Proceedings Paper:

Bin Hezam, R. and Stevenson, R.M. orcid.org/0000-0002-9483-6006 (2024) RLStop: a reinforcement learning stopping method for TAR. In: SIGIR '24: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. The 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 14-18 Jul 2024, Washington D.C., USA. ACM Digital Library , pp. 2604-2608. ISBN 9798400704314

https://doi.org/10.1145/3626772.3657911

RLStop: A Reinforcement Learning Stopping Method for TAR

Reem Bin-Hezam[1,2]
rybinhezam@pnu.edu.sa
[1]Department of Information Systems
College of Computer and Information Sciences
Princess Nourah Bint Abdulrahman University
Riyadh, Saudi Arabia

Mark Stevenson[2]
mark.stevenson@sheffield.ac.uk
[2]Department of Computer Science
Faculty of Engineering
University of Sheffield
Sheffield, United Kingdom

ABSTRACT

We present RLStop, a novel Technology Assisted Review (TAR) stopping rule based on reinforcement learning that helps minimise the number of documents that need to be manually reviewed within TAR applications. RLStop is trained on example rankings using a reward function to identify the optimal point to stop examining documents. Experiments at a range of target recall levels on multiple benchmark datasets (CLEF e-Health, TREC Total Recall, and Reuters RCV1) demonstrated that RLStop substantially reduces the workload required to screen a document collection for relevance. RLStop outperforms a wide range of alternative approaches, achieving performance close to the maximum possible for the task under some circumstances.

CCS CONCEPTS

• **Information systems** → **Retrieval effectiveness**; **Retrieval efficiency**.

KEYWORDS

Reinforcement Learning, Deep Reinforcement Learning, Technology Assisted Review, TAR, Stopping Methods

ACM Reference Format:
Reem Bin-Hezam[1,2] and Mark Stevenson[2]. 2024. RLStop: A Reinforcement Learning Stopping Method for TAR. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626772.3657911

1 INTRODUCTION

Identifying all, or a significant proportion of, the relevant documents in a collection has applications in multiple areas including identification of scientific studies for inclusion in systematic reviews [13, 16–18], satisfying legal disclosure requirements [2, 12, 23], social media content moderation [38] and test collection development [22]. These problems often involve large collections where manually reviewing all documents would be prohibitively time-consuming. Technology Assisted Review (TAR) develops techniques to support these document review processes, including stopping rules which help reviewers to decide when to stop assessing documents, thereby reducing the effort required to screen a collection for relevance.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657911

TAR stopping rules aim to identify when a desired level of recall (the *target recall*) has been reached during document review, while also minimising the number of documents examined. The problem is challenging since these two objectives are in opposition; increasing the number of documents examined provides more information about whether the target has been reached.

A common approach involves estimating the total number of relevant documents in the collection and therefore whether the target recall has been reached, e.g. [4, 9, 15, 25, 31, 36, 39]. Alternative approaches include randomly sampling documents until a sufficient number of relevant ones have been discovered to guarantee that the target recall has been reached [7, 37] and observing the rate at which relevant documents occur within a ranking until it drops below a pre-defined threshold [7].

Existing methods suffer from a number of limitations. Several approaches carry out repeated statistical testing to determine whether the target recall has been achieved, e.g. [4, 15, 21, 31, 36], but this type of sequential testing is statistically invalid [37]. Lewis et al. [19] avoided this problem using techniques from manufacturing quality control [11] but their approach often required more documents to be reviewed than alternative methods [32]. Some methods, e.g. [7, 37], fail to take account of the fact that standard TAR workflows [6, 9, 21] are highly effective at prioritising relevant documents, meaning that their distribution within the ranking is non-uniform. Exploiting this fact reduces the number of documents that need to be examined before a stopping decision is made [9, 21, 32]. However, existing approaches to modelling the distribution of relevant documents rely on a particular rate function (e.g. power law [42] or AP Prior [1]) but the choice of rate function is a modelling decision and may not be appropriate in all circumstances.

Approaches to TAR stopping essentially involve repeated decisions to either stop or examine more documents. Reinforcement learning (RL) is designed for such sequential decision-making scenarios and has been widely applied within Information Retrieval, e.g. [26, 27, 29, 40, 41]. However it has not previously been applied to TAR stopping. This paper proposes a novel TAR stopping method based on RL. This approach does not rely on invalid statistical assumptions and is able to model rankings to make informed decisions about when to stop. Experiments using a range of target recall levels on multiple benchmark datasets demonstrate that the proposed method is able to identify suitable stopping points and performs well in comparison with several previously reported approaches.

The contributions of this paper are: (1) introduces a novel approach to the TAR stopping problem that makes use of RL, (2) evaluates the proposed algorithm using a range of collections commonly used for TAR problems, and (3) demonstrates that the proposed approach effectively identifies an appropriate stopping point and outperforms a wide range of alternative methods, often substantially.[1]

2 APPROACH

RL is a decision-making method in which an agent aims to maximise the reward obtained from interacting with an environment and making sequential decisions through trial-and-search [33]. The agent's actions are guided by a learned policy (π) which maps states in the environment to actions.

RL is applied to the TAR stopping problem by considering an agent that examines a ranked list of documents with the aim of stopping when a predefined target recall has been achieved. The agent examines the ranking sequentially, starting with the highest ranked documents and working down the ranking. For efficiency, documents are examined in batches with relevance judgements obtained for the entire batch simultaneously. After a batch of documents has been examined the agent can either stop examining documents (if it judges that the target recall has been achieved) or continue to the next batch of documents (if not). This approach is similar to previous approaches of the stopping problem in which a ranked list of documents is examined sequentially (e.g. [10, 14, 32]), although the stopping decision is made by the agent (following a policy) rather than according to some alternative criteria.

2.1 RLStop

This section describes how RL is applied to the stopping problem by outlining how the key elements of an RL system are implemented within it.

State Space: A ranking is split into B fixed size batches containing $\frac{N}{B}$ documents for a collection of N documents. The agent examines batches sequentially and obtains relevance judgements for all documents in the batch simultaneously. The initial state for each ranking, S_1, occurs when the first batch (but none of the subsequent batches) has been explored. Additional batches are examined in subsequent states, i.e. in the nth state, S_n, the first n batches have been examined. The final state, S_B, represents the situation in which the entire ranking has been examined. If the agent reaches this state, it will always stop here since all documents in the ranking have been examined.

States are represented by a fixed size vector of length B in which each element represents a batch. For batches that have been examined the corresponding element shows the number of relevant documents within the batch, while elements corresponding to unexamined batches are given a dummy value (-1).

Action Space: At each point in the ranking, the agent has a choice between two discrete actions: STOP and CONTINUE. The first action is chosen when the agent (informed by the policy) judges that the target recall has been reached. The stopping point returned is the end of the last batch that has been examined so far. If the agent does not stop it continues to examine the ranking, i.e. moves from state S_i to S_{i+1}.

Reward function: A reward function, $R(S_i)$, assigns a score to S_i indicating its attractiveness for the agent. The reward function is used to train the policy and designing a suitable one is therefore important. A suitable function should: 1) encourage the agent to continue examining documents until the target recall has been reached, 2) discourage further examination after it has been reached and 3) be independent from each topic's specific properties (e.g. total number of documents, ranking shape). In addition, continuous functions are more straightforward for RL algorithms to optimise.

The following function achieves these goals:

$$R(S_i) = \begin{cases} 1 - \dfrac{i}{T} & \text{if } i \leq T \\ -\dfrac{i-T}{B-T} & \text{if } i > T \end{cases} \tag{1}$$

where (as above) B is the number of batches into which the ranking is split, i is the index of the current state (i.e. ith batch) and T is the batch at which the target recall is reached. (Note that while the value of T is known while the RL algorithm is being trained, it is not known when it is applied.) The function assigns a positive reward when the current state is at, or below, the target recall and a negative reward when it has been exceeded.

The cumulative reward for an RL episode (i.e. examining a ranking until a stopping decision is made) is the sum of rewards for all positions examined by the agent from S_1 to the stopping step S_s, i.e. $\sum_{i=1}^{s} R(S_i)$

Policy: RL aims to learn a policy, $\pi(s, a)$, that maximises the expected cumulative reward obtained by taking an action (a) given a state (s). Since the state space for our problem is high-dimensional, a neural network is a choice for the policy. The policy is a feed-forward network consisting of an input layer of length B, representing the current state, two 64-node hidden layers and a binary output layer indicating the chosen action, which is converted to a probability distribution over actions by a softmax activation function.

RL Algorithm: RLStop uses Proximal Policy Optimization (PPO) [30], a policy gradient approach to RL. PPO is an actor-critic RL algorithm that combines policy-based (actor) and value-based (critic) RL, where the actor decides the actions, and the critic evaluates them. It is based on the REINFORCE [35] algorithm but with several enhancements, including the employment of parallel actors running independently by collecting trajectories of different environments simultaneously which allows a policy to be trained using multiple rankings. PPO is also more sample-efficient than some alternative methods, such as DQN [24], thereby reducing the amount of data required to learn effective policies.

Implementation: The Stable-Baseline3 library [28] was used to implement RLStop. The RL environment was created using the Gymnasium library [34] which allows multiple environments to be stacked, thereby allowing simultaneous training on multiple topics to ensure the agent is as general as possible. The number of batches, B, was set to 100.

3 EXPERIMENTS

Experiments were carried out on multiple datasets and evaluation metrics.

3.1 Datasets

Performance was evaluated on six datasets widely used in previous TAR work and representing multiple domains. All datasets are

[1] Code for the experiments available from https://github.com/ReemBinHezam/RLStop/

highly imbalanced, with a very low percentage of relevant documents per topic.

CLEF 2017/2018/2019 [16–18]: Collections of systematic reviews produced for the Conference and Labs of the Evaluation Forum (CLEF) 2017, 2018, and 2019 e-Health lab Task 2: Technology-Assisted Reviews in Empirical Medicine. The CLEF 2017 dataset contains 42 reviews; CLEF 2018 contains 30 and CLEF 2019 contains 31. RLStop was trained using the 12 reviews provided with the CLEF 2017 dataset.

TREC Total Recall (TR) [12] A collection of 290,099 emails associated with Jeb Bush's eight-year tenure as Governor of Florida (athome4). The collection contains 34 topics. RLStop was trained using the athome1 dataset consisting of 10 topics.

RCV1 [20] A collection of Reuters news articles labelled with subject categories. Following [36, 37], 45 categories were used to represent a range of topics, and the collections were downsampled to 20%. RLStop was trained using the remaining RCV1 topics, excluding those already included in the test set.

Each collection was ranked with the use of a reference implementation [21] of AutoTAR [5], a greedy Active Learning approach representing state-of-the-art performance on total recall tasks widely used for TAR experiments. The use of AutoTAR rankings allows direct comparison between RLStop and alternative approaches used as baselines in previous work [21, 32].

3.2 Evaluation Measures

A wide range of metrics have been used to evaluate TAR stopping methods, e.g. [21, 36]. These essentially measure the method's success of meeting the two objectives of stopping algorithms: (1) reach the target recall and (2) examine as few documents as possible. Two metrics were used to capture these objectives. **Recall**: the proportion of relevant documents identified by the method. **Cost**: percentage of documents examined. Results for these metrics are reported as average scores for all topics in each collection.

An additional metric captures elements of both objectives and is used to quantify the variation in performance across topics contained within a collection. **Excess**: proportion of documents examined after the target recall has been reached or that needs to be examined reach it [3]. Excess is defined as follows:

$$excess = \frac{cost(method) - cost(optimal)}{1 - cost(optimal)} \tag{2}$$

where $cost(method)$ and $cost(optimal)$ are the cost of the method being evaluated and stopping at the optimal point (i.e. oracle). An excess of 0 indicates that the optimal stopping point for the target recall has been reached, a positive score indicates that more documents than absolutely necessary were examined (i.e. the target was overshot) and a negative indicates that the target was not reached (i.e. undershot).

Metrics were calculated using the `tar_eval` open-source evaluation script.[2]

3.3 Training and Hyper-parameter Tuning

RLStop models were trained for 100,000 timesteps on the each target recall using each dataset's training split. PPO hyper-parameters were set via a grid search on the CLEF 2017 training dataset and the following values chosen: batch size = 100, number of steps (used to collect trajectories before each policy update rollout) = 100, learning rate = 0.0001, number of epochs = 8, entropy coefficient (which encourages policy exploration) = 0.1, discount factor = 0.99 and KL coefficient (which controls the clipping range) = 0.2.

3.4 Baselines and Oracle

RLStop was compared against a range of alternative approaches used as baselines in previous work [21, 32]. The target method (TM) [8] randomly samples documents until a set of predefined target set is identified with results reported for the original method [8] and two extensions: TM-adapted [32] and QBCB [19]. SCAL [9] and AutoStop [21] estimate the number of relevant documents by sampling across the entire ranking. SD-training/SD-sampling [14] are score distribution methods. The knee method [7] identifies the inflection point in the gain curve. IP-H [32] uses a counting process to estimate the number of relevant documents.

Baselines are computed using reference implementations from previous work [21, 32] where possible. Otherwise, previously reported results are used and are directly comparable since they are also based on AutoTAR rankings. However, some baselines are not available for RCV1 since we were unable to run the reference code and results have not been provided in previous work.

Performance was also compared against an Oracle method (OR) which examines documents in ranking order and stops when the target recall level has been achieved (or exceeded). The oracle represents the behaviour of an ideal stopping method but is not useful in practise since it requires full information about the ranking.[3]

4 RESULTS AND ANALYSIS

Figure 1 shows the recall and cost for RLStop and alternative methods for each data set at multiple target recall levels: {1.0, 0.9, 0.8}. Performance of RLStop (denoted by blue hexagon) is often close to the optimal oracle results (e.g. sub-figures (b), (c), (f), (g), (j) and (n)) and is also Pareto optimal[4] in the majority of cases. RLStop is not Parteo optimal in two cases (sub-figures (n) and (o)) but is very close to the Pareto frontier both times. Other baselines that are commonly Pareto optimal include IP-H and the Knee method (cyan star and a green square, respectively). However, both methods tend to overshoot the target more than RLStop. Other baselines are substantially more costly and either undershoot or overshoot the target more frequently than RLStop.

Although RLStop tends to follow the target recall it does not do so exactly and tends to overshoot for lower target recalls (e.g. 0.8). This overshooting is particularly pronounced for the TR dataset where the target recall is often reached very quickly (as demonstrated by the oracle performance). For some topics overshooting is caused by the ranking being divided into fixed batches (1% of the ranking) where examining only a single batch (i.e. the earliest possible stopping point) overshoots the target recall. Increasing the number of batches may be a potential solution to this problem.

[2]https://github.com/CLEF-TAR/tar

[3]Note that the recall achieved by the oracle is higher than the target recall when it is not possible to achieve the target recall exactly, e.g. given a target recall of 0.8 and collection containing 9 relevant documents, the oracle will stop after 8 relevant documents have been found, i.e. recall 0.89.

[4]No other approach achieves the same, or greater, recall with lower cost.

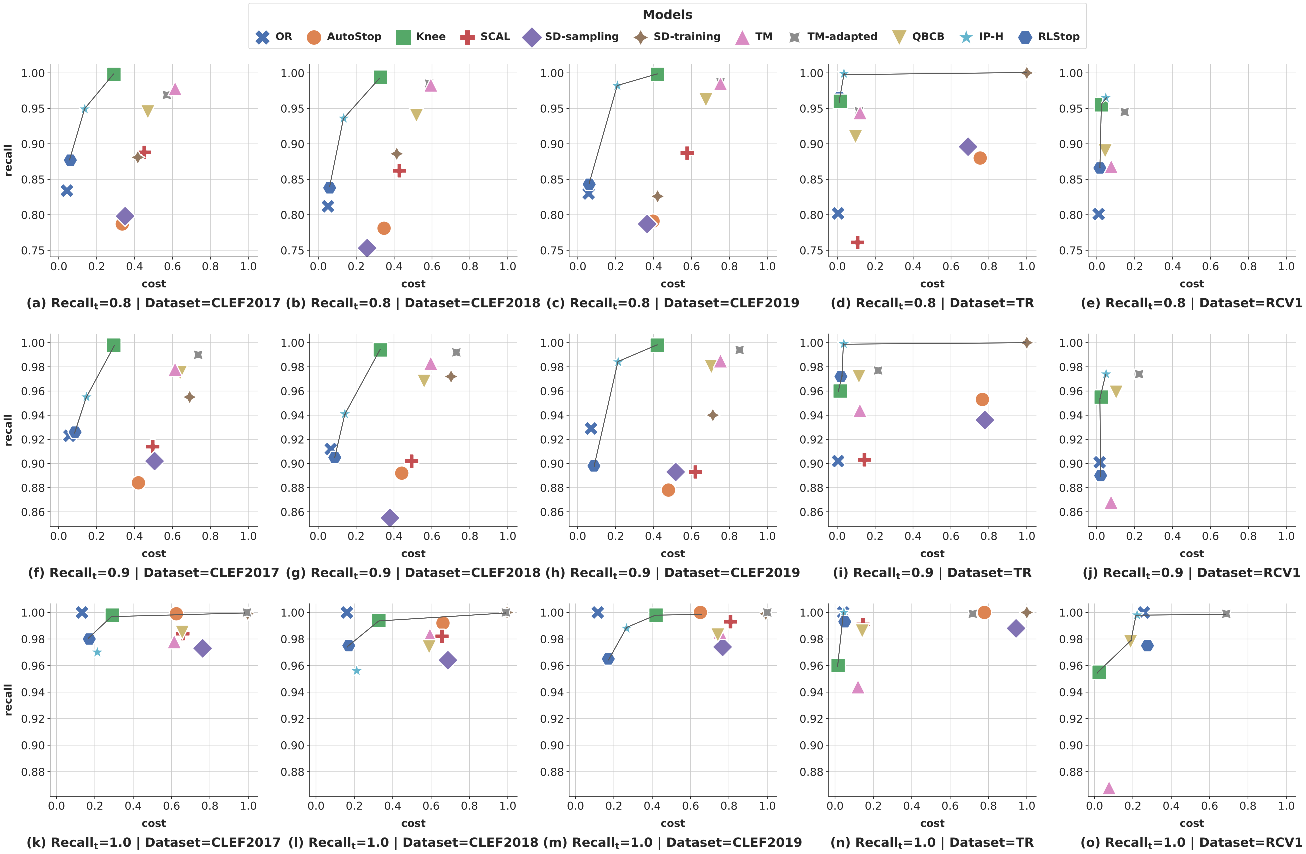

Figure 1: Performance of RLStop and baselines for Recall vs. Cost metrics. Grey lines indicates non-oracle Pareto optimal approaches. (Note differences in range of y-axis (recall) to avoid clustering of results.)

On the other hand, RLStop tends to undershoot for higher target recall levels, although it normally stops within a few percentage points of the target. This undershooting may be due to topics that contain a small number of relevant documents towards the end of their rankings, which RLStop does not reach.

Figure 2 shows how RLStop's excess varies across topics for all datasets at different target recall levels. For the majority of topics the excess costs is confined within a fairly narrow range, particularly for the TR collection. The variation is higher for all collections when target recall is 1.0 due to the additional challenge of identifying all relevant documents. This is more pronounced for RCV1, although the excess is within a relatively narrow range for the majority of topics.

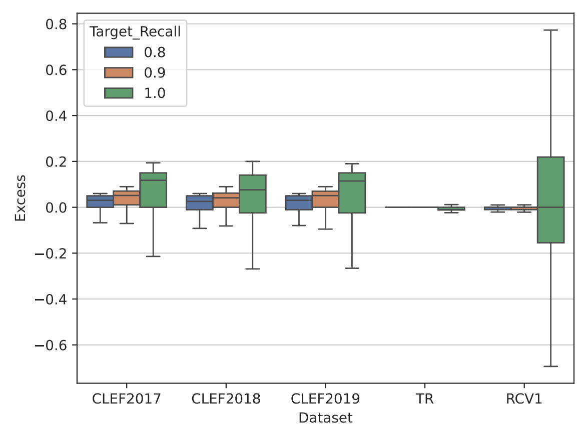

Figure 2: Distribution of RLStop excess across topics. (Outliers for target recall 1.0 removed for clarity.)

5 CONCLUSION

This paper proposes RLStop, a novel TAR stopping rule based on reinforcement learning. RLStop substantially reduces the workload required to screen collections for relevance. RLStop performs well in comparison with several baselines on multiple benchmark datasets at different target recall levels.

RLStop requires training data which may be available (e.g. from previous relevance screening carried out within a similar environment, such as within a systematic review team). It may not be suitable if these are not available or if confidence guarantees of reaching the target recall are required. RLStop also requires a new model to be trained for each target recall level. We plan to address these issues in future work.

REFERENCES

[1] Javed A Aslam, Virgiliu Pavlu, and Emine Yilmaz. 2005. Measure-based metasearch. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 571–572.

[2] Jason R Baron, Mahmoud F Sayed, and Douglas W Oard. 2020. Providing More Efficient Access To Government Records: A Use Case Involving Application of Machine Learning to Improve FOIA Review for the Deliberative Process Privilege. *arXiv preprint arXiv:2011.07203* (2020).

[3] Reem Bin-Hezam and Mark Stevenson. 2023. Combining Counting Processes and Classification Improves a Stopping Rule for Technology Assisted Review. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 2603–2609. https://doi.org/10.18653/v1/2023.findings-emnlp.171

[4] Max W Callaghan and Finn Müller-Hansen. 2020. Statistical stopping criteria for automated screening in systematic reviews. *Systematic Reviews* 9, 1 (2020), 1–14.

[5] Gordon Cormack and Maura Grossman. 2015. Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review. *arXiv preprint arXiv:1504.06868* (apr 2015). arXiv:1504.06868

[6] Gordon V Cormack and Maura R Grossman. 2014. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 153–162.

[7] Gordon V Cormack and Maura R Grossman. 2016. Engineering Quality and Reliability in Technology-Assisted Review. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 75–84.

[8] Gordon V. Cormack and Maura R. Grossman. 2016. Engineering quality and reliability in technology-assisted review. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Inc, 75–84. https://doi.org/10.1145/2911451.2911510

[9] Gordon V Cormack and Maura R Grossman. 2016. Scalability of continuous active learning for reliable high-recall text classification. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. 1039–1048.

[10] Giorgio Maria Di Nunzio. 2018. A study of an automatic stopping strategy for technologically assisted medical reviews. In *European Conference on Information Retrieval*. Springer, 672–677.

[11] Eugene Lodewick Grant and Richard S Leavenworth. 1980. *Statistical quality control*. Vol. 7. McGraw-Hill New York.

[12] Maura R. Grossman, Gordon V. Cormack, and Adam Roegiest. 2016. TREC 2016 Total Recall Track Overview. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016 (NIST Special Publication, Vol. 500-321)*. National Institute of Standards and Technology (NIST).

[13] Julian PT Higgins, James Thomas, Jacqueline Chandler, Miranda Cumpston, Tianjing Li, Matthew J Page, and Vivian A Welch. 2019. *Cochrane Handbook for Systematic Reviews of Interventions*. John Wiley & Sons.

[14] Noah Hollmann and Carsten Eickhoff. 2017. Ranking and Feedback-based Stopping for Recall-Centric Document Retrieval. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum*. 7–8.

[15] Brian E. Howard, Jason Phillips, Arpit Tandon, Adyasha Maharana, Rebecca Elmore, Deepak Mav, Alex Sedykh, Kristina Thayer, B. Alex Merrick, Vickie Walker, Andrew Rooney, and Ruchir R. Shah. 2020. SWIFT-Active Screener: Accelerated document screening through active learning and integrated recall estimation. *Environment International* 138 (2020), 105623. https://www.sciencedirect.com/science/article/pii/S0160412019314023

[16] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. 2017. CLEF 2017 Technologically Assisted Reviews in Empirical Medicine Overview. In *CEUR workshop proceedings*, Vol. 1866.

[17] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. 2018. CLEF 2018 Technologically Assisted Reviews in Empirical Medicine Overview. In *CEUR workshop proceedings*, Vol. 2125.

[18] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. 2019. CLEF 2019 Technology Assisted Reviews in Empirical Medicine Overview. In *CEUR workshop proceedings*, Vol. 2380.

[19] David Lewis, Eugene Yang, and Ophir Frieder. 2021. Certifying One-Phase Technology-Assisted Reviews. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

[20] David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research* 5 (2004), 361–397. https://doi.org/10.5555/1005332.1005345

[21] Dan Li and Evangelos Kanoulas. 2020. When to Stop Reviewing in Technology-Assisted Reviews: Sampling from an Adaptive Distribution to Estimate Residual Relevant Documents. *ACM Trans. on Information Systems* 38, 4 (2020), 1–36. https://doi.org/10.1145/3411755

[22] David E. Losada, Javier Parapar, and Alvaro Barreiro. 2019. When to stop making relevance judgments? A study of stopping methods for building information retrieval test collections. *Journal of the Association for Information Science and Technology* 70, 1 (2019), 49–60.

[23] Graham Mcdonald, Craig Macdonald, and Iadh Ounis. 2020. How the accuracy and confidence of sensitivity classification affects digital sensitivity review. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–34.

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602 [cs.LG]

[25] Alessio Molinari and Andrea Esuli. 2023. SALτ: efficiently stopping TAR by improving priors estimates. *Data Mining and Knowledge Discovery* (2023), 1–34.

[26] Ali Montazeralghaem, Hamed Zamani, and James Allan. 2020. A reinforcement learning framework for relevance feedback. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval*. 59–68.

[27] Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. *arXiv preprint arXiv:1704.04572* (2017).

[28] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research* 22, 268 (2021), 1–8. http://jmlr.org/papers/v22/20-1364.html

[29] Corby Rosset, Damien Jose, Gargi Ghosh, Bhaskar Mitra, and Saurabh Tiwary. 2018. Optimizing query evaluations using reinforcement learning for web search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1193–1196.

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[31] Ian Shemilt, Antonia Simon, Gareth J Hollands, Theresa M Marteau, David Ogilvie, Alison O'Mara-Eves, Michael P Kelly, and James Thomas. 2014. Pinpointing needles in giant haystacks: use of text mining to reduce impractical screening workload in extremely large scoping reviews. *Research Synthesis Methods* 5, 1 (2014), 31–49.

[32] Mark Stevenson and Reem Bin-Hezam. 2023. Stopping Methods for Technology-assisted Reviews Based on Point Processes. *ACM Transactions on Information Systems* 42, 3 (2023), 1–37.

[33] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning: An introduction*. MIT Press. http://incompleteideas.net/book/RLbook2020.pdf

[34] Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. 2023. Gymnasium. https://doi.org/10.5281/zenodo.8127026

[35] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8 (1992), 229–256.

[36] Eugene Yang, David Lewis, and Ophir Frieder. 2021. Heuristic Stopping Rules for Technology-Assisted Review. In *Proceedings of the 21st ACM Symposium on Document Engineering 2021 (DocEng '21)*. Article 31, 10 pages. https://doi.org/10.1145/3469096.3469873

[37] Eugene Yang, David D Lewis, and Ophir Frieder. 2021. On minimizing cost in legal document review workflows. In *Proceedings of the 21st ACM Symposium on Document Engineering 2021 (DocEng '21)*. Article 30, 10 pages. https://doi.org/10.1145/3469096.3469872

[38] Eugene Yang, David D Lewis, and Ophir Frieder. 2021. TAR on social media: A framework for online content moderation. In *2nd international conference on Design of Experimental Search & Information REtrieval Systems (DESIRES 2021)*. 147–155.

[39] Zhe Yu and Tim Menzies. 2019. FAST2: An Intelligent Assistant for Finding Relevant Papers. *Expert Systems with Applications* 120 (2019), 57–71.

[40] Wei Zeng, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2018. Multi page search with reinforcement learning to rank. In *Proceedings of the 2018 ACM SIGIR international conference on theory of information retrieval*. 175–178.

[41] Jianghong Zhou and Eugene Agichtein. 2020. Rlirank: Learning to rank with reinforcement learning for dynamic search. In *Proceedings of The Web Conference 2020*. 2842–2848.

[42] Justin Zobel. 1998. How Reliable Are the Results of Large-Scale Information Retrieval Experiments?. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 307–314.