Certified Error Control of Candidate Set Pruning for Two-Stage Relevance Ranking

Minghan Li[*], **Xinyu Zhang**[*], **Ji Xin, Hongyang Zhang,** and **Jimmy Lin**

David R. Cheriton School of Computer Science
University of Waterloo

{m692li,x978zhang,ji.xin,hongyang.zhang,jimmylin}@uwaterloo.ca

Abstract

In information retrieval (IR), candidate set pruning has been commonly used to speed up two-stage relevance ranking. However, such an approach lacks accurate error control and often trades accuracy against computational efficiency in an empirical fashion, missing theoretical guarantees. In this paper, we propose the concept of *certified error control* of candidate set pruning for relevance ranking, which means that the test error after pruning is guaranteed to be controlled under a user-specified threshold with high probability. Both in-domain and out-of-domain experiments show that our method successfully prunes the first-stage retrieved candidate sets to improve the second-stage reranking speed while satisfying the pre-specified accuracy constraints in both settings. For example, on MS MARCO Passage v1, our method reduces the average candidate set size from 1000 to 27, increasing reranking speed by about 37 times, while keeping MRR@10 greater than a pre-specified value of 0.38 with about 90% empirical coverage. In contrast, empirical baselines fail to meet such requirements. Code and data are available at: https://github.com/alexlimh/CEC-Ranking.

1 Introduction

A two-stage relevance ranking architecture has been an indispensable component for knowledge-intensive natural language processing tasks such as information retrieval (IR) (Manning et al., 2008) and open-domain question answering (OpenQA) (Chen et al., 2017). Such a system usually consists of a high-recall first stage that retrieves a set of documents from a massive corpus and a high-precision reranker that improves the ranking of the retrieved candidate sets. The first-stage retrieval, often implemented by approximate nearest neighbour search (Johnson et al., 2021) or inverted index search (Lin et al., 2021), is quite efficient while the second-stage reranking usually has high latency due to the trend of using over-parameterized pre-trained language models and large candidate set sizes. Previous work in early exiting proposed to predict the ranking score using only a partial model (Xin et al., 2020a; Lucchese et al., 2020; Busolin et al., 2021) or prune the candidate set before reranking (Wang et al., 2011; Fisch et al., 2021) to trade accuracy off against speed. However, such methods lack accurate error control which can not provide guarantees to satisfy the exact accuracy constraints specified by users.

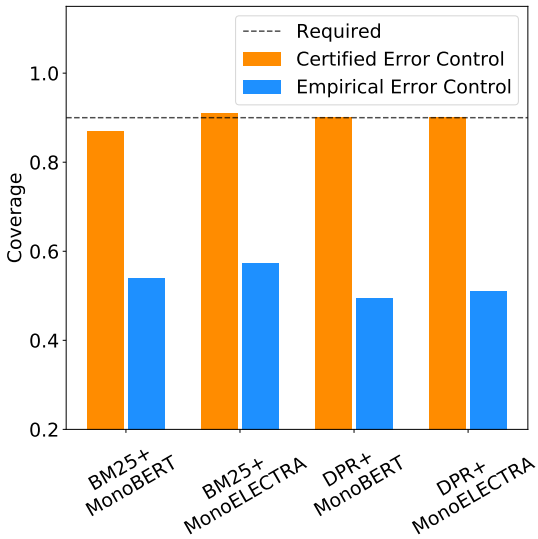


Figure 1: Comparisons between certified and empirical error control methods on MS MARCO Passage v1. Coverage: Percentage of 100 independent runs that satisfies a pre-specified MRR@10$\geq$0.35 (the dotted line).

In this paper, we focus on *candidate set pruning* methods of early exiting for two-stage relevance ranking, and we show that a simple score-thresholding method can yield predictions with *certified* error control using the prediction sets theory (Wilks, 1941, 1942; Wald, 1943; Bates et al., 2021). Instead of predicting a single target, prediction sets will yield sets that contain the desired fraction of the population with high probability.

[*] Equal contribution

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 333–345
December 7-11, 2022

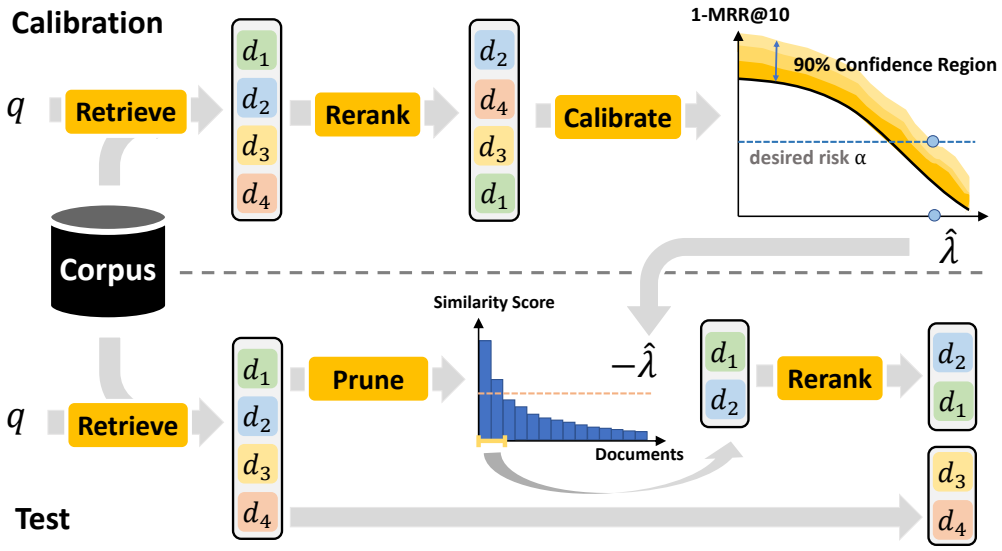


Figure 2: Calibration and test procedures of candidate set pruning with certified error control. During calibration, we use the confidence region and desired risk α to find the threshold $-\hat{\lambda}$. During testing, we use the threshold to prune the retrieved documents, which guarantees control of the expected loss under α with probability $1-\delta$.

Moreover, we allow users to specify the error tolerance of their custom metrics and our method will return the pruned set of candidates that satisfies the constraints with a finite sample guarantee. Our method makes no assumption about the data distributions and models, except that the calibration data are exchangeable with the test data. Fig. 2 illustrates the calibration and test procedures for certified error control of candidate set pruning.

Challenges. However, directly applying prediction set methods to relevance ranking can be problematic. Unlike classification, where the true label will be eventually included in the predicted set as the set size grows, in relevance ranking, users care more about the *rank* of the positives. Therefore, a rank requirement from users might not always be satisfied as the ranking system might not be good enough to rank the positive documents correctly no matter how large the candidate set is.

Contributions. To this end, we propose to correct the coverage level pre-specified by the user if the constraints are impossible to satisfy, and it is up to the user to decide whether to abandon the prediction or accept the corrected results. Empirically, we evaluate our method on IR and OpenQA benchmarks, including MS MARCO Passage v1 (Nguyen et al., 2016) and Quora (Thakur et al., 2021). We also test different combinations of retrievers and rerankers for the two-stage relevance ranking system under both in-domain and out-of-domain settings. Fig. 1 shows the results of certified and empirical error control methods using different ranking systems on MS MARCO Passage v1, and we can see that the empirical method fails to provide the required coverage while our method succeeds to meet the requirement (see Tbl. 3 for more details). For example, if we pre-specify MRR@10 $\geq$ 0.38, we can reduce the average candidate set size from 1000 to 27, increasing the reranking speed by 37$\times$ while satisfying the constraint with 90% empirical coverage. We further confirm that the risk-confidence correction of our method is able to consistently correct the risk/confidence when the pre-specified conditions are impossible to achieve.

To sum up, our contributions are three-fold:

- We propose the first certified error control method of candidate set pruning for relevance ranking, providing a guarantee to satisfy user-specified constraints with high probability.
- We propose a risk-confidence correction method to adjust constraints that may be otherwise impossible to satisfy for ranking tasks.
- Our method achieves consistent results under both in-domain and out-of-domain settings. With at least 90% coverage, our method returns a candidate set size less than 50 with $1 \sim 2\%$ accuracy drop for most two-stage ranking systems.

2 Related Work

Early exiting for relevance ranking. Early exiting (Xin et al., 2020b; Liu et al., 2020; Xin et al., 2021) is a popular latency-accuracy tradeoff method in document ranking. Xin et al. (2020a) and Soldaini and Moschitti (2020) proposed to output the question-document similarity score at earlier layers of a pre-trained language model, while Lucchese et al. (2020) and Busolin et al. (2021) proposed to use a set of decision trees in the ensemble for prediction. Cambazoglu et al. (2010) proposed optimization strategies that allow short-circuiting score computations in additive learning systems. Wang et al. (2011) presented a boosting algorithm for learning such cascades to optimize the tradeoff between effectiveness and efficiency. Despite their popularity, the above early exiting methods mainly use fixed rules for efficiency-accuracy tradeoffs without performance guarantees.

Prediction sets and cascade systems. Prediction sets are essentially tolerance regions (Wilks, 1941, 1942; Wald, 1943), which are sets that contain the desired fraction of the collection with high probability. Recently, tolerance regions have been applied to yield prediction sets for deep learning models (Park et al., 2020a, 2021; Bates et al., 2021). In addition, conformal prediction (Vovk et al., 1999, 2005) has been recognized as an attractive way of producing predictive sets with finite-sample guarantees. In retrieval, structured prediction cascades (Weiss and Taskar, 2010) optimize their cascades for overall pruning efficiency, and Fisch et al. (2021) proposed a cascade system to prune the unnecessarily large conformal prediction sets for OpenQA. However, conformal prediction is only suitable for metrics like recall. Other works on inverted file systems also provide correctness guarantees on keyword and document pruning (Ntoulas and Cho, 2007).

3 Background

3.1 Notation

In the rest of the paper, we will use upper-case letters (e.g., $Q, D, \ldots$) to denote the random variables, script letters (e.g., $\mathcal{Q}, \mathcal{D}, \ldots$) to denote the event space, and lower-case letters (e.g., $q, d, \ldots$) to denote an actual value of a random variable in its event space. Specially, we use $\mathcal{X}'$ to denote the space of all possible subsets in $\mathcal{X}$ where $\mathcal{X}' = 2^{\mathcal{X}}$.

3.2 Two-Stage Relevance Ranking

Given a question q, the relevance ranking task is to return a sorted list of documents from a large text corpus to maximize a metric of interest. Particularly, in a two-stage ranking system, a first-stage retriever generates a set of candidate documents $\{d_1, d_2, ..., d_k\}$ for the second-stage reranker to reorder the candidate lists.

3.2.1 Retrieval

Dense Retrievers encode the question and documents separately and project them into a low-dimensional space (e.g., 768 dim, which is "lower" than the size of the corpus vocabulary). Representative methods include DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2021), ColBERT (Khattab and Zaharia, 2020), and MeBERT (Luan et al., 2021).

Lexical/Sparse Retrievers use the corpus vocabulary as the basis for vector representations. Static methods include tf–idf (Salton and Buckley, 1988) and BM25 (Robertson and Zaragoza, 2009). Contextualized methods include SPLADE (Formal et al., 2021), DeepCT (Dai and Callan, 2020), DeepImpact (Mallia et al., 2021), and COIL (Gao et al., 2021; Lin and Ma, 2021).

Despite their differences, all the above methods can be viewed as a logical scoring model (Lin, 2021). Let $\eta_Q : \mathcal{Q} \to \mathbb{R}^n$ be an arbitrary function that maps a question to an n dimensional vector representation, and let $\eta_D : \mathcal{D} \to \mathbb{R}^n$ be an arbitrary function that maps a document to an n dimensional vector. The similarity score s_v between a question q and a document d can be defined as:

$$s_v(q, d) \doteq \phi(\eta_q(q), \eta_d(d)), \tag{1}$$

where ϕ is a metric that measures the similarity between encoded vectors of $\eta(q)$ and $\eta(d)$, such as dot product or cosine similarity.

3.2.2 Reranking

The reranker module is responsible for improving the ranking quality of the candidate documents returned from the first-stage retrievers. We focus on recent neural rerankers based on pre-trained language models such as MonoBERT (Nogueira et al., 2019) and MonoELECTRA. The reranker could also be seen as a logical scoring model. However, instead of using a bi-encoder structure, a cross-encoder structure is often applied, where the question-document pairs are encoded and fed into a

single model together for more fine-grained token-level interactions:

$$s_r(q, d) \doteq \zeta(\text{concat}(q, d)), \tag{2}$$

where ζ is the reranker that takes the question-document pair as input and outputs the similarity score s_r. One way to implement the "concat" function is using special tokens as indicators, such as [CLS] q [SEP] d [SEP].

4 Methods

4.1 Settings

Formally, given a question $Q \in \mathcal{Q}$, a set of documents $D' \in \mathcal{D}'$ retrieved by the first-stage retriever ϕ, and a relevance-judged set of gold documents $D'_\omega \in \mathcal{D}'_\omega$, we consider a pruning function $\mathcal{T} : \mathcal{D}' \to \mathcal{P}'$, where $\mathcal{P}'$ denotes the space of subsets of D'. We then use a loss function on the pruned document sets that also depends on the reranker ζ, i.e., $L(\cdot, \cdot; \zeta) : \mathcal{D}'_\omega \times \mathcal{P}' \to \mathbb{R}$, to encode a metric of the user's interest, and seek a pruning function $\mathcal{T}$ that controls the risk (i.e., error) $R(\mathcal{T}; \zeta) = \mathbb{E}[L(D'_\omega, \mathcal{T}(D'); \zeta)]$.

Definition 1 (Certified Error Control of Candidate Set Pruning)**.** *Let $\mathcal{T} : \mathcal{D}' \to \mathcal{P}'$ be a random function. We say that $\mathcal{T}$ is a pruning function for reranker ζ with certified error control if, with probability at least $1 - \delta$, we have $R(\mathcal{T}; \zeta) \leq \alpha$.*

The risk level $\alpha > 0$ is pre-specified by users, and the same goes for $\delta \in (0, 1)$ where $0.1/0.01$ is often chosen as a rule of thumb.

4.1.1 Pruning Function and Risk Function

We use a calibration set to certify the error control of the pruning function and apply the certified pruner during testing. Let $\{Q_i, D'_i, D'_{\omega i}\}_{i=1}^m$ be an i.i.d. sampled set of random variables representing a calibration set of queries, candidate document sets, and gold document sets. For the pruning function, we define a parameter $\lambda \in \Lambda$ as its index, which is essentially a score threshold with the following property:

$$\lambda_1 < \lambda_2 \Rightarrow \mathcal{T}_{\lambda_1}(d') \subset \mathcal{T}_{\lambda_2}(d'). \tag{3}$$

Let $L(D'_\omega, P'; \zeta) : \mathcal{D}'_\omega \times \mathcal{P}' \to \mathbb{R}_{\geq 0}$ be a loss function on the pruned subsets. In ranking, we could take, for instance, $L(D'_\omega, P'; \zeta) = 1 - \text{MRR@10}(D'_\omega, P'_\zeta)$, where MRR@10 is a popular measurement of ranking quality and P'_ζ is the reranked version P' by ζ. In general, the loss function has the following nesting property:

$$P'_1 \subset P'_2 \Rightarrow L(D'_\omega, P'_1; \zeta) \geq L(D'_\omega, P'_2; \zeta). \tag{4}$$

That is, larger sets lead to smaller losses (i.e., monotonicity) (Bates et al., 2021). We then define the risk of a pruning function $\mathcal{T}_\lambda$ to be

$$R(\mathcal{T}_\lambda; \zeta) = \mathbb{E}[L(D'_\omega, P'; \zeta)].$$

4.1.2 Confidence Region

In practice, to find the parameter λ, we need to search across the collection of functions $\{\mathcal{T}_\lambda\}_{\lambda \in \Lambda}$ and estimate their risks on the calibration set. However, the true risk is often unknown and the empirical risk function is often used as an approximation:

$$\widehat{R}(\mathcal{T}_\lambda; \zeta) = \frac{1}{m} \sum_{i=1}^{m} L(D'_{\omega i}, \mathcal{T}_\lambda(D'_i); \zeta).$$

To compute the confidence region, we leverage the concentration inequalities and assume that we have access to a pointwise confidence region for the risk function for each λ:

$$\Pr(R(\mathcal{T}_\lambda; \zeta) \leq \widehat{R}^+_\delta(\mathcal{T}_\lambda; \zeta)) \geq 1 - \delta, \tag{5}$$

where $\widehat{R}^+_\delta(\mathcal{T}_\lambda; \zeta)$ is the upper bound of the empirical risk $\widehat{R}(\mathcal{T}_\lambda; \zeta)$. Bates et al. (2021) presented a generic strategy to obtain such bounds by inverting a concentration inequality as well as concrete bounds for various settings. For this paper, we use the Waudby-Smith-Ramdas (WSR) bound (Waudby-Smith and Ramdas, 2020) which is adaptive to variance. We provide the specific form of the WSR bound in Appendix A.2.

We choose the smallest λ such that the entire confidence region to the right of λ falls below the target risk level α following Bates et al. (2021):

$$\hat{\lambda} \doteq \inf \left\{ \lambda \in \Lambda : \widehat{R}^+_\delta(\mathcal{T}_{\lambda'}; \zeta) < \alpha, \forall \lambda' \geq \lambda \right\}. \tag{6}$$

In this way, $\mathcal{T}_{\hat{\lambda}}$ is a pruning function with certified error control. Theorems and proofs are provided in Appendix A.1. In the following sections, we will discuss the problems of truncated risk functions in relevance ranking and how to modify the certification for impossible constraints.

4.2 Truncated Risk Function for Ranking

In Section 4.1, we mentioned that the risk function is often related to the metrics that we care about. For example, in ranking, metrics such as MRR@K are often used to assess the ranking quality, where K is the maximal set size that we choose. For example, the MRR@10 score given a set of questions $\{q_i\}_{i=1}^m$, the positive document sets $\{d'_i\}_{i=1}^m$, and the retrieved document candidate sets $\{d'_{\omega i}\}_{i=1}^m$ is

$$\text{MRR@10} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{f(d'_i, d'_{\omega i})}, \tag{7}$$

where

$$f(d'_i, d'_{\omega i}) = \begin{cases} +\infty, & \text{if } r_i > 10; \\ r_i, & \text{otherwise,} \end{cases} \tag{8}$$

and

$$r_i = \min_j(r(d'_i, d'_{\omega ij})).$$

$d'_{\omega ij}$ means the j^{th} positive document for query i and $r(d'_i, d'_{\omega ij})$ means the rank of $d'_{\omega ij}$ in the candidate set d'_i. If we use $1-$MRR@10 for the empirical risk function $\widehat{R}(\mathcal{T}_\lambda;\zeta)$, we can see that the risk function and its upper-bound plateaus after a certain λ value (Fig. 3b) due to the threshold function in Eq (8). Therefore, naive certification will fail if the risk level is specified too low.

4.3 Correction for Risk Level and Confidence

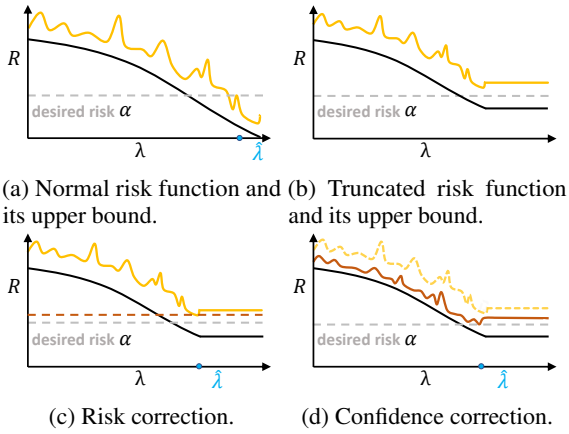
To this end, we propose a safe certification method, which will automatically correct the risk level α or the confidence $1-\delta$ if the risk level specified by the user is too low. Given a specific (α, δ) pair, the basic idea is that if the minimum of the upper confidence bound $\widehat{R}^+_\delta(\mathcal{T}_\lambda;\zeta)$ over all possible λ values is bigger than the specified risk level α, we will either replace the risk level with the best-possible minimal risk that we could achieve:

$$\alpha_c \doteq \inf_{\hat{\lambda}\in\Lambda} \widehat{R}^+_\delta(\mathcal{T}_{\hat{\lambda}};\zeta), \tag{9}$$

where α_c is the calibrated risk level as shown in Fig. 3c, or increase δ to shrink the upper-confidence bound until $\delta = 1$:

$$\delta_c \doteq \inf\left\{\delta \in (0,1] : \inf_{\hat{\lambda}\in\Lambda} \widehat{R}^+_\delta(\mathcal{T}_{\hat{\lambda}};\zeta) \le \alpha\right\}, \tag{10}$$

where δ_c is the calibrated significance level as shown in Fig. 3d. Therefore, the previous pruning function found in Eq (6) does not necessarily hold for some specific (α, δ) values in ranking, and we propose a new theorem for the corrected version of certification:



(a) Normal risk function and its upper bound.

(b) Truncated risk function and its upper bound.

(c) Risk correction.

(d) Confidence correction.

Figure 3: (Corrected) Confidence region of the risk function. (a) shows the risk function for metrics like recall; (b) shows the risk function that is truncated in ranking. (c) and (d) show two different types of corrections: Either setting the risk level to the minimum of the upper bound or increasing δ to shrink the upper bound.

Theorem 1 (Correction of Certified Error Control)**.** *In the setting of Section 4.1, assume that there exists $\alpha > 0$ and $\delta > 0$ such that for every $\hat{\lambda} \in \Lambda$, $\widehat{R}^+_\delta(\mathcal{T}_{\hat{\lambda}};\zeta) > \alpha$. In this case, $\mathcal{T}_{\hat{\lambda}}$ is no longer a pruning function with certified error control. Instead, with the corrected risk level α_c and confidence δ_c in Eq (9) and (10), there exists $\hat{\lambda} \in \Lambda$ such that,*

$$\Pr(R(\mathcal{T}_{\hat{\lambda}};\zeta) \le \alpha_c) \ge 1-\delta,$$

or

$$\Pr(R(\mathcal{T}_{\hat{\lambda}};\zeta) \le \alpha) \ge 1-\delta_c.$$

In both cases, $\mathcal{T}_{\hat{\lambda}}$ is a pruning function with certified error control.

Proofs are provided in Appendix A.1. In addition, we provide an algorithmic implementation in Appendix A.3 for the readers' further reference.

5 Experimental Setup

5.1 Datasets

In this paper, we evaluate our method on the following datasets: MS MARCO Passage v1 (Nguyen et al., 2016) contains 8.8M English passages with an average length of around 55 tokens, which is a standard retrieval benchmark for comparing in-domain results. Quora Duplicate Ques-

Retriever	MARCO	Quora
BM25	0.185	0.781
DPR	0.311	0.434
UniCOIL	0.348	0.659

Table 1: In-domain (MS MARCO Passage v1) and out-of-domain (Quora) first-stage retrieval test MRR@10 scores averaged over 100 random dev/test splits.

Retriever	MARCO	Quora
BM25+MonoBERT	0.369	0.840
DPR+MonoBERT	0.378	0.728
UniCOIL+MonoBERT	0.383	0.832
BM25+MonoELECTRA	0.399	0.823
DPR+MonoELECTRA	0.415	0.653
UniCOIL+MonoELECTRA	0.415	0.817

Table 2: In-domain (MS MARCO Passage v1) and out-of-domain (Quora) second-stage reranking (with evidence fusion) test MRR@10 scores averaged over 100 random dev/test splits.

tions[1] (Thakur et al., 2021) contains 522K passages with an average length of around 11 tokens and mostly consist of duplicate entity questions that were found to be challenging for out-of-domain generalization of neural retrievers like DPR (Sciavolino et al., 2021). The above datasets label documents with shallow judgements (Yilmaz and Robertson, 2009) for each query and therefore MRR@10 is often used as the evaluation metric. Other datasets such as TREC DL 2019 (Voorhees and Ellis, 2019) use metrics like nDCG, but such densely labelled data are very scarce and therefore they are not suitable for finite-sample calibration, which we leave for future work.

5.2 Retrievers and Rerankers

For retrievers, we use DPR (dense retriever), BM25 (static lexical retriever), and UniCOIL (contextualized lexical retriever). For rerankers, we use two cross-encoder models, MonoBERT (Nogueira et al., 2019) and MonoELECTRA (Pradeep et al., 2022). Although some of these models are no longer state of the art, they remain competitive and have been widely adopted by researchers in the community as points of reference. For the pipeline, we use the retriever (e.g., DPR) to retrieve the top-1000 candidates from the corpus, and then use the reranker (e.g., MonoELECTRA) to rerank the retrieved candidate sets. We believe the above choices cover the basic types of modern two-stage ranking systems; our approach is model agnostic and can be easily applied to other models as well. For implementation, we use off-the-shelf pre-trained models from Pyserini (Lin et al., 2021) and Caprelous (Yates et al., 2020).

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

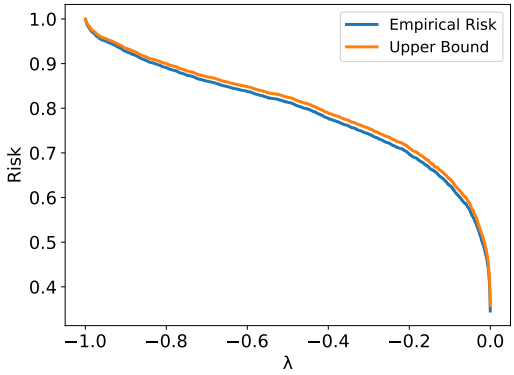


Figure 4: Empirical risk function and its upper bound on one calibration-test data split.

Finally, for rerankers based on neural networks, we need to consider both in-domain and out-of-domain situations, as it is possible that the reranker overfits to certain domains and has worse out-of-domain effectiveness than the retriever. To solve this, we linearly interpolate the score of the retriever ϕ and the reranker ζ for each (q, d) pair during reranking:

$$s_f(q,d) = \beta \cdot \phi(\eta_q(q), \eta_d(d)) + (1-\beta) \cdot \zeta(\text{concat}(q,d)),$$

which is known as evidence fusion (Ma et al., 2022). The weight $\beta \in [0, 1]$ is searched on the calibration set for the best MRR@10 score, such that the fusion model will consistently yield better ranking results than both ζ and ϕ. Tbl. 1 and 2 show the retrieval and reranking effectiveness with evidence fusion under both in-domain and out-of-domain settings.

5.3 Baselines

We modify two *empirical error control* methods from Cambazoglu et al. (2010) as the baselines:

Empirical Score Threshold (EST). A score threshold on the calibration set such that the pruned MRR@10 just meets the required score.

Empirical Rank Threshold (ERT). Similar to EST, but we tune a threshold on the rank of the documents to prune the candidate set instead.

Methods	MRR@10	Confidence	Coverage	Size
BM25 + MonoBERT (required MRR@10=0.350)				
CEC	0.359	0.870	**0.870**	451
EST	0.350	-	0.540	139
ERT	0.350	-	0.510	149
UniCOIL + MonoBERT (required MRR@10=0.350)				
CEC	0.360	0.900	**0.900**	15
EST	0.350	-	0.480	9
ERT	0.352	-	0.460	7
DPR + MonoBERT (required MRR@10=0.350)				
CEC	0.360	0.900	**0.900**	22
EST	0.350	-	0.500	11
ERT	0.353	-	0.730	11
BM25 + MonoELECTRA (required MRR@10=0.380)				
CEC	0.389	0.894	**0.910**	602
EST	0.381	-	0.580	280
ERT	0.381	-	0.550	246
UniCOIL + MonoELECTRA (required MRR@10=0.380)				
CEC	0.390	0.900	**0.910**	18
EST	0.380	-	0.520	13
ERT	0.383	-	0.750	9
DPR + MonoELECTRA (required MRR@10=0.380)				
CEC	0.389	0.900	**0.900**	27
EST	0.381	-	0.580	16
ERT	0.382	-	0.580	17

Methods	MRR@10	Confidence	Coverage	Size
BM25 + MonoBERT (required MRR@10=0.780)				
CEC	0.789	0.900	**0.910**	2
EST	0.780	-	0.510	2
ERT	0.780	-	0.600	3
UniCOIL + MonoBERT (required MRR@10=0.780)				
CEC	0.790	0.900	**0.900**	16
EST	0.780	-	0.560	13
ERT	0.782	-	0.640	14
DPR + MonoBERT (required MRR@10=0.620)				
CEC	0.632	0.900	**0.940**	40
EST	0.620	-	0.500	30
ERT	0.621	-	0.490	29
BM25 + MonoELECTRA (required MRR@10=0.780)				
CEC	0.790	0.900	**0.900**	3
EST	0.780	-	0.460	3
ERT	0.780	-	0.560	2
UniCOIL + MonoELECTRA (required MRR@10=0.780)				
CEC	0.790	0.900	**0.910**	3
EST	0.785	-	0.540	3
ERT	0.782	-	0.600	3
DPR + MonoELECTRA (required MRR@10=0.620)				
CEC	0.634	0.900	**0.950**	282
EST	0.620	-	0.530	155
ERT	0.620	-	0.540	149

Table 3: In-domain results on MS MARCO Passage v1 (left) and out-of-domain results on Quora (right). CEC: Certified error control. EST: Empirical score threshold. ERT: Empirical rank threshold. Confidence: $1 - \delta_c$ as in Eq (10). Coverage: Proportion of 100 runs that satisfy the risk constraints. Size: Average candidate set size out of 1,000. See Section 6.2 for details.

6 Results

6.1 Risk Function and Upper Bound

Fig. 4 shows the empirical risk function and its WSR upper bound of the pruning function $\mathcal{T}_\lambda$ on Quora using the DPR + MonoELECTRA ranking system. We can see that the empirical risk function is a monotone function and the minimum of the risk is greater than 0, which is consistent with the assumptions we made in Section. 4. In addition, we can see that the bound is very tight, providing a good estimation of the true risk.

6.2 In-Domain Results

Empirically, if we set the risk threshold $\alpha = 0.62$ and confidence level $1 - \delta = 0.9$, then there should be at least 90% of independent runs (i.e., coverage) for which the MRR@10 score is greater than 0.38 (i.e., $1 - \alpha$). To verify this, we mix the test set and dev set and then randomly sample a calibration set of size 5,000 and a test set of size 6,980, repeating for 100 trials.

Tbl. 3 (left) shows the MRR@10 score, corrected confidence, empirical coverage, and average candidate set size on MS MARCO Passage v1. We choose different performance thresholds for different ranking systems such that the final reranking score is around $1 \sim 2\%$ less than the highest obtainable score, which is a very typical setting in real-world applications. We use $\delta = 0.1$ for all experiments, but it could be corrected if the risk threshold is unable to be satisfied. For example, for BM25 + MonoBERT, the confidence is corrected from 0.90 to 0.87, which is more consistent with the empirical coverage.

We can see that our method achieves the required risk constraint with the required coverage (i.e., confidence) for multiple ranking systems, while the average candidate set size is also drastically reduced from 1,000 to less than 50. In comparison, although they are able to obtain smaller candidate set sizes, both empirical error control methods do not achieve the expected coverage. Despite the fact that we can choose other thresholds to achieve better coverage, it is unclear how much accuracy should be sacrificed in order to achieve the required coverage.

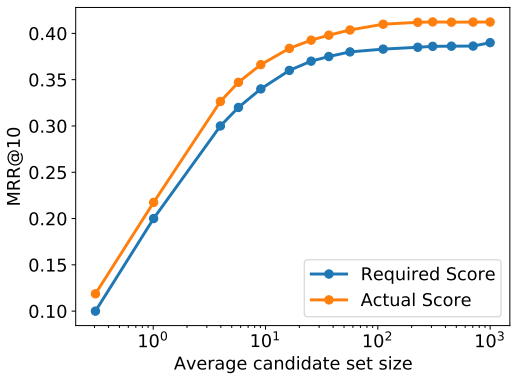


Figure 5: Tradeoffs between candidate set size and MRR@10 scores on MS MARCO Passage v1 based on the DPR + MonoELECTRA ranking system. The empirical coverage for each point is greater than 90%.

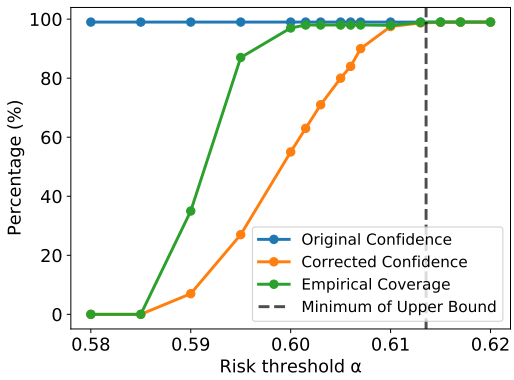


Figure 6: Confidence correction on MS MARCO Passage v1 using DPR + MonoELECTRA. The x-axis represents different risk thresholds (i.e., α) and the y-axis represents the percentage.

6.3 Out-of-Domain Results

We also test our method on Quora under an out-of-domain setting, where we use the retrievers and rerankers trained on MS MARCO Passage v1 as the prediction models. We can see from Tbl. 1 and Tbl. 2 that the out-of-domain retrieval and reranking results are drastically different from the MS MARCO dataset, where BM25 outperforms the other neural retrievers. This is because the Quora dataset mostly consists of duplicate, entity-based questions, which are naturally biased toward static lexical retrievers.

Similar to the in-domain experiments, we calibrate the pruning function on the calibration set with 5,000 data points and test it on a test set with 10,000 data points over 100 trial runs. Tbl. 3 (right) shows the MRR@10 scores of different ranking systems. However, unlike the in-domain setting, the ranking effectiveness of the first stage retrievers varies a lot under the out-of-domain setting and it is hard to align the pruning results for intuitive comparison. Therefore, we set different MRR@10 thresholds such that the effectiveness drop is around $1 \sim 10\%$ to align the results of different ranking systems. The results are similar to the in-domain experiments, where the certified error control method manages to provide a performance guarantee while the empirical method fails to do so. Our method also yields reasonable set sizes that are close to the empirical baseline's. This is consistent with our claims that our method does not make assumptions about data distributions and prediction models, as long as the data from the calibration set and test set are exchangeable.

6.4 Efficiency-Accuracy Tradeoffs with Certified Error Control

In this section, we investigate the guarantee of the overall tradeoffs between efficiency and accuracy. Fig. 5 illustrates the efficiency-accuracy tradeoff results on MS MARCO Passage v1 using DPR + MonoELECTRA. Similarly, we set the confidence level $1 - \delta$ to be 0.9. Our method (blue line) achieves the best MRR@10 score at around 20% of the original top-1000 candidate set size, which is a very good tradeoff between accuracy and efficiency. In addition, the MRR@10 score of our method (blue line) is higher than the specified score threshold (orange line) with at least 90% coverage, which further verifies the guarantee claims we made about our method. We can see that our method achieves a good tradeoff while satisfying different values of the risk level α.

6.5 Confidence-Risk Correction

In Section 4.3, we mentioned that it might be impossible to achieve the risk threshold if it is specified too low in ranking tasks. Our solution approaches this problem by correcting either the risk threshold α or the significance level δ as shown in Fig. 3c and 3d. In practice, the risk threshold correction is rather straightforward: if the risk is lower than the minimal upper bound over all λ, we just reset the risk threshold to the minimal upper bound as shown in Eq (9). For the confidence correction, we need to fix the risk threshold and shrink the upper bound by increasing the significance δ until the confidence $(1 - \delta)$ is 0 as shown in Eq (10). Fig. 6 shows the confidence correction of our method. We use the DPR+MonoELECTRA ranking system whose best

reranking MRR@10 score is 0.41 on the dev small set, meaning that the minimal risk is around 0.59, and the minimal UCB of the risk function is around 0.61 (the vertical line). From right to left, we can see that the confidence does not change too much until the risk passes 0.6137. As the risk threshold decreases, the confidence (the orange line) also gradually decreases, which is consistent with the empirical coverage (the green line).

7 Conclusion

We present a theoretically principled method for candidate set pruning of two-stage ranking systems, allowing users to control a customized loss under the desired threshold with high probability. We further propose to correct the risk threshold or confidence level if the desired risk cannot be achieved given the ranking system. Experiments performed under in-domain (MS MARCO Passage v1) and out-of-domain (Quora) settings show that our method provides a consistent performance guarantee to candidate set pruning across multiple ranking systems.

8 Limitations

This work has two limitations. The first one is that our method assumes the calibration data and test data are exchangeable and sampled from the same data distribution, which limits its utility in out-of-domain evaluation. However, the training data does not need to have the same distribution as the test data, as we have shown in Section 6.3. The second limitation is that our method needs a calibration set that is big enough (usually 1000∼10000 data points) in order to provide tight upper confidence bounds, which otherwise will become very conservative in pruning and increase reranking latency. This drawback limits our method's utility in a low-resource regime where calibration data are scarce.

In this paper, we mainly care about in-domain calibration (i.e., exchangeable calibration-test data) for ranking, which was not addressed properly before. A promising future direction for out-of-domain ranking calibration is unsupervised domain adaptation, where the labelled data are scarce but the unlabeled data are abundant (Shimodaira, 2000; Park et al., 2020b). It has also been proved that a classifier's target error in terms of its source error and the divergence between the two domains could be bounded (Ben-David et al., 2010).

Acknowledgements

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada; computational resources were provided by Compute Canada.

References

Stephen Bates, A. Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. 2021. Distribution-free, risk-controlling prediction sets. *J. ACM*, 68:43:1–43:34.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1):151–175.

Francesco Busolin, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2021. Learning early exit strategies for additive ranking ensembles. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Berkant Barla Cambazoglu, Hugo Zaragoza, Olivier Chapelle, Jiang Chen, Ciya Liao, Zhaohui Zheng, and Jon Degenhardt. 2010. Early exit optimizations for additive machine learned ranking systems. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada.

Zhuyun Dai and Jamie Callan. 2020. Context-aware term weighting for first stage passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1533–1536.

Adam Fisch, Tal Schuster, T. Jaakkola, and Regina Barzilay. 2021. Efficient conformal prediction via cascaded inference with expanded admission. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 3030–3042, Online.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei A. Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jimmy Lin. 2021. A proposed conceptual framework for a representational approach to information retrieval. *ArXiv*, abs/2110.01529.

Jimmy Lin and Xueguang Ma. 2021. A few brief notes on DeepImpact, COIL, and a conceptual framework for information retrieval techniques. *ArXiv*, abs/2106.14807.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, R. Perego, and Salvatore Trani. 2020. Query-level early exit for additive learning-to-rank ensembles. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Xueguang Ma, Kai Sun, Ronak Pradeep, Minghan Li, and Jimmy Lin. 2022. Another look at DPR: Reproduction of training and replication of retrieval. In *Proceedings of the 44th European Conference on Information Retrieval (ECIR 2022), Part I*, pages 613–626, Stavanger, Norway.

Antonio Mallia, Omar Khattab, Nicola Tonellotto, and Torsten Suel. 2021. Learning passage impacts for inverted indexes. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *ArXiv*, abs/1910.14424.

Alexandros Ntoulas and Junghoo Cho. 2007. Pruning policies for two-tiered inverted index with correctness guarantee. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 191–198.

Sangdon Park, Osbert Bastani, Nikolai Matni, and Insup Lee. 2020a. PAC confidence sets for deep neural networks via calibrated prediction. *ArXiv*, abs/2001.00106.

Sangdon Park, Osbert Bastani, James Weimer, and Insup Lee. 2020b. Calibrated prediction with covariate shift via unsupervised domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pages 3219–3229. PMLR.

Sangdon Park, Shuo Li, Osbert Bastani, and Insup Lee. 2021. PAC confidence predictions for deep neural network classifiers. *ArXiv*, abs/2011.00716.

Ronak Pradeep, Yuqi Liu, Xinyu Zhang, Yilin Li, Andrew Yates, and Jimmy Lin. 2022. Squeezing water from a stone: A bag of tricks for further improving cross-encoder effectiveness for reranking. In *European Conference on Information Retrieval*, pages 655–670. Springer.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. *ArXiv*, abs/2109.08535.

Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Luca Soldaini and Alessandro Moschitti. 2020. The cascade transformer: an application for efficient answer sentence selection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5697–5708, Online.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663.

Ellen M. Voorhees and Angela Ellis, editors. 2019. *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. 2005. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg.

Vladimir Vovk, Alexander Gammerman, and Craig Saunders. 1999. Machine-learning applications of algorithmic randomness. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999)*.

Abraham Wald. 1943. An extension of Wilks' method for setting tolerance limits. *Annals of Mathematical Statistics*, 14:45–55.

Lidan Wang, Jimmy Lin, and Donald Metzler. 2011. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th International ACM SIGIR Conference on Research and development in Information Retrieval*, pages 105–114.

Ian Waudby-Smith and Aaditya Ramdas. 2020. Variance-adaptive confidence sequences by betting. *arXiv*, abs/2010.09686.

David J. Weiss and Ben Taskar. 2010. Structured prediction cascades. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*.

Samuel Stanley Wilks. 1941. Determination of sample sizes for setting tolerance limits. *Annals of Mathematical Statistics*, 12:91–96.

Samuel Stanley Wilks. 1942. Statistical prediction with special reference to the problem of tolerance limits. *Annals of Mathematical Statistics*, 13:400–409.

Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy Lin. 2020a. Early exiting BERT for efficient document ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 83–88, Online.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020b. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online. Association for Computational Linguistics.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. BERxiT: Early exiting for BERT with better fine-tuning and extension to regression. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 91–104, Online.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.

Andrew Yates, Siddhant Arora, Xinyu Zhang, Wei Yang, Kevin Martin Jose, and Jimmy Lin. 2020. Capreolus: A toolkit for end-to-end neural ad hoc retrieval. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM 2020)*, pages 861–864, Houston, Texas.

Emine Yilmaz and Stephen Robertson. 2009. Deep versus shallow judgments in learning to rank. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 662–663.

A Appendix

A.1 Proofs

Proof of Theorem 1. We first prove that if there exists $\alpha > 0$ and $\delta > 0$ such that for every $\hat{\lambda} \in \Lambda$, $\widehat{R}^+_\delta(\mathcal{T}_{\hat{\lambda}}; \zeta) > \alpha$, then $\mathcal{T}_{\hat{\lambda}}$ will no longer be a pruning function certified error control. Suppose there exists an $\hat{\lambda}$ such that $\Pr(R(\mathcal{T}_{\hat{\lambda}}; \zeta) \leq \alpha) \geq 1 - \delta$, by the coverage property in Eq (5) we know that $\widehat{R}^+_\delta(\mathcal{T}_{\hat{\lambda}}; \zeta) \leq \alpha$. Contradiction.

Next, we prove that for (α_c, δ) in Eq (9), $\mathcal{T}_{\hat{\lambda}}$ has certified error control. By definition in Eq (9), we know that there exists an $\hat{\lambda}$ such that $\widehat{R}^+_\delta(\mathcal{T}_{\hat{\lambda}}; \zeta) = \alpha_c$, and by the coverage property in Eq (5), we have $\Pr(R(\mathcal{T}_{\hat{\lambda}}; \zeta) \leq \alpha_c) \geq 1 - \delta$. Done.

Finally, we prove that for (α_c, δ) in Eq (10), $\mathcal{T}_{\hat{\lambda}}$ has certified error control. By definition in Eq (10), we know that there exists an $\hat{\lambda}$ such that $\widehat{R}^+_{\delta_c}(\mathcal{T}_{\hat{\lambda}}; \zeta) = \alpha$, and by the coverage property in Eq (5), we have $\Pr(R(\mathcal{T}_{\hat{\lambda}}) \leq \alpha; \zeta) \geq 1 - \delta_c$. □

Theorem 2 (Validity of Certified Error Control)**.** *(Bates et al., 2021) Let $\{P'_i, D'_{\omega i}\}_{i=1}^m$ be an i.i.d. sample and $L(P', D'_\omega; \zeta)$ is monotone w.r.t. λ as in Eq (4). Let $\{\mathcal{T}_\lambda\}_{\lambda \in \Lambda}$ be a collection of pruning function satisfying the nesting property in Eq (3). Suppose Eq (5) holds pointwise for each λ, and that $R(\mathcal{T}_\lambda; \zeta)$ is continuous. Then for $\hat{\lambda}$ chosen as in Eq (6), we have*

$$\Pr(R(\mathcal{T}_{\hat{\lambda}}; \zeta) \leq \alpha) \geq 1 - \delta.$$

That is, $\mathcal{T}_{\hat{\lambda}}$ is a pruning function with certified error control.

Proof of Theorem 2. Our proof follows the framework in (Bates et al., 2021). Consider the smallest λ that controls the risk:

$$\lambda^* \doteq \inf \{\lambda \in \Lambda : R(\mathcal{T}_\lambda; \zeta) \leq \alpha\}.$$

Suppose $R(\mathcal{T}_{\hat{\lambda}}; \zeta) > \alpha$. By the definition of λ^* and the monotonicity and continuity of $R(\cdot; \zeta)$, this implies $\lambda^* < \hat{\lambda}$. By the definition of $\hat{\lambda}$, this further implies that $\widehat{R}^+_\delta(\mathcal{T}_{\lambda^*}) < \alpha$. But since $R(\mathcal{T}_{\lambda^*}; \zeta) = \alpha$ (by continuity) and by the coverage property in Eq (5), this happens with probability at most δ. □

A.2 Waudby-Smith–Ramdas Bound

Bates et al. (2021) provide a one-sided variant of the Waudby-Smith–Ramdas (WSR) bound (Waudby-Smith and Ramdas, 2020; Bates et al., 2021):

Proposition 1 (Waudby-Smith–Ramdas bound)**.** *Let $L_i(\lambda) = L(D'_\omega, T_\lambda(D'); \zeta)$, and*

$$\hat{\mu}_i(\lambda) = \frac{\frac{1}{2} + \sum_{j=1}^i L_j(\lambda)}{1+i},$$
$$\hat{\sigma}_i^2(\lambda) = \frac{\frac{1}{4} + \sum_{j=1}^i (L_j(\lambda) - \hat{\mu}_i(\lambda))^2}{1+i},$$
$$\nu_i(\lambda) = \min\left\{1, \sqrt{\frac{2\log(1/\delta)}{n\hat{\sigma}_i^2(\lambda)}}\right\}.$$

Further let

$$\mathcal{K}_i(R; \lambda) = \prod_{j=1}^i \{1 - \nu_j(\lambda)(L_j(\lambda) - R)\},$$

and

$$\widehat{R}^+_\delta(\mathcal{T}_\lambda) = \inf\left\{R \geq 0 : \max_i \mathcal{K}_i(R; \lambda) > \frac{1}{\delta}\right\}.$$

Then $\widehat{R}^+_\delta(\mathcal{T}_\lambda)$ is a $(1-\delta)$ upper confidence bound for $R(\lambda)$.

The proofs are basically a restatement of the Theorem 4 in Waudby-Smith and Ramdas (2020) and Proposition 5 in Bates et al. (2021).

A.3 Algorithmic Implementation

Alg. 1 provides a detailed implementation of the certified error control method for relevance ranking using the form of pseudo-code, where we use MRR@10 as the metric. The algorithm takes the ranking system and calibration set as the inputs and returns the set predictor, the corrected risk, and corrected confidence.

Algorithm 1: Calibration procedure.

Parameter: Risk Level α, Confidence $1-\delta$
Model: Retriever $\{\eta_Q, \eta_D\}$, Reranker ζ
Data: Calibration Set $\{q_i, d_i', d_{\omega i}'\}_{i=1}^{m}$
Metric: MRR@10
Result: $\hat{\lambda}$, α_c, $1-\delta_c$

/*Retrieval Prediction*/
$S_v' \leftarrow \emptyset, D_v' \leftarrow \emptyset$
for $i \leftarrow 1$ **to** m **do**
    $u_i \leftarrow \eta_Q(q_i)$
    $S_v \leftarrow \emptyset, D_v \leftarrow \emptyset$
    for $j \leftarrow 1$ **to** k **do**
        $v_{ij} \leftarrow \eta_D(d_{ij}'), s_v = u_i^T v_{ij}$
        $S_v \leftarrow S_v \cup \{s_v\}, D_v \leftarrow D_v \cup \{d_{ij}'\}$
    Sort D_v and S_v in desc. order of S_v
    $S_v' \leftarrow S_v' \cup S_v, D_v' \leftarrow D_v' \cup D_v$
$S_v' \leftarrow$ Platt-Scaling(S_v')
/*Reranking and Compute Upper Bound*/
$\widehat{R}^+ \leftarrow \emptyset, L' \leftarrow \emptyset, \Lambda \leftarrow \emptyset$
for $\lambda \leftarrow 1$ **to** 0 **by** -10^{-5} **do**
    $P' \leftarrow D'_{v(S_v' \geq \lambda)}, L \leftarrow \emptyset$
    for $i \leftarrow 1$ **to** m **do**
        $S_r \leftarrow \emptyset, P_r \leftarrow \emptyset$
        for p **in** P_i' **do**
            $s_r = \beta \cdot \phi(\eta_q(q_i), \eta_d(p)) + (1-\beta) \cdot \zeta(\text{concat}(q, p))$
            $S_r \leftarrow S_r \cup \{s_r\}, P_r \leftarrow P_r \cup \{p\}$
            Sort P_r in desc. order of S_r
        $L \leftarrow L \cup \{1 - \text{MRR@10}(P_r, d_{\omega i}')\}$
    $\widehat{R}^+ \leftarrow \widehat{R}^+ \cup \{\text{WSR}(L, \delta)\}$
    $\Lambda \leftarrow \Lambda \cup \{\lambda\}, L' \leftarrow L' \cup L$
/*Compute Lambda*/
if $\min(\widehat{R}^+) \leq \alpha$ **then**
    for $\hat{\lambda}, R$ **in** $\Lambda, \widehat{R}^+$ **do**
        if $R \geq \alpha$ **then**
            break
    return $\hat{\lambda}, \alpha, 1-\delta$
else
    /*Risk-Confidence Correction*/
    $\alpha_c = \min(\widehat{R}^+)$
    for $\delta_c \leftarrow \delta$ **to** 0 **by** -10^{-2} **do**
        for $\hat{\lambda}, L$ **in** Λ, L' **do**
            if $\text{WSR}(L, \delta_c) \leq \alpha$ **then**
                break
    return $\hat{\lambda}, \alpha_c, 1-\delta_c$