Proceedings Paper:
Bin-Hezam, R. and Stevenson, R. orcid.org/0000-0002-9483-6006 (2025) A generalised and adaptable reinforcement learning stopping method. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval. The 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2025), 13-17 Jul 2025, Padua, Italy. ACM , pp. 761-770. ISBN 9798400715921

https://doi.org/10.1145/3726302.3729879

A Generalised and Adaptable Reinforcement Learning Stopping Method

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

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

ABSTRACT

This paper presents a Technology Assisted Review (TAR) stopping approach based on Reinforcement Learning (RL). Previous such approaches offered limited control over stopping behaviour, such as fixing the target recall and tradeoff between preferring to maximise recall or cost. These limitations are overcome by introducing a novel RL environment, GRLStop, that allows a single model to be applied to multiple target recalls, balances the recall/cost tradeoff and integrates a classifier. Experiments were carried out on six benchmark datasets (CLEF e-Health datasets 2017-9, TREC Total Recall, TREC Legal and Reuters RCV1) at multiple target recall levels. Results showed that the proposed approach to be effective compared to multiple baselines in addition to offering greater flexibility.

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 and Mark Stevenson. 2025. A Generalised and Adaptable Reinforcement Learning Stopping Method. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3726302.3729879

1 INTRODUCTION

Identifying all, or a significant proportion of, the relevant documents in a collection has applications in multiple areas including development of systematic reviews [16, 20–22], satisfying legal disclosure requirements [3, 15, 28], social media content moderation [45] and test collection development [27]. 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. 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.

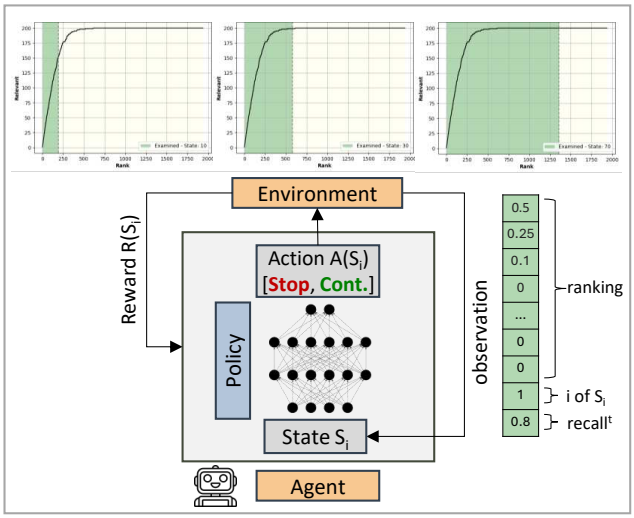

Figure 1: RL Environment for TAR Stopping

A wide range of approaches have been applied to the problem, including examination of the rate at which relevant documents are observed [10, 37], estimating the number of remaining relevant documents by sampling or classification [8, 11, 25, 36] and analysis of ranking scores [13, 17]. A recent approach is based on Reinforcement Learning (RL) (see Figure 1). *RLStop* uses deep RL to train a model to make stopping decisions which were found to perform well in comparison with a wide range of alternative stopping models [6]. However, RLStop suffers from a range of limitations, the most significant being that each model is trained for a specific target recall, limiting their generalisability. Also, in common with many other stopping methods, it does not provide a mechanism to adapt behaviour to balance the two stopping objectives: maximising the likelihood of reaching target recall and minimising the number of

SIGIR '25, July 13–18, 2025, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729879

documents examined. Finally, RLStop only uses information from documents that have already been examined, despite work demonstrating the value of incorporating predictions about likely relevance of unexamined documents [5, 43, 46].

Inspired by RLStop, this paper proposes an alternative RL-based stopping algorithm which overcomes these limitations and improves performance. In particular, it proposes a new reward function for use within the RL algorithm which allows the creation of a single stopping model that can be used with a range of target recall levels and also provides control over whether proposed stopping points prefer to maximise the number of relevant documents identified or minimise the number of documents examined. In addition, it demonstrates how information from unexamined documents can be incorporated within the RL framework. Experiments on six data sets commonly used in TAR evaluation demonstrate that these extensions improve performance. Further analysis explores the effect of ranking quality.

This work makes the following contributions: (1) introduces a RL-based stopping model that can be applied to multiple target recalls and adapted for different stopping preferences (e.g. maximising reaching target recall or minimising number of documents examined), (2) integrates a classifier within the RL environment to provide information about the unexamined documents, (3) demonstrates that this approach provides state-of-the-art performance through evaluation using multiple benchmark data sets, and (4) investigates the effect of a ranking quality on these approaches.[1]

2 BACKGROUND

A wide range of approaches have been proposed for TAR stopping methods. The most common is to estimate the total number of relevant documents in the collection and use this information to determine whether enough documents have been observed for the target recall to have been achieved. The total number of relevant documents has been estimated in a range of ways including sampling the unexamined documents [7, 8, 11, 19, 25, 30, 36], training a classifier using relevance judgments from the examined documents and applying it to those yet to be examined to estimate the number that are relevant [43, 44, 46], applying score distribution approaches that make use of the scores assigned by the ranking algorithm [17] and by applying counting processes [5, 37].

Other approaches identify a stopping point without explicitly estimating the total number of relevant documents. The *knee method* examines the gain curve produced as relevant documents are encountered to identify an inflection point where these become less frequent [10], while *target methods* apply sampling theory to randomly sample documents until a pre-specified number of relevant documents have been observed [10, 23, 44].

RLStop also identifies a stopping point without explicitly estimating the total number of relevant documents [6]. RLStop treats the stopping problem as a sequential decision-making problem and employs RL to make repeated decisions to either stop or continue examining documents, in contrast with the more common approach of treating stopping as an estimation problem. Although RL has been widely applied within Information Retrieval [1, 31, 33, 42], RLStop was the first application to stopping in TAR. RLStop was found to identify suitable stopping points at a range of target recall levels and outperformed other approaches on a range of TAR problems.

Despite its strong performance, RLStop suffers from several limitations. Each model is trained using a single target recall and is intended to make stopping decisions for that recall alone. This limits the generalisability of each model and multiple models would need to be trained in any cases where different target recalls are of interest. This limitation is unusual within stopping methods, for example it is straightforward for approaches based on estimating the total number of relevant documents to adjust the target recall and target methods account of the target recall in the formulae used to determine the number of relevant documents that need to be observed before stopping.

Applications of TAR stopping rules may have different requirements in terms of the balance between ensuring that the target recall is achieved and minimising the number of documents examined. For example, ensuring that a high proportion of relevant documents are identified is a priority for systematic reviews in the medical domain [16], while stopping as close to the target recall as possible is more likely to be important in the legal domain where minimising cost and unnecessary disclosure are important factors [14]. Some stopping methods, notably target methods [10, 23], do provide this functionality by incorporating the probability that the target recall is reached within the stopping decision. However, RLStop does not allow behaviours to be adapted in this way.

These limitations are addressed in this work by making use of an alternative reward function within the RL algorithm which allows a single model to be trained and then applied to multiple target recalls. This reward function also allows the balance between ensuring that target recall is achieved and number of documents examined to be considered during training, allowing different models to be created.

In addition, RLStop examines the documents in the order in which they are ranked and bases its stopping decisions only on information from the documents that have been examined so far. Although this approach has the advantage of prioritising the documents that are more likely to be relevant, information about the unexamined documents has been demonstrated to be useful for this problem [5, 44, 46]. This paper also demonstrates how information about unexamined documents can be integrated into the RL approach by training a classifier on the documents for which relevance judgments are available and applying it to the unexamined documents.

3 REINFORCEMENT LEARNING APPROACH

In RL an agent interacts with an environment and receives rewards depending upon its actions. The environment consists of a *state space*, S, which defines the set of possible states that the agent can occupy and an *action space*, A, that lists the set of possible actions that the agent can take at each state. The goal of RL is to learn a *policy*, $\pi(s, a)$ where $a \in A$ and $s \in S$, which defines the value of choosing action a given states s and thereby guides the agent's behaviour. A *reward function*, $R(s)$, assigns a score to each state and is used to train the policy. An RL algorithm explores the environment by making sequential choices of action through trial-and-error with the goal of maximising the reward obtained. Each choice of action may affect not only the immediate reward obtained but also the rewards obtained following subsequent actions.

[1] Code available from https://github.com/ReemBinHezam/GRLStop

RL algorithms must balance exploitation and exploration, that is maximising the reward obtained at each step by exploiting what it has already experienced versus maximising cumulative future rewards by exploring the environment by taking new actions [38].

3.1 Stopping RL environment

This section describes how a stopping algorithm is implemented within an RL environment. The agent examines a ranked list of documents in batches, starting with the highest ranked documents. After each batch the agent decides to either continue examining documents or stop.

State Space: The state space represents the document ranking and target recall. A ranking is split into B fixed size batches, each containing $\frac{N}{B}$ documents for a collection of N documents. The agent examines batches sequentially and obtains relevance judgments 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 examined by the agent. 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.

States are represented by a fixed size vector of length B+2 where each of the first B elements represent a single batch and the final two represent the number of batches that have been examined so far (E) and the target recall. The way in which the values representing each batch are computed depends on whether or not the batch has been examined by the agent. For examined batches, i.e. batches $1 \ldots E$, the value shows the proportion of relevant documents within the batch. However, since the remaining batches, i.e. $E+1 \ldots N$, have not yet been examined, the number of relevant documents within each is not known. For these batches a classifier is trained using the relevance judgments from the examined batches and applying it to each of the unexamined batches to provide an estimate of the number of relevant documents it contains.

Note that the state representation used by RLStop only included information about relevant documents within examined batches. This work integrates estimates of the number of relevant documents in unexamined batches produced by a classifier, information that has previously been demonstrated to be useful [43, 46], into the RL approach.

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}. The last possible agent step is to state S_B since this represents the end of the ranking and all documents have been examined.

Policy: A neural network is an appropriate choice of policy when the number of potential states is large, such as the state space encoding used here [2]. The policy used is a feed-forward network consisting of an input layer of length $B+2$, representing the current state, two hidden layers and a binary output layer indicating the chosen action, which is converted to a probability distribution over the two possible actions by a softmax activation function.

Reward function: The policy is trained using a reward function, $R(S_i)$, which assigns a score to each state indicating its desirability to the agent. A suitable reward function should have the following properties 1) encourages the agent to continue examining documents until the target recall has been reached, 2) penalises further examination after it has been reached, 3) have the same range for different document rankings and target positions, and 4) can be adapted to offer control over the balance between undershooting and overshooting of target recall.

Note that the third property is necessary for to develop a single model that can be applied to multiple target recalls since it is difficult for RL algorithms to learn good policies when reward function values across the state space cannot be compared directly.

A function that meets these properties is:

$$R(S_i) = \begin{cases} \frac{i^m - (i-1)^m}{T^m} & \text{if } i \leq T \\ \\ \frac{(B-i)^n - (B-i+1)^n}{(B-T)^n} & \text{if } i > T \end{cases} \quad (1)$$

where i is the current state (i.e. ith batch), B is the number of batches into which the ranking is split and T is the batch at which the target recall is reached. (Note that while the value of T is known while the algorithm is being trained, it is not known when it is applied.) In addition, m and n are parameters allowing the function to be adapted between preferring to maximise the likelihood of reaching the target recall and minimising the number of documents examined.

This function assigns a positive reward for states at, or below, the target recall and a negative reward for states after it has been exceeded, thereby meeting the first two properties for a suitable reward function.

Other properties of Equation 1 are best understood by considering the *cumulative reward* produced for an RL episode, i.e. the sum of rewards for all states visited by the agent during that episode. In this application an episode consists of examining a ranking from the first batch until a stopping decision is made. The cumulative reward for an episode that ends at S_i, $CR(S_i)$, is given by:

$$CR(S_i) = \begin{cases} \left(\frac{i}{T}\right)^m & \text{if } i \leq T \\ \\ \left(\frac{B-i}{B-T}\right)^n & \text{if } i > T \end{cases} \quad (2)$$

The maximum value for this function, which occurs when the target recall has been reached (i.e. $i = T$) is always 1. The function's value decreases for values below or above T with a minimum possible value of 0 reached when $i = B$ (provided $T \neq B$). The range of the cumulative reward is therefore invariant to the point in the ranking at which the target recall is reached, so the third property is satisfied.

The properties of the reward function, and therefore the cumulative reward, can be controlled by varying the values of the parameters m and n. The value of m determines the reward for states before the target recall has been reached. Setting it < 1 increases this reward while it is reduced for values > 1. Similarly, varying n changes the reward after the target recall has been exceeded. Choosing appropriate values for m and n provides a mechanism to control the balance between ensuring the target recall has been reached and minimising

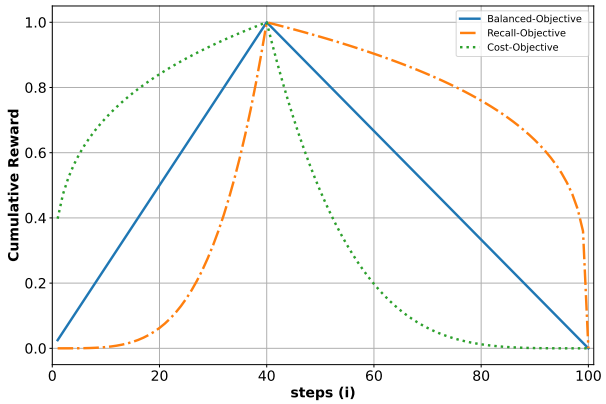

Figure 2: Example cumulative reward functions produced by varying parameters m and n. In this example, the target recall is achieved at batch 40 of 100 in this example (i.e. $T = 40$ and $B = 100$). The solid blue line shows the function produced when $m = n = 1$. The dashed orange line, produced when $m = 4$ and $n = 0.25$, shows a function that assigns less reward before the target recall is achieved and more after it. Similarly, the dotted green line, $m = 0.25$ and $n = 4$, assigns more reward before and less after.

the number of documents examined. For example, setting $m = 4$ and $n = 0.25$ alters the function to increase the reward when the target recall is exceeded and reduce the reward when it is not met, thereby encouraging policies that ensure enough documents have been identified to achieve the target recall even at the expense of examining more documents than necessary. Similarly, setting $m < 1$ and $n > 1$ reverses this to increase reward for minimising the number of documents examined, even when the target recall has not been achieved (see Figure 2).

Note that a reward function could be created using a single parameter to control the balance between maximising recall or minimising workload, and such a function would be adequate for the majority of applications. However, we chose to include two parameters to increase the possible reward functions available.

The reward function presented here can be compared against the one used by RLStop. That function met the first two properties by assigning positive reward until the target recall was reached and then negative reward thereafter. However, the maximum and minimum cumulative reward varied depending on the batch at which the target recall was reached (i.e. value of T) with higher maximum values when this batch occurred later in the rankings. (The maximum cumulative reward for RLStop is $\frac{T-1}{2}$ while the minimum, $\frac{2T-B-2}{2}$, is negative in some cases.) In addition, RLStop's reward function did not offer any mechanism to adapt the policy to prefer to overshoot or undershoot the target recall.

4 EXPERIMENTS

4.1 Datasets

Performance was evaluated on six datasets from multiple domains that are widely used in high recall retrieval studies. The datasets are highly imbalanced, with a very low percentage of relevant documents for each topic.

CLEF Technology-Assisted Review in Empirical Medicine (CLEF 2017/2018/2019) [20–22]: A collection of systematic reviews from 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. The training dataset consists of 12 reviews from CLEF2017.

TREC Total Recall (TR) [15]: A collection of 290,099 emails related to Jeb Bush's eight-year tenure as Governor of Florida (athome4). The collection contains 34 topics. Each topic is labelled with a short title and based on an issue associated with Jeb Bush's governorship. The training dataset is (athome1) which consists of 10 topics from the same collection.

TREC Legal (Legal) [12]: A collection of 685,592 Enron emails made available by the Federal Energy Review Commission during their investigation into the company's collapse. Two topics were used for testing and two for training.

RCV1: [24] A collection of Reuters news articles labelled with categories. Following [43, 44], 45 categories were used to represent a range of topics at different prevalence and difficulty levels, and the collections downsampled to 20% for efficiency. The remaining unused 94 topics in the collection were used for training.

To ensure fair comparison, all stopping methods are applied to the same rankings. AutoTAR [9] is a greedy Active Learning approach that represents state-of-the-art performance on total recall tasks and is commonly used within work on stopping methods. It allows comparison with a range of alternative approaches [25]. AutoTAR rankings for each dataset were created using a reference implementation and default parameters.[2] These rankings were used for all experiments, except those reported in Section 5.4 which explore the effect of varying ranking quality.

4.2 Baselines

Multiple stopping methods representing a range of approaches were used as baselines, including those widely used in previous work.

RLStop [6] is the existing RL-based approach described previously. Separate models are trained for each dataset and target recall.

SCAL [11] estimates the number of relevant documents in the collection by sampling across the entire ranking.

AutoStop [25] employs a similar approach to SCAL. Sampling is carried out using unbiased estimators [18, 39] to account for the decreasing prevalence of relevant documents.

SD-training/SD-sampling [17] make use of the scores assigned by the ranking algorithm to estimate the total number of relevant documents. They differ in how they identify relevant documents needed to model ranking scores. SD-training from the training data and SD-sampling by sampling documents to obtain relevance judgments from a simulated user.

IP-H [37] examines the rate at which relevant documents are observed and uses this information within a statistical model (counting process) to estimate the total number of relevant documents.

Knee [10] examines the "gain curve" produced by plotting the cumulative total of relevant documents identified against rank. A "knee

[2] https://github.com/dli1/auto-stop-tar

detection" algorithm [34] is used to examine this curve's gradient to determine when the frequency of relevant documents decreases.

TM-adapted [10] randomly samples from the collection until a pre-specified number of relevant documents have been found. An extension of this approach that allows this figure to be adapted for different target recalls is used [37].

QBCB [23] similar to the previous approach but relies on a set of known relevant documents. Sampling continues until all documents in the set have been found.

Baselines are computed using reference implementations from previous work [25, 37] 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 the RCV1 dataset 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. Note that in some cases the oracle can only achieve a recall higher than the target when it is not possible to stop exactly at target, e.g. given a target recall of 0.8 and collection containing 7 relevant documents, the oracle will stop after 6 relevant documents have been found, i.e. recall 0.86.

4.3 Evaluation Metrics

Approaches were evaluated using metrics commonly used in previous work on TAR stopping criteria, e.g. [10, 25], calculated using the `tar_eval` open-source evaluation script.[3]

Recall: Proportion of relevant documents within the collection identified before the method stops examining documents.

Reliability (Rel.): Percentage of topics where the desired target recall was reached (or exceeded). For each topic, the reliability is 1 if the target recall is reached before the stopping examining documents, and 0 otherwise.

Cost: Percentage of documents examined.

Cost Difference (CostDiff): In addition to the above metrics, the cost difference score is also introduced. This is the difference between the proportion of documents that were examined and would have been examined by the oracle method (i.e. stopping immediately upon reaching the target recall). It is computed as $cost(method) - cost(oracle)$. Positive scores indicate that the target recall has been reached and negative that it was not while the absolute value indicates the proportion of the collection by which the ideal stopping point was missed. Cost difference combines information about whether the target recall has been reached and number of documents examined within a single metric.

4.4 Implementation

Proximal Policy Optimization (PPO) [35] is used as the RL algorithm. PPO is a policy gradient approach that directly learn the policy that maps states to actions rather than indirectly extracting it from state-action pairs. It is an actor-critic RL algorithm that combines policy-based (actor) and value-based (critic) RL, where the actor network decides the actions, and the critic network evaluates them to optimise the value function (i.e. reward value). PPO is based on the REINFORCE [41] algorithm but with several enhancements, and most importantly in our case, collects trajectories from different environments simultaneously from independent parallel actors, which allows a policy to be trained using rankings from multiple topics. PPO employs a clipping mechanism limiting policy updates within a specific range and leading to more stable training which is important when training the agent with multiple environments.

PPO is more suitable than alternative approaches like DQN [29] which relies on Q-values to map actions to specific states and does not easily generalise across different environments such as multiple rankings. PPO is also more sample-efficient than some alternative methods, thereby reducing the amount of data required to learn effective policies.

Classifier: The classifier applied to the unexamined portion of the ranking was implemented using logistic regression with a TF-IDF document representation. Despite its simplicity, this approach has proved successful for TAR problems and has commonly been used by previous approaches [5, 25, 43, 44, 46]. The classifier was implemented using the `scikit-learn` library with `LogisticRegression` and TF-IDF scores generation using `TfidfVectorizer` and default configurations, which proved to work well in previous work [5, 25]. Documents features used were title and abstract for the CLEF 2017-19 datasets, title and content of emails for TREC TR and Legal and the entire news article for RCV1. Cost sensitive learning was used to mitigate class imbalance during classifier training by using a weight of 1 for the minority class (i.e. relevant) and weighing the minority class (i.e. not relevant) as the ratio of the number of documents in the minority and majority classes [26].

Implementation and Hyperparameters: The RL environment was created using the `Gymnasium` library [40] with vector environments, which allows multiple independent environments to be stacked together, thereby allowing simultaneous training on multiple topics. The `Stable-Baseline3` [32] implementation of PPO was used.

Following previous work, the number of batches, B, was set to 100 to ensure a reasonable number of possible stopping positions without the environment becoming too sparse [6].

Experiments were carried out exploring multiple configurations which required different hyperparameters due to the changes to the environment and reward behaviour. The best values for each setting were identified using a grid search. For all configurations, the number of epochs, discount factor, learning rate, and neural network hidden layers nodes were set to their default values of 10, 0.99, 0.0003 and 64, which also proved to provide the best results. The number of steps per environment at each rollout was set to 10, the entropy coefficient to 0.1 and the clipping range to 0.1. Early stopping callback [32] was applied when there was no improvement in the policy for ten consecutive rollouts to help avoid overfitting, save training time and ensure fair comparison.

Section 5.2 describes a configuration in which the classifier was not included. For this configuration, the number of steps per environment at each rollout was increased to 100 since the environment is less informative and more steps are required for to explore it fully. The entropy coefficient was also set to 0.001 to encourage a balance between the number of steps and exploration, thereby introducing more stable policy updates.

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

The reward function parameters m and n were set to 1 for all experiments except those in Section 5.3 which explore the effect of altering the objectives.

5 RESULTS

The first experiment compares the proposed approach, referred to as **GRLStop**, with the alternative methods (baselines and oracle) described in Section 4.2. Results are reported using recall and cost, averaged across all topics in each collection. (Note that the reliability and CostDiff scores for GRLStop are also available in Table 1.) Recall and cost metrics were chosen because they provide information about how well the stopping algorithm has achieved its two key objectives: identifying relevant documents and minimising the total number of documents examined. Suitable stopping methods need to take account of both objectives so Pareto efficient approaches (i.e. those that reach the highest recall for a particular cost) are identified for each setting.

Results are shown in Figure 3 for all datasets with target recalls set to 0.8, 0.9 and 1.0. (Target recall 0.7 is also included for subsequent experiments but not for this one since it was not possible to obtain scores for several of the baseline approaches.) Each dataset is represented in a single column and each target recall in a single row. Grey lines in each sub-figure indicate the Pareto front (i.e. set of Pareto efficient approaches).

The variation in relative performance across the range of configurations included in the experiment demonstrates the difficulty of selecting a single approach that is optimal in all circumstances. However, GRLStop is Pareto optimal in almost every case. The single exception is for the TR dataset for target recall 1.0 (subfigure (p)) where GRLStop is very close to being Pareto optimal. The approach was consistently able to reach the target recall level with lower cost compared to the baselines in almost all non-total recall scenarios when the target recall is 0.8 or 0.9. It is also consistently closer to the optimal Oracle results than other approaches.

GRLStop often fails to meet the target recall when it is set to 1.0 (bottom row of Figure 3), most noticeably for the CLEF 2018 and 2019 datasets. However, the cost is substantially lower than for other approaches (several of which examine the majority of the collection) and achieves a good balance between cost and recall.

The knee and IP-H approaches are also Pareto optimal in several cases. However, their cost is generally higher then GRLStop (indicating that they require more documents to be examined) and they tend to be further from the oracle. The remaining approaches are not Pareto optimal under any scenario, or only occasionally.

These results demonstrate that GRLStop is comparable with state-of-the-art approaches for TAR stopping and is often able to achieve performance closer to the optimal oracle than alternative methods.

5.1 Fixed vs Varying Target Recall

A key feature of GRLStop is its ability to train a single model that can be used to identify stopping points for different target recalls. This is in contrast with RLStop where a different model has to be trained for each target recall, limiting their flexibility. This also provides RLStop with an advantage in the results shown in Figure 3 since all GRLStop results for a dataset are produced by a single model, while the RLStop ones are produced by three separate models.

To explore this further, an experiment was carried out in which GRLStop was compared against versions of RLStop trained for two target recalls: RLStop7 for target recall 0.7 and RLStop9 for target recall 0.9. Each model was then applied in scenarios with for different target recalls (0.7, 0.8, 0.9 and 1.0). The difference between this experiment and the one reported in Figure 3 is that the RLStop7 and RLStop9 models are applied to all target recalls, rather than just the one that they have been trained for.

Results are shown in Table 1 where recall, reliability, cost and CostDiff metrics are reported for each approach. Note that the recall and cost metrics for the RLStop7 and RLStop9 models are the same for all target recalls, since these models only aim to achieve the target recall they were trained for. The reliability and CostDiff scores for these models do change since these metrics consider the target recall. However, for GRLStop, results for all metrics vary depending on the target recall, demonstrating the models generalisability across target recalls.

Unsurprisingly, the RLStop7 and RLStop9 models perform best when applied to the target recall they had been trained for. But the performance of these models tends to degrade when they are applied to different target recalls. For example, performance of RLStop7 reduces as it is applied to increasingly higher target recalls. In these cases the models tends to undershoot the target, as indicated by the drop in reliability and CostDiff scores. Similarly, RLStop9 overshoots when applied to lower target recalls, as demonstrated by the high cost and cost difference figures in comparison with the other two approaches.

The costDiff scores for GRLStop are consistently lower than those for RLStop7 and RLStop9, even in the cases where it was more costly (such as target recall 1.0), demonstrating its ability to adapt the stopping decision to the target recall being sought.

5.2 Classifier Effect

A further analysis was carried out to determine the effect of including the classifier to predict relevance of unobserved documents on overall performance. A version of GRLStop that did not employ the classifier was created by adapting the RL environment described by Section 3.1 so that the classifier-predicted values for each unobserved document (i.e. all batches from $E+1 \ \ldots \ N$) are replaced by a dummy value -1, an approach similar to the one used by RLStop.

The box plot in Figure 4 compares the results obtained with and without the classifier over all topics. CostDiff scores are reported since to allows per-topic differences to be represented. It can be seen that including classifier prediction labels moves the CostDiff score closer to the optimal score of 0 in the majority of scenarios. In the majority of circumstances, particularly for target recalls 0.7 and 0.8 and the CLEF data sets, removing the classifier leads to more documents being examined than necessary, indicated by the overall increase in CostDiff. However, in other cases such as target recall 1.0, removing the classifier results in an increased failure to meet the target recall (as indicated by increased negative CostDiff).

The differences between the results obtained with and without the classifier for each metric were compared across all target recalls and found to be statistically significant for all target recalls except 0.9 (paired t-test with Bonferroni correction, $p < 0.05$).

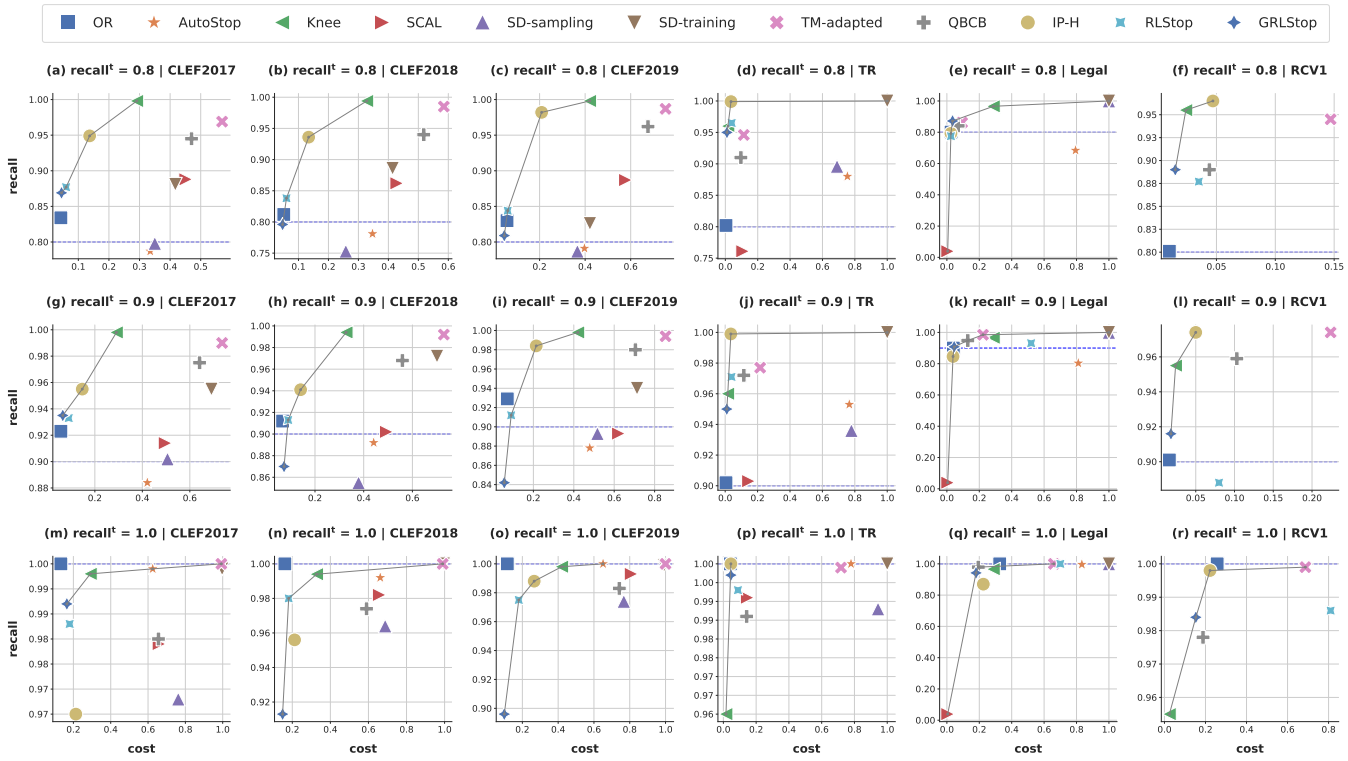

Figure 3: Recall vs cost. Target recall indicated by horizontal blue dashed line and Pareto front as grey line. Note that scale of y-axis varies across sub-figures.

Table 1: Comparison of GRLStop and RLStop trained on target recalls 0.7 and 0.9

		Target Recall = 1.0				Target Recall = 0.9				Target Recall = 0.8				Target Recall = 0.7			
Dataset	**Model**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**	**Recall**	**Rel.**	**Cost**	**CostDiff**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**
CLEF2017	**RLStop7**	0.851	0.500	0.050	-0.084	0.851	0.667	0.050	-0.009	0.851	0.733	0.050	0.004	0.851	0.767	0.050	0.014
	RLStop9	0.933	0.600	0.090	-0.044	0.933	0.767	0.090	0.031	0.933	0.800	0.090	0.044	0.933	0.933	0.090	0.054
	GRLStop	0.992	0.767	0.165	0.032	0.935	0.767	0.065	0.006	0.869	0.700	0.045	0.000	0.791	0.733	0.031	-0.006
CLEF2018	**RLStop7**	0.803	0.333	0.050	-0.110	0.803	0.500	0.050	-0.019	0.803	0.667	0.050	-0.004	0.803	0.700	0.050	0.006
	RLStop9	0.913	0.500	0.090	-0.070	0.913	0.767	0.090	0.021	0.913	0.867	0.090	0.036	0.913	0.867	0.090	0.046
	GRLStop	0.913	0.700	0.149	-0.012	0.870	0.700	0.073	0.003	0.796	0.633	0.047	-0.007	0.732	0.667	0.034	-0.010
CLEF2019	**RLStop7**	0.811	0.500	0.050	-0.062	0.811	0.600	0.050	-0.018	0.811	0.633	0.050	-0.005	0.811	0.667	0.050	0.004
	RLStop9	0.912	0.633	0.090	-0.022	0.912	0.767	0.090	0.022	0.912	0.800	0.090	0.035	0.912	0.900	0.090	0.044
	GRLStop	0.896	0.633	0.099	-0.013	0.842	0.700	0.056	-0.012	0.809	0.667	0.045	-0.010	0.743	0.633	0.034	-0.011
Legal	**RLStop7**	0.504	0.000	0.010	-0.320	0.504	0.000	0.010	-0.035	0.504	0.000	0.010	-0.025	0.504	0.000	0.010	-0.015
	RLStop9	0.931	0.500	0.520	0.190	0.931	0.500	0.520	0.475	0.931	1.000	0.520	0.485	0.931	1.000	0.520	0.495
	GRLStop	0.942	0.500	0.180	-0.150	0.908	0.500	0.045	0.000	0.872	1.000	0.035	0.000	0.836	1.000	0.030	0.005
TR	**RLStop7**	0.965	0.324	0.039	-0.010	0.965	0.971	0.039	0.027	0.965	0.971	0.039	0.028	0.965	0.971	0.039	0.028
	RLStop9	0.971	0.324	0.039	-0.010	0.971	0.971	0.039	0.028	0.971	0.971	0.039	0.028	0.971	0.971	0.039	0.028
	GRLStop	0.997	0.618	0.046	-0.004	0.950	0.941	0.010	-0.002	0.950	0.941	0.010	-0.001	0.950	0.941	0.010	-0.001
RCV1	**RLStop7**	0.850	0.156	0.034	-0.229	0.850	0.533	0.034	0.012	0.850	0.689	0.034	0.017	0.850	0.800	0.034	0.020
	RLStop9	0.888	0.200	0.080	-0.182	0.888	0.578	0.080	0.058	0.888	0.778	0.080	0.064	0.888	0.911	0.080	0.067
	GRLStop	0.984	0.356	0.153	-0.109	0.916	0.644	0.018	-0.004	0.890	0.778	0.015	-0.002	0.877	0.889	0.014	0.000

These results demonstrate that integrating the classifier's predictions into the RL environment improves stopping decision performance.

5.3 Adapting Objectives

GRLStop allows the reward function to be varied to achieve different objectives such as encouraging the policy to meet the target recall or to minimise the number of documents examined. The effect of doing so was explored in an experiment comparing versions

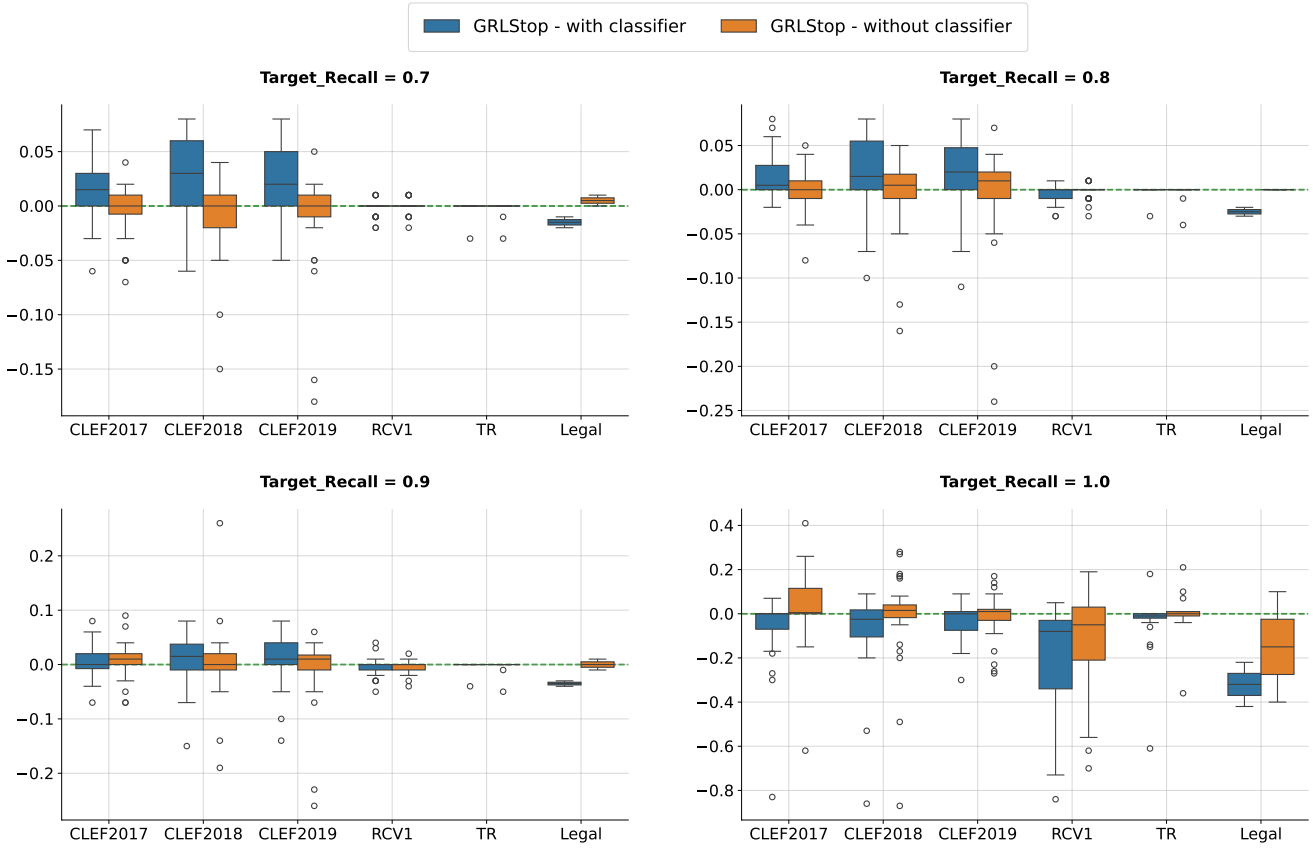

Figure 4: Effect of including/excluding classifier on CostDiff. Green dashed horizontal line indicates optimal value (i.e. 0).

of GRLStop developed using different reward functions. The first, *GRLStop-recall-obj*, is designed to encourage a policy that ensures the target recall is achieved, even if this requires more documents to be examined. Its reward function was created by setting $m = 4$ and $n = 0.25$, values chosen to encourage the intended behaviour without being too extreme. The second, *GRLStop-cost-obj*, aims to minimise the number of documents examined, even if raises the risk that the target recall is not met. It was created by setting $m = 0.25$ and $n = 4$. These approaches are compared against the standard GRLStop ($m = n = 1$) which balances these objectives (*GRLStop-cost-balanced*). The classifier predictions were not included in this experiment since it was found that the impact of varying the reward function was more pronounced when it was not included.

Table 2 shows the performance of the three approaches for a range of target recalls over each data set. These results demonstrate that GRLStop-recall-obj consistently reached higher recall levels than GRLStop and GRLStop-cost-obj, normally at the expense of an increase in cost although this is sometimes quite limited (e.g. for the TR dataset and RCV1 with lower target recalls). The improvement in recall and reliability is particularly noticeable for the Legal dataset, although it only consists of two topics which limits the possible reliability scores. In this case, GRLStop-recall-obj is always able to achieve the target recall for both topics with a cost that does not exceed 20%. Examination of the learning process revealed that the RL algorithm needed more training steps to converge on a policy for the GRLStop-recall-obj objective, although these policies had higher cumulative rewards than the ones learned using the other two objectives.

On the other hand, the cost for GRLStop-cost-obj is consistently lower than the other two models, often considerably so. This reduction in the number of documents examined comes at the expense of lower recall, which often fails to reach the target. However, the reward function used to train GRLStop-cost-obj is designed to prefer minimising effort over ensuring the target recall has been reached so, in that sense, it has met its objective.

These results demonstrate that adapting the reward function used by GRLStop provides a mechanism to control the balance between preferring to ensure that target recall is achieved and minimising the number of documents examined.

5.4 Ranking Quality

Results for the majority of stopping methods have only been reported for a single ranking, despite previous work demonstrating the ranking quality can affect stopping method performance [37]. GRLStop's performance across ranking qualities was explored by generating three sets of rankings with different levels of effectiveness. These were produced using AutoTAR with low, mid-range and good rankings being created by stopping the active learning process

Table 2: Effect of Varying Reward Function

		Target Recall = 1.0				Target Recall = 0.9				Target Recall = 0.8				Target Recall = 0.7			
Dataset	**Model**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**
CLEF2017	**GRLStop-recall-obj**	**0.967**	0.700	0.129	-0.005	**0.952**	0.867	0.126	0.067	**0.944**	0.900	0.124	0.078	**0.937**	0.967	0.116	0.080
	GRLStop-balanced	0.942	0.533	0.067	-0.066	0.925	0.733	0.064	0.005	0.914	0.867	0.062	0.016	0.883	0.900	0.051	0.015
	GRLStop-cost-obj	0.685	0.267	**0.026**	-0.108	0.616	0.267	**0.019**	-0.040	0.514	0.267	**0.010**	-0.036	0.514	0.367	**0.010**	-0.026
CLEF2018	**GRLStop-recall-obj**	**0.969**	0.667	0.164	0.003	**0.960**	0.833	0.156	0.086	**0.942**	0.900	0.140	0.086	**0.941**	0.967	0.136	0.092
	GRLStop-balanced	0.928	0.367	0.084	-0.076	0.924	0.700	0.081	0.012	0.898	0.900	0.073	0.019	0.887	0.900	0.070	0.025
	GRLStop-cost-obj	0.686	0.167	**0.033**	-0.128	0.605	0.200	**0.026**	-0.043	0.436	0.267	**0.010**	-0.044	0.436	0.300	**0.010**	-0.034
CLEF2019	**GRLStop-recall-obj**	**0.971**	0.767	0.158	0.046	**0.966**	0.900	0.147	0.079	**0.957**	0.933	0.139	0.084	**0.952**	0.967	0.136	0.090
	GRLStop-balanced	0.933	0.533	0.081	-0.031	0.926	0.833	0.077	0.009	0.915	0.867	0.074	0.019	0.899	0.833	0.068	0.022
	GRLStop-cost-obj	0.661	0.367	**0.029**	-0.083	0.620	0.367	**0.025**	-0.042	0.459	0.267	**0.010**	-0.045	0.459	0.267	**0.010**	-0.036
Legal	**GRLStop-recall-obj**	**0.998**	0.000	0.200	-0.130	**0.998**	1.000	0.200	0.155	**0.998**	1.000	0.200	0.165	**0.998**	1.000	0.200	0.175
	GRLStop-balanced	0.504	0.000	**0.010**	-0.320	0.504	0.000	**0.010**	-0.035	0.504	0.000	**0.010**	-0.025	0.504	0.000	**0.010**	-0.015
	GRLStop-cost-obj	0.504	0.000	**0.010**	-0.320	0.504	0.000	**0.010**	-0.035	0.504	0.000	**0.010**	-0.025	0.504	0.000	**0.010**	-0.015
TR	**GRLStop-recall-obj**	**0.971**	0.324	0.011	-0.038	**0.970**	0.971	0.011	-0.001	**0.970**	0.971	0.011	-0.001	**0.964**	0.971	**0.010**	-0.001
	GRLStop-balanced	**0.971**	0.324	0.016	-0.033	**0.970**	0.971	0.011	-0.001	0.970	0.971	0.011	-0.001	0.964	0.971	**0.010**	-0.001
	GRLStop-cost-obj	0.950	0.294	**0.010**	-0.039	0.950	0.941	**0.010**	-0.002	0.950	0.941	**0.010**	-0.001	0.950	0.941	**0.010**	-0.001
RCV1	**GRLStop-recall-obj**	**0.992**	0.578	0.189	-0.074	**0.956**	0.822	0.060	0.038	**0.922**	0.844	0.023	0.007	**0.888**	0.911	0.015	0.002
	GRLStop-balanced	0.976	0.222	0.066	-0.197	0.898	0.578	0.017	-0.005	0.872	0.733	0.013	-0.003	0.850	0.800	0.012	-0.002
	GRLStop-cost-obj	0.870	0.156	**0.016**	-0.247	0.842	0.511	**0.013**	-0.010	0.802	0.556	**0.010**	-0.006	0.802	0.733	**0.010**	-0.004

Table 3: Performance on Range of Ranking Qualities

	Ranking	**Target Recall = 1.0**				**Target Recall = 0.9**				**Target Recall = 0.8**				**Target Recall = 0.7**			
Dataset	**Quality**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**	**Recall ↑**	**Rel. ↑**	**Cost↓**	**CostDiff**	**Recall ↑**	**Rel. ↑**	**Cost ↓**	**CostDiff**
CLEF2017	**Low**	0.941	0.567	0.423	0.005	0.898	0.667	0.235	0.079	0.856	0.733	0.133	0.051	0.767	0.767	0.061	0.005
	Mid	0.961	0.700	0.228	0.002	0.935	**0.833**	0.071	0.015	0.885	**0.800**	**0.048**	0.005	**0.837**	**0.833**	0.037	0.002
	Good	**0.989**	**0.767**	**0.226**	0.066	**0.957**	**0.833**	**0.067**	0.008	**0.890**	0.767	**0.048**	0.002	0.809	0.767	**0.035**	-0.002
CLEF2018	**Low**	**0.988**	0.633	0.498	0.078	**0.942**	**0.833**	0.259	0.110	**0.897**	**0.900**	0.141	0.057	0.805	0.800	0.091	0.029
	Mid	0.973	0.700	0.235	-0.011	0.892	0.700	0.073	-0.007	0.842	0.800	0.054	0.002	**0.822**	**0.833**	0.046	0.002
	Good	0.954	**0.733**	**0.191**	-0.018	0.876	0.667	**0.066**	-0.002	0.832	0.733	**0.050**	-0.005	0.734	0.600	**0.034**	-0.011
CLEF2019	**Low**	**0.911**	**0.600**	0.493	0.081	0.848	0.633	0.321	0.083	0.798	0.633	0.175	0.017	0.732	0.633	0.120	0.002
	Mid	0.866	0.567	0.192	-0.080	0.842	0.700	0.077	-0.024	**0.802**	**0.667**	0.058	-0.015	**0.811**	**0.800**	0.054	-0.012
	Good	0.882	**0.600**	**0.168**	-0.074	**0.854**	**0.733**	**0.067**	-0.022	0.789	**0.667**	**0.048**	-0.028	0.750	0.667	**0.037**	-0.031

after assessing 25%, 50% and 75% of the collection respectively. Quality of the rankings produced was measured by computing the area under recall curve (AURC) and found to average 0.87, 0.92 and 0.96 for the low, mid and good rankings. For comparison, the equivalent score for the rankings used in the previous experiments was 0.97.

Table 3 shows the results of GRLStop on the CLEF datasets. Results for other data sets display similar trends but are not included for brevity. These results indicate that, as expected, GRLStop's performance is affected by the ranking quality. The most noticeable difference is in the cost scores which increase as the ranking quality declines, with a particular jump between mid and low quality rankings. For example, for the CLEF2017 data set with target recall 1.0 the cost increases from 0.226 to 0.228 when moving from high to mid quality rankings but then to 0.423 for low quality. The reason for this increase is that the relevant documents occur later in poorer rankings, forcing GRLStop to progress further down the ranking before stopping.

GRLStop displays some robustness to ranking quality since this does not appear to affect reliability scores which, although they vary with ranking quality, do not always decrease for lower quality rankings (e.g. CLEF2018 dataset for target recalls 0.9 and 0.8). However, there is a general trend for the CostDiff scores to increase as ranking quality decreases, indicating that the approach is forced to examine more documents than necessary before stopping. This is probably because lower quality rankings represent a more noisy environment and therefore a more challenging one for the RL algorithm to learn a good policy for.

6 CONCLUSION

This paper introduces a generalised and adaptable RL-based stopping method for TAR stopping approach. Unlike previous RL-based approaches, the proposed approach allows a single model to be applied to multiple target recall levels and the balance between minimising cost versus achieving target recall to be controlled. It also demonstrates how the output from a text classifier can be used within an RL-based stopping framework.

Results on several benchmark datasets showed that the proposed approach proved to be effective compared to multiple baselines including a previous RL-based method. Further experiments demonstrated that the approach achieved its goals of being applicable to multiple target recall levels and controlling objectives. The integration of text classification was also found to be beneficial.

Possibilities for future work include making use of curriculum learning to reduce the time required to train RL models [4] and experimenting with LLMs to produce relevance judgments for the unexamined batches as an alternative to training a text classifier.

REFERENCES

[1] Afsar, M.M., Crump, T., Far, B.: Reinforcement Learning Based Recommender Systems: A Survey. ACM Computing Surveys **55**(7), 1–38 (2022)

[2] Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep Reinforcement Learning: A Brief Survey. IEEE Signal Processing Magazine **34**(6), 26–38 (2017). https://doi.org/10.1109/MSP.2017.2743240

[3] Baron, J.R., Sayed, M.F., Oard, D.W.: 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)

[4] Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum Learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 41–48 (2009)

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

[6] Bin-Hezam, R., Stevenson, M.: RLStop: A Reinforcement Learning Stopping Method for TAR. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (2024)

[7] Bron, M.P., van der Heijden, P.G., Feelders, A.J., Siebes, A.P.: Using Chao's Estimator as a Stopping Criterion for Technology-Assisted Review. arXiv preprint arXiv:2404.01176 (2024)

[8] Callaghan, M.W., Müller-Hansen, F.: Statistical Stopping Criteria for Automated Screening in Systematic Reviews. Systematic Reviews **9**(1), 1–14 (2020)

[9] Cormack, G., Grossman, M.: Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review. arXiv preprint arXiv:1504.06868 (apr 2015)

[10] Cormack, G.V., Grossman, M.R.: Engineering Quality and Reliability in Technology-Assisted Review. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 75–84 (2016)

[11] Cormack, G.V., Grossman, M.R.: 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. pp. 1039–1048 (2016)

[12] Cormack, G.V., Grossman, M.R., Hedin, B., Oard, D.W.: Overview of the TREC 2010 Legal Track. In: Proceedings of The Nineteenth Text REtrieval Conference, TREC 2010, Gaithersburg, Maryland, USA, November 16-19, 2010. NIST Special Publication, vol. 500-294. National Institute of Standards and Technology (NIST) (2010)

[13] Di Nunzio, G.M.: A Study of an Automatic Stopping Strategy for Technologically Assisted Medical Reviews. In: European Conference on Information Retrieval. pp. 672–677. Springer (2018)

[14] Gray, L., Lewis, D.D., Pickens, J., Yang, E.: High Recall Retrieval Via Technology-Assisted Review. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2987–2988 (2024)

[15] Grossman, M.R., Cormack, G.V., Roegiest, A.: 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) (2016)

[16] Higgins, J.P., Thomas, J., Chandler, J., Cumpston, M., Li, T., Page, M.J., Welch, V.A.: Cochrane Handbook for Systematic Reviews of Interventions. John Wiley & Sons (2019)

[17] Hollmann, N., Eickhoff, C.: Ranking and Feedback-based Stopping for Recall-Centric Document Retrieval. In: Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum. pp. 7–8 (2017)

[18] Horvitz, D.G., Thompson, D.J.: A Generalization of Sampling without Replacement from a Finite Universe. Journal of the American Statistical Association **47**(260), 663–685 (1952)

[19] Howard, B.E., Phillips, J., Tandon, A., Maharana, A., Elmore, R., Mav, D., Sedykh, A., Thayer, K., Merrick, B.A., Walker, V., Rooney, A., Shah, R.R.: SWIFT-Active Screener: Accelerated Document Screening through Active Learning and Integrated Recall Estimation. Environment International **138**, 105623 (2020), https://www.sciencedirect.com/science/article/pii/S0160412019314023

[20] Kanoulas, E., Li, D., Azzopardi, L., Spijker, R.: CLEF 2017 Technologically Assisted Reviews in Empirical Medicine Overview. In: CEUR workshop proceedings. vol. 1866 (2017)

[21] Kanoulas, E., Li, D., Azzopardi, L., Spijker, R.: CLEF 2018 Technologically Assisted Reviews in Empirical Medicine Overview. In: CEUR workshop proceedings. vol. 2125 (2018)

[22] Kanoulas, E., Li, D., Azzopardi, L., Spijker, R.: CLEF 2019 Technology Assisted Reviews in Empirical Medicine Overview. In: CEUR workshop proceedings. vol. 2380 (2019)

[23] Lewis, D., Yang, E., Frieder, O.: Certifying One-Phase Technology-Assisted Reviews. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management (2021)

[24] Lewis, D.D., Yang, Y., Russell-Rose, T., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research **5**, 361–397 (2004). https://doi.org/10.5555/1005332.1005345

[25] Li, D., Kanoulas, E.: 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), 1–36 (2020), https://doi.org/10.1145/3411755

[26] Ling, C.X., Sheng, V.S.: Cost-Sensitive Learning and the Class Imbalance Problem. Encyclopedia of machine learning **2011**, 231–235 (2008)

[27] Losada, D.E., Parapar, J., Barreiro, A.: 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), 49–60 (2019)

[28] Mcdonald, G., Macdonald, C., Ounis, I.: How the Accuracy and Confidence of Sensitivity Classification Affects Digital Sensitivity Review. ACM Transactions on Information Systems (TOIS) **39**(1), 1–34 (2020)

[29] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing Atari with Deep Reinforcement Learning (2013)

[30] Molinari, A., Esuli, A.: SALτ: Efficiently Stopping TAR by Improving Priors Estimates. Data Mining and Knowledge Discovery pp. 1–34 (2023)

[31] Montazeralghaem, A., Zamani, H., Allan, J.: A Reinforcement Learning Framework for Relevance Feedback. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 59–68 (2020)

[32] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-Baselines3: Reliable Reinforcement Learning Implementations. Journal of Machine Learning Research **22**(268), 1–8 (2021), http://jmlr.org/papers/v22/20-1364.html

[33] Ren, Z., Huang, N., Wang, Y., Ren, P., Ma, J., Lei, J., Shi, X., Luo, H., Jose, J., Xin, X.: Contrastive State Augmentations for Reinforcement Learning-Based Recommender Systems. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 922–931 (2023)

[34] Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B.: Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In: Proceedings of the 31st International Conference on Distributed Computing Systems workshops. pp. 166–171. IEEE (2011)

[35] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347 (2017)

[36] Shemilt, I., Simon, A., Hollands, G.J., Marteau, T.M., Ogilvie, D., O'Mara-Eves, A., Kelly, M.P., Thomas, J.: Pinpointing needles in giant haystacks: use of text mining to reduce impractical screening workload in extremely large scoping reviews. Research Synthesis Methods **5**(1), 31–49 (2014)

[37] Stevenson, M., Bin-Hezam, R.: Stopping Methods for Technology-assisted Reviews Based on Point Processes. ACM Transactions on Information Systems **42**(3), 1–37 (2023), https://doi.org/10.1145/3631990

[38] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (2018)

[39] Thompson, S.: Sampling. John Wiley & Sons, Hoboken, New Jersey (2012)

[40] Towers, M., Terry, J.K., Kwiatkowski, A., Balis, J.U., Cola, G.d., Deleu, T., Goulão, M., Kallinteris, A., KG, A., Krimmel, M., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J.J., Shen, A.T.J., Younis, O.G.: Gymnasium (Mar 2023), https://zenodo.org/record/8127025

[41] Williams, R.J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. Machine Learning **8**, 229–256 (1992)

[42] Xin, X., Karatzoglou, A., Arapakis, I., Jose, J.M.: Self-Supervised Reinforcement Learning for Recommender Systems. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 931–940 (2020)

[43] Yang, E., Lewis, D., Frieder, O.: Heuristic Stopping Rules for Technology-Assisted Review. In: Proceedings of the 21st ACM Symposium on Document Engineering 2021 (DocEng '21). pp. 1–10 (2021). https://doi.org/10.1145/3469096.3469873

[44] Yang, E., Lewis, D.D., Frieder, O.: On Minimizing Cost in Legal Document Review Workflows. In: Proceedings of the 21st ACM Symposium on Document Engineering 2021 (DocEng '21). pp. 1–10 (2021). https://doi.org/10.1145/3469096.3469872

[45] Yang, E., Lewis, D.D., Frieder, O.: TAR on Social Media: A Framework for Online Content Moderation. In: 2nd International Conference on Design of Experimental Search & Information REtrieval Systems (DESIRES 2021). pp. 147–155 (2021)

[46] Yu, Z., Menzies, T.: FAST2: An Intelligent Assistant for Finding Relevant Papers. Expert Systems with Applications **120**, 57–71 (2019)