Refresh Strategies for Continuous Active Learning

Nimesh Ghelani
Gordon V. Cormack
Mark D. Smucker

University of Waterloo

First International Workshop on Professional Search, 2018

Introduction

Find all or nearly all relevant documents using minimal assessment costs

Introduction

Find all or nearly all relevant documents using minimal assessment costs

High Recall problem

Some problems:

- Legal eDiscovery
- Systematic Review
- Building test collection

Introduction

Technology Assisted Review (TAR):
computer-assisted methods to do eDiscovery

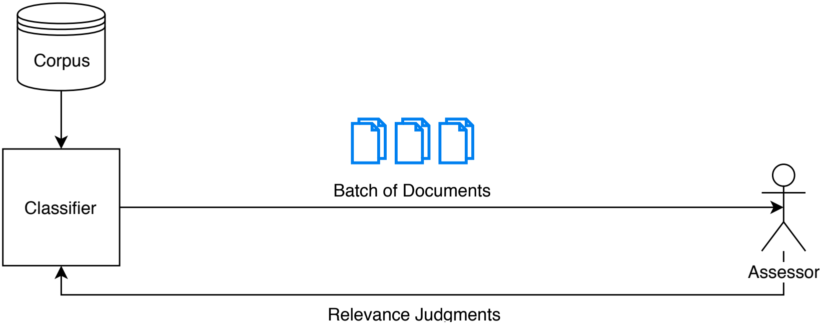
Introduction

Technology Assisted Review (TAR):
computer-assisted methods to do eDiscovery

Continuous Active Learning (CAL):

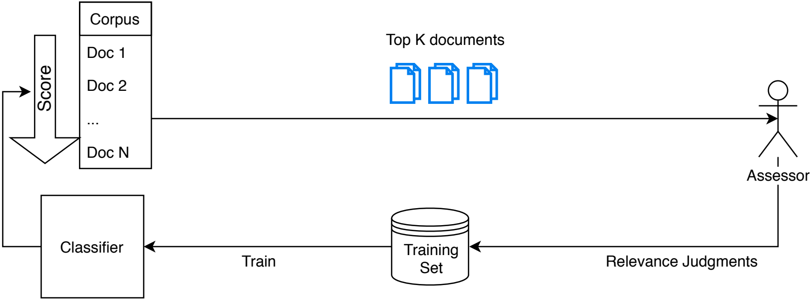
- A TAR protocol
- Human in loop with a machine learning model

Introduction



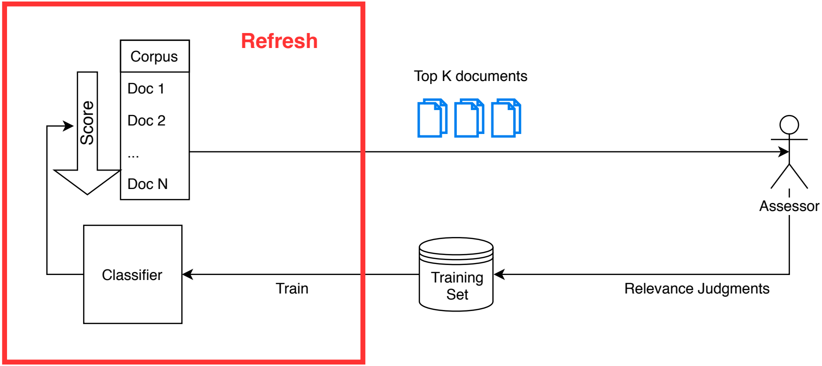
Relevance Feedback Loop

Introduction



Relevance Feedback Loop

Introduction



Relevance Feedback Loop

Introduction

Refresh:

- Use available judgments to build a classifier
- Produce next set of documents to be judged

Introduction

Refresh:

- Use available judgments to build a classifier
- Produce next set of documents to be judged

Refresh Strategy

- When to refresh?
- How to refresh?

Introduction

Refresh:

- Use available judgments to build a classifier
- Produce next set of documents to be judged

Refresh Strategy

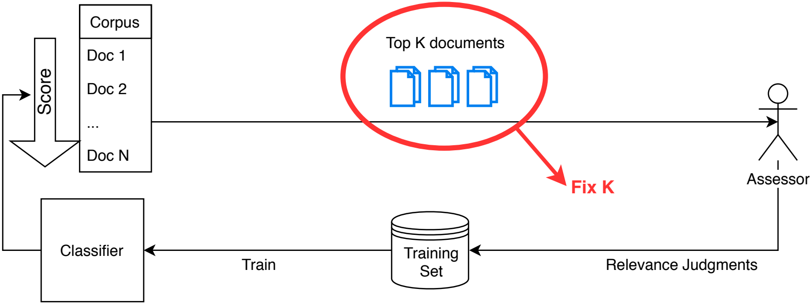
- When to refresh?
- How to refresh?

Objective: Investigate various refresh strategies; their effectiveness and efficiency

Outline

- Refresh Strategies
  - Static batch sizes
  - Partial refresh
  - Precision based
- Results
- Summary

Static Batch Strategy



Static Batch Strategy

BMI Strategy

Used in the Baseline Model Implementation (BMI) at the TREC 2015 and 2016 Total Recall tracks

Train and score all documents every K assessments (K increases exponentially)

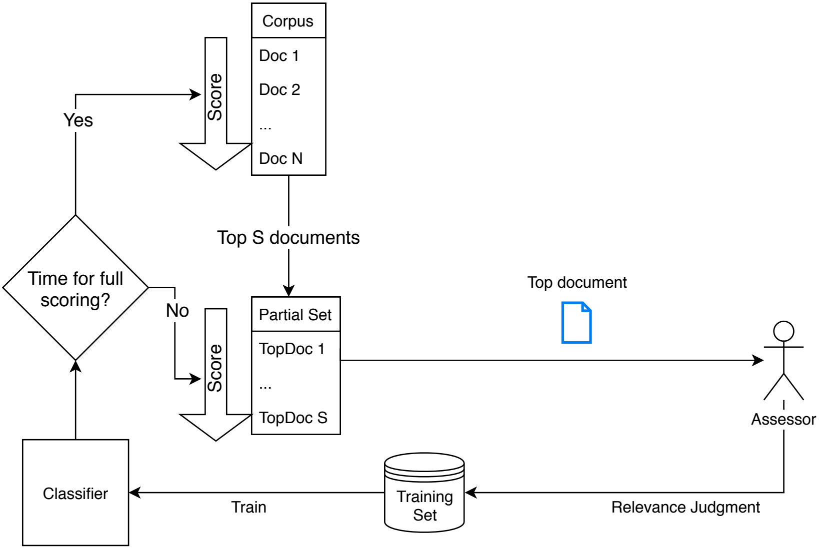
After every refresh, $K \leftarrow K + (K + 9)/10$

Partial Refresh

Perform frequent scoring on a smaller set of data

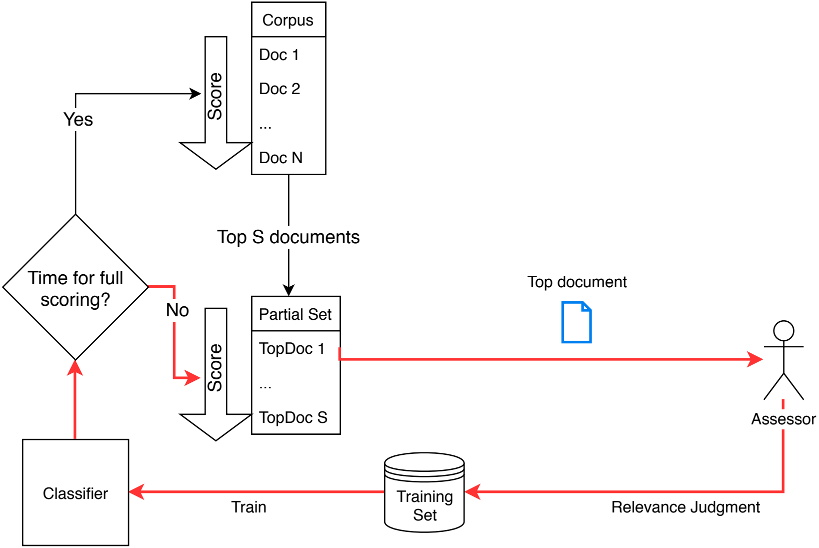
Periodic complete scoring

Partial Refresh Strategy



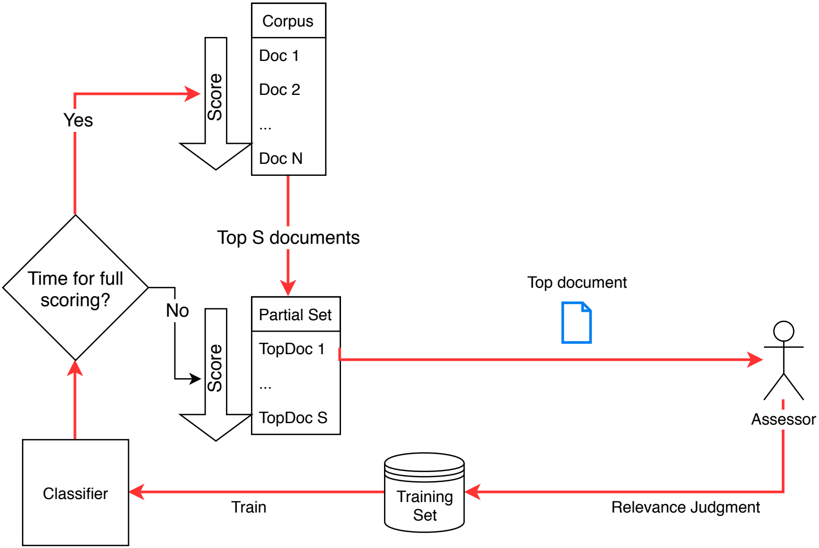
Partial Refresh Strategy

Partial Refresh Strategy



After every judgment

Partial Refresh Strategy



After every K judgments

Precision Based Refreshing

Refresh when "output quality" falls below some threshold

Precision Based Refreshing

Refresh when "output quality" falls below some threshold

Problem: Defining "output quality"

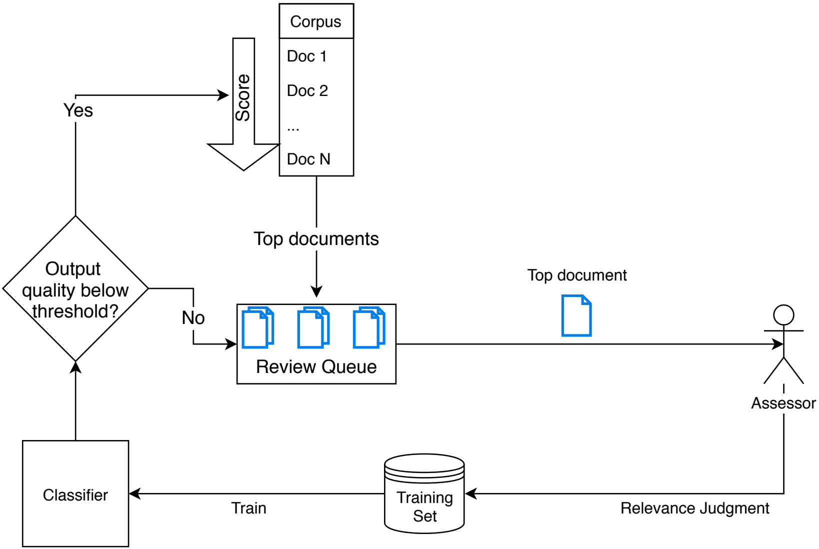
Precision Based Refreshing

Refresh when “output quality” falls below some threshold

Problem: Defining “output quality”

Refresh when the precision of the last m assessed documents fall below p

Precision Based Refreshing



Dataset and Experiment

- `Athome1` test collection from the TREC 2015 Total Recall track
- Around 290k documents; 10 topics
- Implementation of CAL from HiCAL[1]
- Recall at certain effort
  - Normalized Effort = No. of Assessments / Total no. of relevant documents
- Simulation running time

[1] `http://hical.github.io`

Results

BMI vs Static Batch Refreshing

Strategy	Avg. Recall @(E_{norm}=1)	Avg. Recall @(E_{norm}=2)	E_{norm} for 75% recall	Running Time (in min)
`bmi`	0.715	0.905	1.128	0.22
`static(k=1)`	0.750	0.926	1.021	49.29
`static(k=100)`	0.704	0.887	1.167	0.47

`bmi`: exponentially increasing batch size

`static(k)`: fixed batch size of `k`

Results

Partial Refresh Strategy

Strategy	**Avg. Recall @(E_{norm}=1)**	**Avg. Recall @(E_{norm}=2)**	**E_{norm} for 75% recall**	**Running Time (in min)**
`static(k=1)`	0.750	0.926	1.021	49.29
`partial(k=10,s=1000)`	0.753	0.926	1.008	40.92
`partial(k=100,s=1000)`	0.754	0.922	1.013	39.57
`partial(k=100,s=5000)`	0.756	0.921	1.016	40.70
`partial(k=500,s=1000)`	0.700	0.815	1.324	38.63

`static(k)`: fixed batch size of `k`
`partial(k,s)`: complete scoring after `k` judgments, partial set size of `s` documents

Results

Partial Refresh Strategy

Strategy	**Avg. Recall @(E_{norm}=1)**	**Avg. Recall @(E_{norm}=2)**	**E_{norm} for 75% recall**	**Scoring Time (in min)**
`static(k=1)`	0.750	0.926	1.021	23.88
`partial(k=10,s=1000)`	0.753	0.926	1.008	2.25
`partial(k=100,s=1000)`	0.754	0.922	1.013	0.39
`partial(k=100,s=5000)`	0.756	0.921	1.016	0.82
`partial(k=500,s=1000)`	0.700	0.815	1.324	0.17

`static(k)`: fixed batch size of `k`
`partial(k,s)`: complete scoring after `k` judgments, partial set size of `s` documents

Results

Precision Based Refreshing

Strategy	Avg. Recall @(E_{norm}=1)	Avg. Recall @(E_{norm}=2)	E_{norm} for 75% recall	Running Time (in min)
static(k=1)	0.750	0.926	1.021	49.29
precision(m=25,p=0.4)	0.698	0.915	1.129	35.68
precision(m=25,p=0.6)	0.735	0.923	1.059	40.20
precision(m=25,p=0.8)	0.750	0.926	1.024	44.64
precision(m=25,p=1.0)	0.752	0.926	1.014	47.41

`static(k)`: fixed batch size of `k`

`precision(m,p)`: perform refresh when precision of last `m` documents fall below `p`

Summary

- Frequent refreshing helps achieving higher recall using lesser assessment effort
- Static batch size of 1 performs great but is computationally expensive
  - Practical for reasonably sized datasets and modern hardware
  - Various alternative strategies can achieve similar effectiveness with reduced computations

Summary

- Frequent refreshing helps achieving higher recall using lesser assessment effort
- Static batch size of 1 performs great but is computationally expensive
  - Practical for reasonably sized datasets and modern hardware
  - Various alternative strategies can achieve similar effectiveness with reduced computations

Questions?