Relevant Paper (Unscreened)
Relevant Paper (Screened)
Irrelevant Paper (Unscreened)
Irrelevant Paper (Screened)

When to stop?

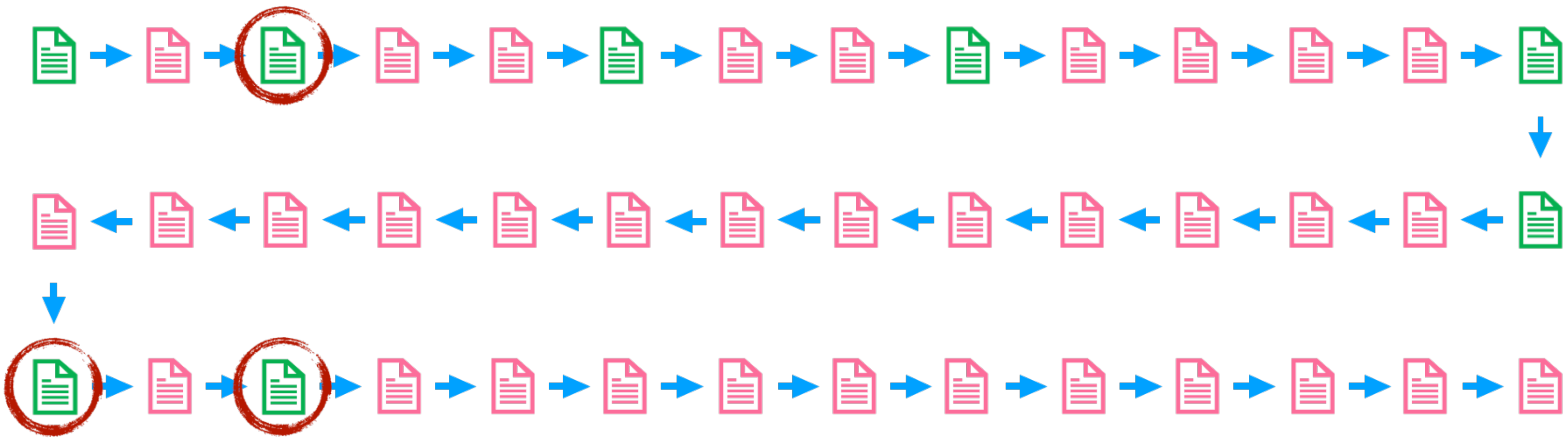

? recall
~45% Work Saved
? recall
~62% Work Saved
? recall
~17% Work Saved
Relevant Paper (Unscreened)
Relevant Paper (Screened)
Irrelevant Paper (Unscreened)
Irrelevant Paper (Screened)

A Resample Algorithm



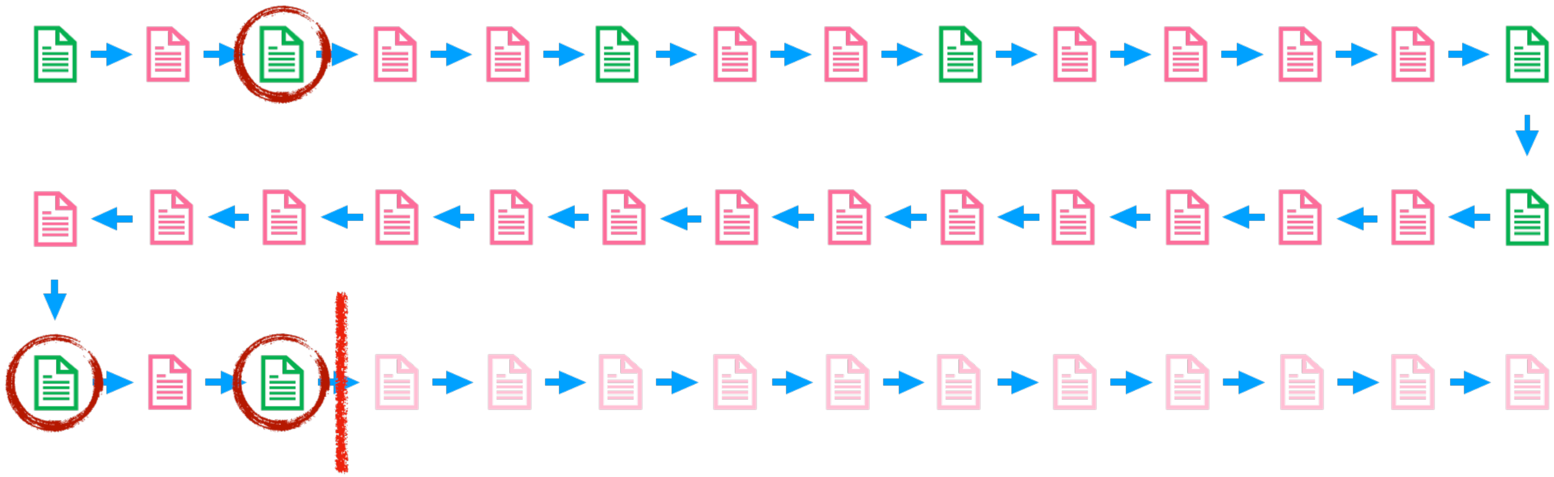
- Draw a random sample of k relevant papers (with replacement).

A Resample Algorithm



- Draw a random sample of k relevant papers (with replacement).
- Rank the papers by predicted relevancy

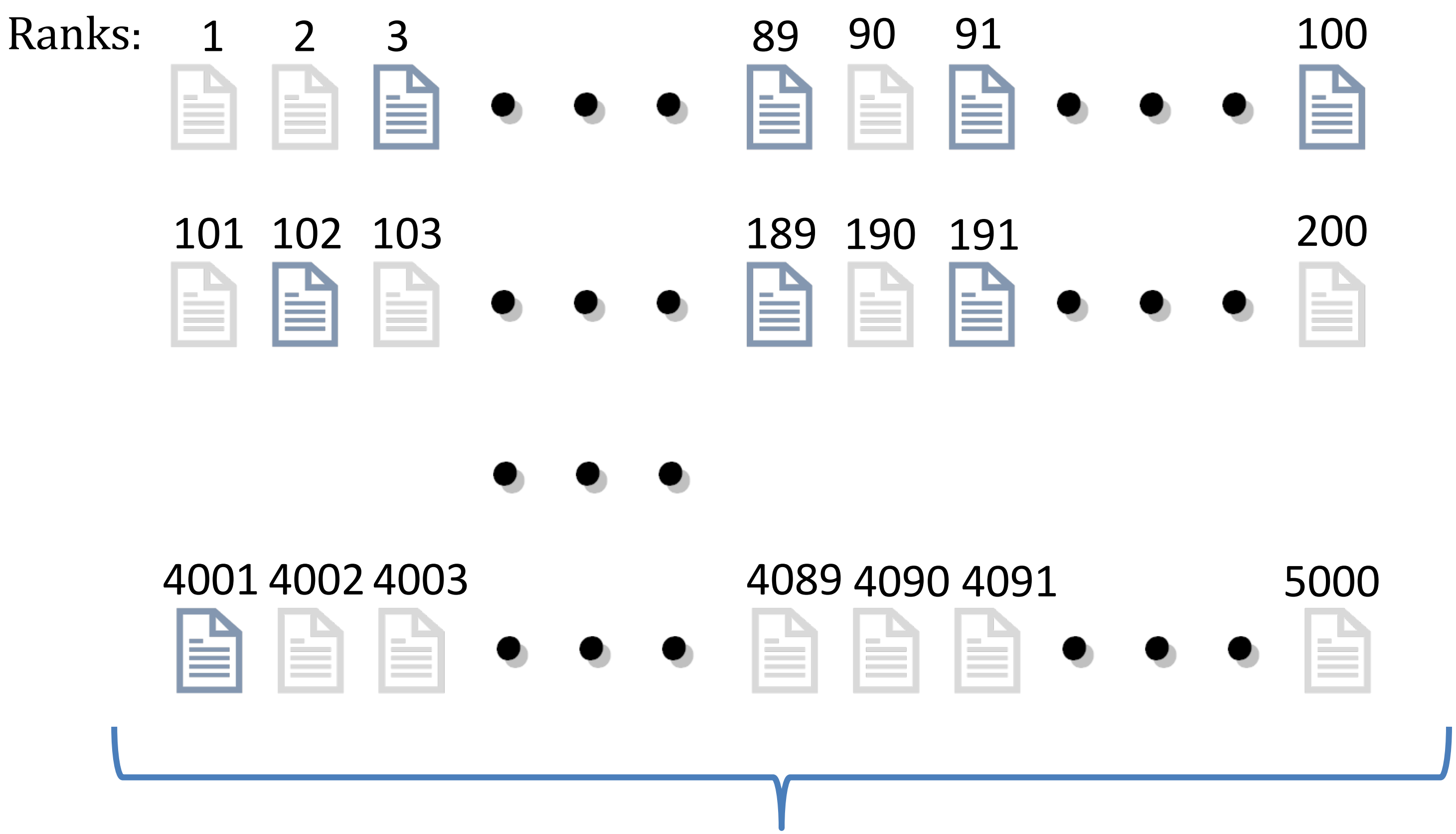
A Resample Algorithm



- Draw a random sample of k relevant papers (with replacement).
- Rank the papers by predicted relevancy.
- Stop screening when all of the sampled papers are identified.
- Probability(Recall $\geq c$) $\geq 1 - c^k$. (When k = 10, Probability(Recall $\geq 80\,\%$) $\geq 89.2\,\%$)

How does the algorithm work?

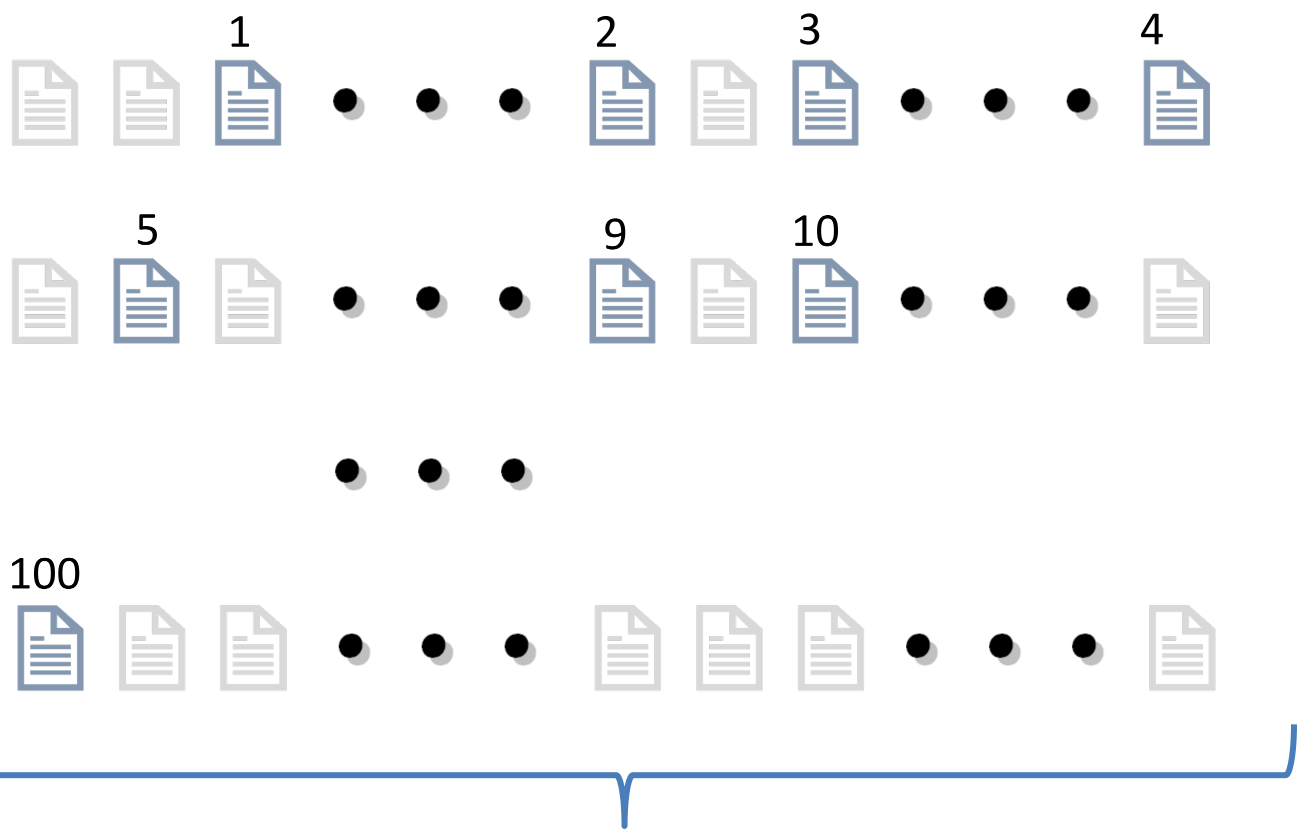
A Deeper Dive Into The Algorithm



100 relevant papers in 5000 candidate papers (N=5000, R =100)

A Deeper Dive Into The Algorithm

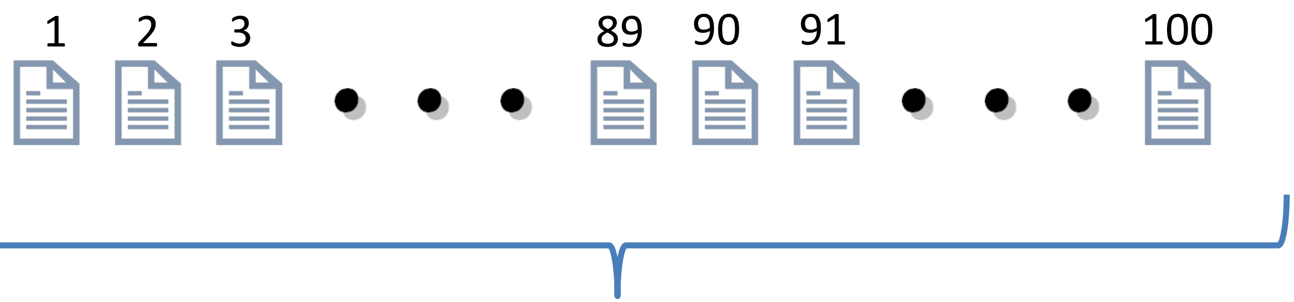
Relative Ranks:



100 relevant papers in 5000 candidate papers (N=5000, R =100)

A Deeper Dive Into The Algorithm

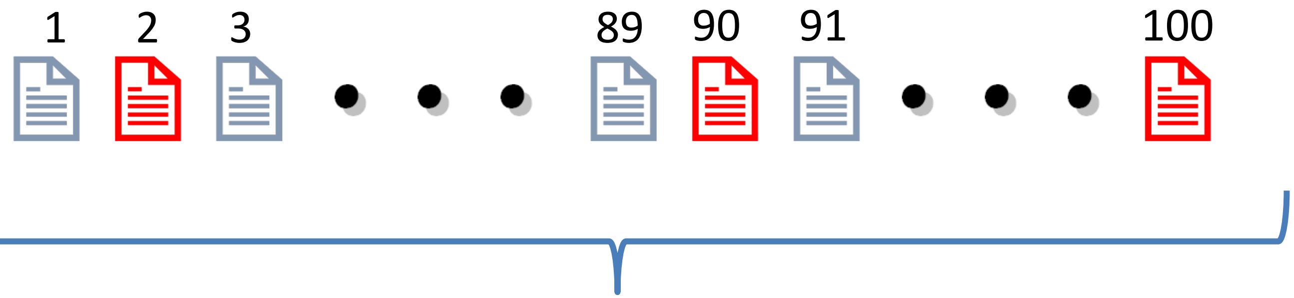
Relative Ranks:



100 relevant papers (R=100)

A Deeper Dive Into The Algorithm

Relative Ranks:

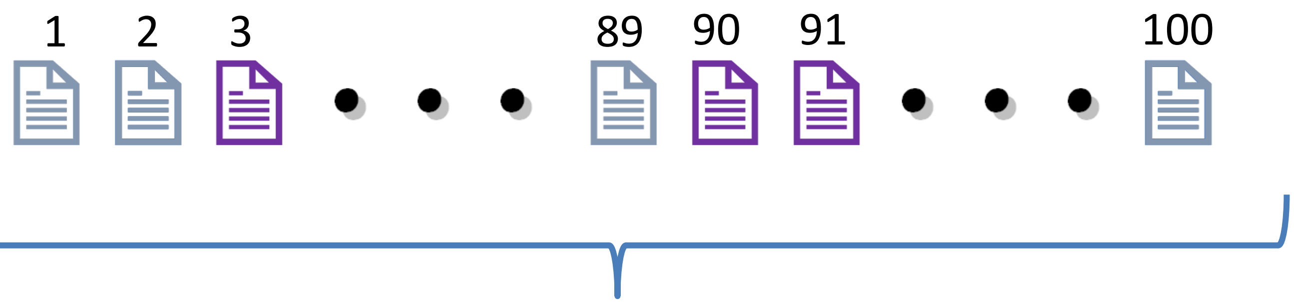


100 relevant papers (R=100)

Random sample of 10 (Target set): 

A Deeper Dive Into The Algorithm

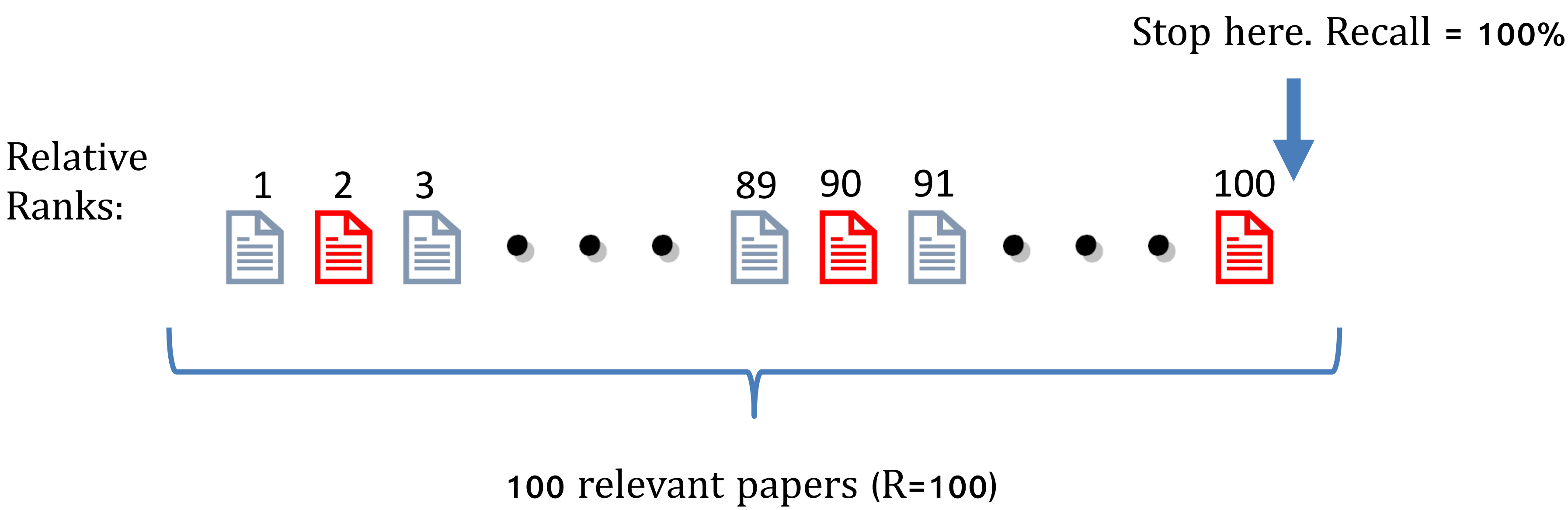
Relative Ranks:



100 relevant papers (R=100)

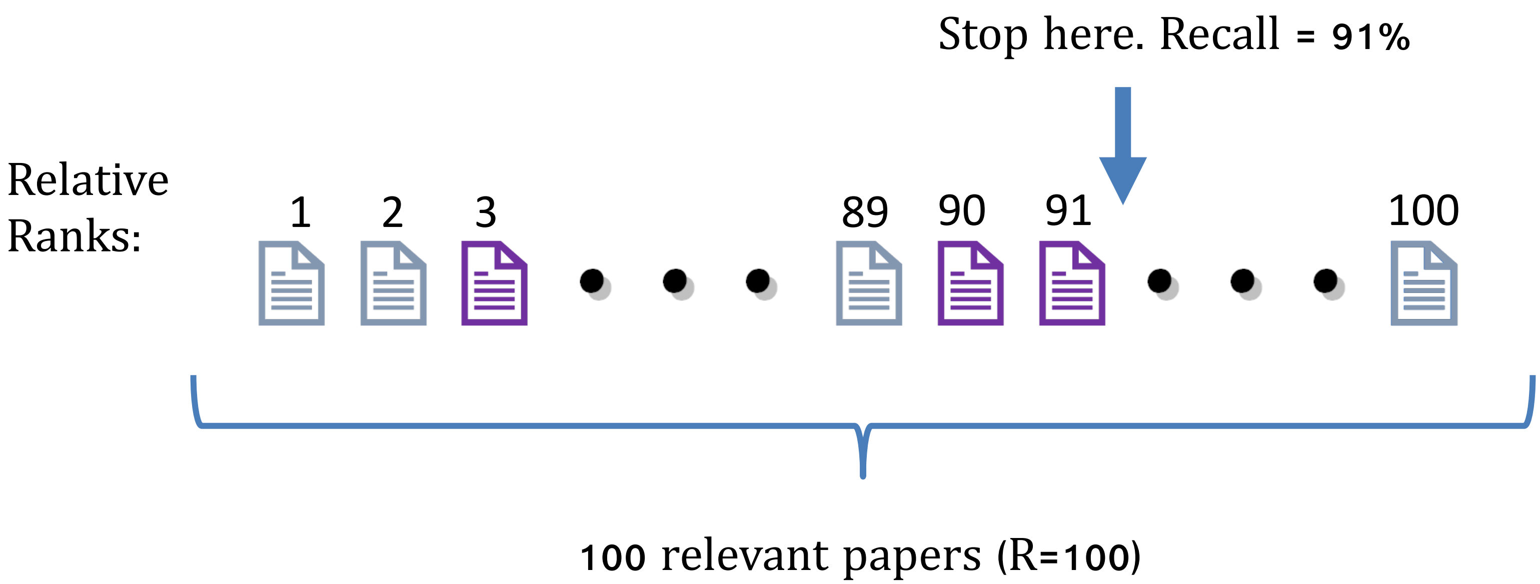
Random sample of 10 (Target set):

A Deeper Dive Into The Algorithm



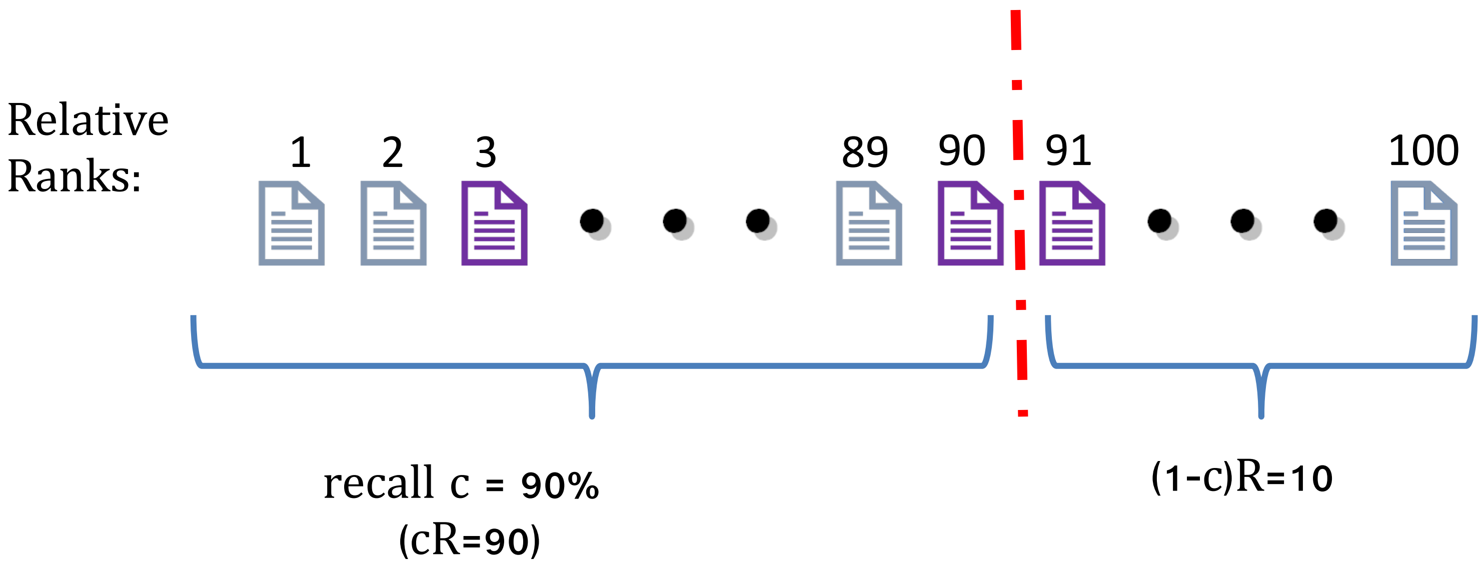
Algorithm: keep screening until all 10 sample papers are found.

A Deeper Dive Into The Algorithm



Algorithm: keep screening until all 10 sample papers are found.

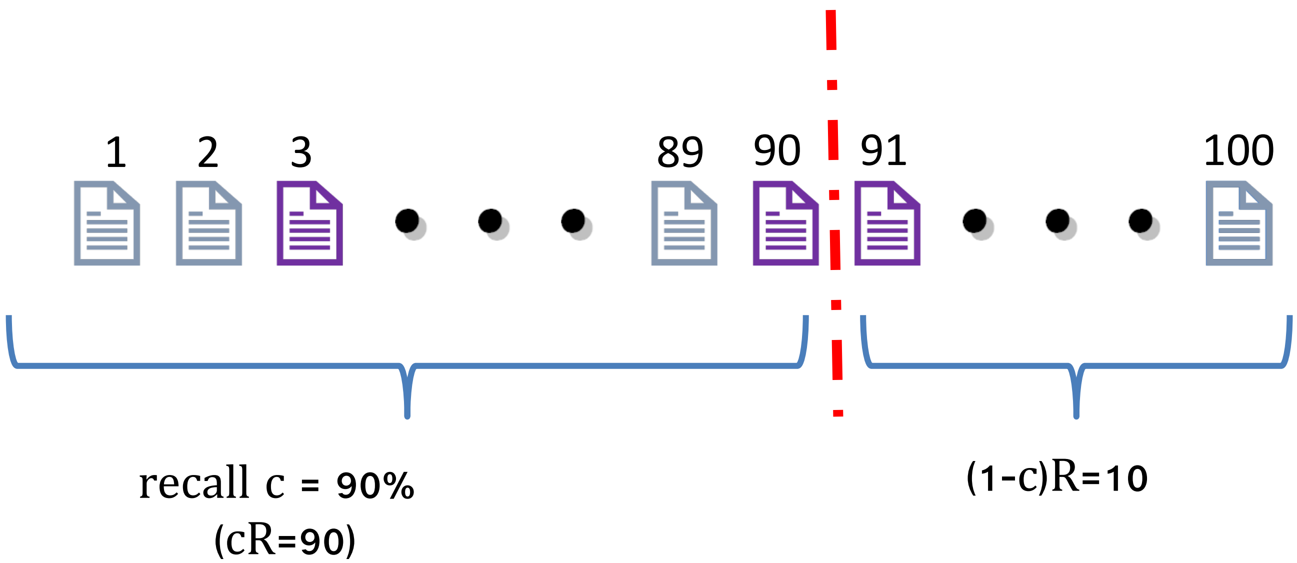
A Deeper Dive Into The Algorithm



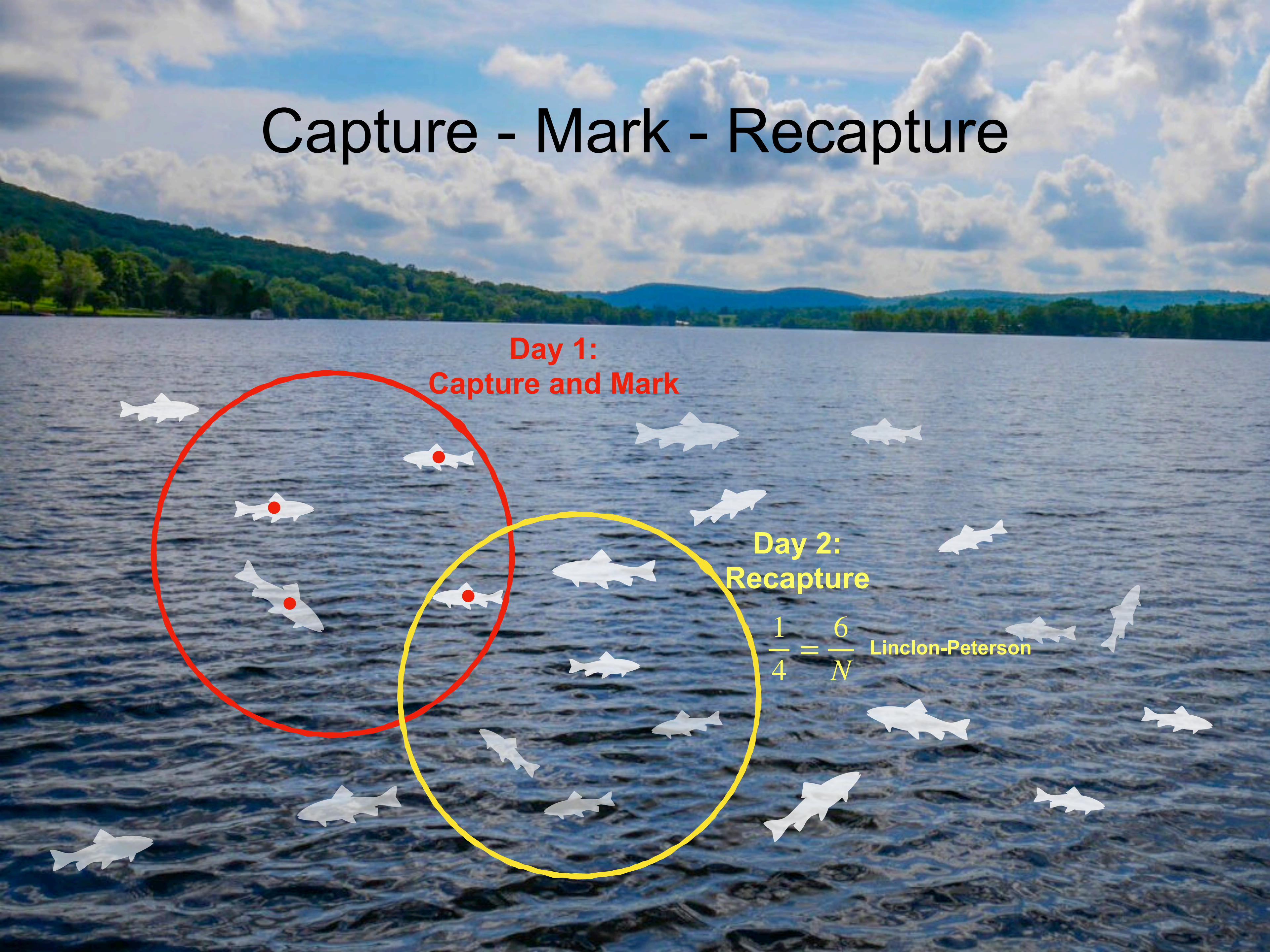
$$
\begin{aligned}
P(recall \le 90\,\%) &= P(All\ 10\ sample\ papers\ are\ the\ top\ 90\ papers) \\
&= 0.9^{10}.
\end{aligned}
$$

A Deeper Dive Into The Algorithm

Relative Ranks:



$$P(recall > 90\ \%) = 1 - P(All\ 10\ sample\ papers\ are\ the\ top\ 90\ papers)$$
$$= 1 - 0.9^{10}.$$



Capture - Mark - Recapture
Day 1:
Capture and Mark
Day 2:
Recapture
$\frac{1}{4} = \frac{6}{N}$
Linclon-Peterson

Discussion

1. This resample algorithm is compatible with any predictive models or AI.
2. A Better predictive model results in greater workload savings.
3. Questions?