An Assumption-Free Approach to the Dynamic Truncation of Ranked Lists

Yen-Chieh Lien
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
ylien@cs.umass.edu

Daniel Cohen
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
dcohen@cs.umass.edu

W. Bruce Croft
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
croft@cs.umass.edu

ABSTRACT

In traditional retrieval environments, a ranked list of candidate documents is produced without regard to the number of documents. With the rise in interactive IR as well as professional searches such as legal retrieval, this results in a substantial ranked list which is scanned by a user until their information need is satisfied. Determining the point at which the ranking model has low confidence in the relevance score is a challenging, but potentially very useful, task. Truncation of the ranked list must balance the needs of the user with the confidence of the retrieval model. Unlike query performance prediction where the task is to estimate the performance of a model based on an initial query and a given set documents, dynamic truncation minimizes the risk of viewing a non-relevant document given an external metric by estimating the confidence of the retrieval model using a distribution over its already calculated output scores, and subsequently truncating the ranking at that position. In this paper, we propose an assumption-free approach to learning a non-parametric score distribution over any retrieval model and demonstrate the efficacy of our method on Robust04, significantly improving user defined metrics compared to previous approaches.

CCS CONCEPTS

• **Information systems → Retrieval effectiveness**.

KEYWORDS

Ranked list truncation, assumption-free method, custom loss function

ACM Reference Format:
Yen-Chieh Lien, Daniel Cohen, and W. Bruce Croft. 2019. An Assumption-Free Approach to the Dynamic Truncation of Ranked Lists. In *ICTIR '19, October 02–05, 2019, Santa Clara, CA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

ICTIR '19, October 02–05, 2019, Santa Clara, CA

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

A typical output of an information retrieval (IR) system is a ranked list of documents with their corresponding scores. However, certain applications of IR are sensitive to the amount of candidate documents retrieved. In the case of legal retrieval, for example, each additional document results in a monetary penalty due to the additional time spent investigating the document, while mobile and dialogue systems incur a negative user impact when excess non-relevant documents are returned.

In situations where there is a penalty on the number of documents returned, based on a user-defined constraint, it is crucial for an IR system to minimize the risk of showing non-relevant documents. As opposed to query performance prediction, here the task is to capture the confidence of the model and approximate performance over a fixed set size of scores; the truncation problem attempts to find the maximum partition of the original ranked list with respect to a user defined penalty for non-relevant documents.

As the given ranked list is already ordered, this problem decomposes into identifying an appropriate cutoff k for each query given the documents' relevance scores. We formulate this problem as a sequential decision process over a pre-computed ranked list and propose a novel approach that takes into account neighboring documents. Given a fixed scoring function, we implement an LSTM-based model, referred to as BiCut, over the ordered list and partition it to minimize a predefined cost function tailored to a specific user need.

In attempt to solve this, past work [2] has assumed a parametric distribution over relevant documents in a ranked list and selected models that fit that parametric distribution. In this paper, we introduce a neural approach that is able to learn any parametric or non parametric distribution over the score list [8], thus removing the model selection problem required previously. We validate the model on the TREC Robust collection, and our experiments demonstrate that BiCut produces a significantly better partition than previous approaches over a variety of user scenarios.

2 RELATED WORK

While the need for dynamic truncation has only recently become a significant issue [1], past work has addressed the concept of identifying the most relevant documents for a user. Arampatzis et al. [2] approach this task for the TREC 2008 Legal Track. The authors leverage the score distribution to estimate a threshold score for each query given a ranked list of 100000 documents. By modeling the score distribution of query-document pairs via a normal-exponential mixture, they estimate the truncation position using the Expectation Maximization algorithm [10]. The authors achieve

significant performance improvements in F1; however, this relies on their distribution assumption, which the authors themselves state does not always hold. In contrast, the proposed BiCut model learns a latent distribution within its parameters during training, and thus does not rely on having a prior assumption on the ranked list. Furthermore, we approach the dynamic truncation problem from an *ad-hoc* retrieval perspective and focus on shorter ranked lists.

A key motivator of this work is query performance prediction (QPP) [3–5, 15] where the aim is to estimate the effectiveness of a retrieval model given a query. As the drive behind QPP is that a retrieval model's performance is dependent on the query, it is reasonable to approach this from a post query perspective, and dynamically reduce the final ranked list. In particular, the work done by Culpepper et al. [4] approximates this paper. The key difference is the authors use only the score information over a set of sampled documents to estimate the performance of the query on a fixed set using a feed forward model. The proposed method in this paper selects a maximum partition via a seq2seq approach, and uses document information to learn the stability of the retrieval model over a ranked list.

In addition, the truncation decision is intuitively related to cascade methods. Cascade approaches [14] build a multi-stage ranker and use a hyperparameter to decide the number of documents to pass on to the following stage based off of efficiency and retrieval performance. Conversely, the work proposed in this paper focuses on the problem of learning a distribution over this truncation hyperparameter given any utility function.

3 DYNAMIC TRUNCATION

The dynamic truncation problem can be represented as maximizing an external metric at a given truncation position.

Formally, for a ranked document list $D = \{d_1, d_2, \cdots, d_n\}$ for query q, and retrieval function $f(q, d)$, where $f(q, d_i) > f(q, d_j)$ if $i < j$, the cutoff model needs to predict $p_i = Pr(d_i \in D_t | d_0 ... d_{i-1} \in D_t, f, q, D)$ for all $d_i \in D$ where D_t represents the truncated ranked list. Then, given a evaluation metric C, such as F1 and NDCG@k, this problem can be defined as predicting a k_{opt} s.t.

$$\min_{0<k_{opt}<n} C(D_{1,\ldots,k_{opt}}) \text{ s.t. } C(D_{1,\ldots,k_{opt}}) \geq C(D_{1,\ldots,l}) \; \forall \; l > k_{opt}$$

3.1 Neural Approach

While past work [15] has used a feedforward neural model to predict the performance of a query given a fixed input, the task of truncating a ranked list requires a model to consider all possible cuts of an ordered set. The fixed position nature of the traditional feedforward approach makes it difficult to determine where the IR model's score fails to capture relevance within the ranked list.

As it is these relative score fluctuations that determine where to truncate, a BiLSTM network [6] is adopted to model these sequential relations. For a document d_i, consider a relevance score $f(q, d_i)$ and corresponding document statistics $s(d_i)$, including document length, number of unique tokens, and term frequency. By feeding the scores of the ranked list with concatenated document statistics into the BiLSTM, the sequence to sequence nature of RNNs results in a confidence value of $\{f(q, d_i), s(d_i)\}$ given its context.

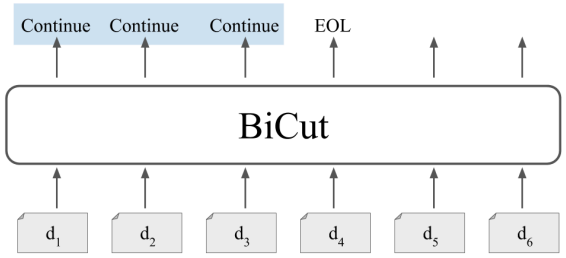

Figure 1: Toy example of BiCut over a ranked list with 6 documents. EOL indicates the truncation point of the ranked list.

For example, where $\{f(q, d_i), s(d_i)\}$ has a significant change from $\{f(q, d_{i-1}), s(d_{i-1})\}$, the model can identify when a significant shift occurs and incorporate this into the truncation prediction. This approach allows for a non-parametric view of the score and document probability distribution it is modeling, and thus is an assumption-free approach.

We train the BiCut via Adam [9] with a learning rate of 10^{-4}, where a ranked list is input sequentially in both forward and reverse order into two 128 dimension Bi-LSTM layers. Each position is then passed into a 256 dimension feedforward layer prior to a two dimension softmax. In a similar vein to machine translation with the end of sentence token, the BiCut output consists of a two dimension softmax, representing `Continue` and `EOL` for end-of-list. The ranked list is truncated at the first instance of `EOL`.

To clarify the process of BiCut, we illustrate a toy example in Figure 1. In this example, the input is a ranked list of the top 6 documents retrieved by a model. BiCut sequentially decides `Continue` for the first 3 documents, and then judges d_4 as not relevant enough to include given some predefined metric discussed in the following section. The final retrieved result will be $\{d_1, d_2, d_3\}$.

3.2 Cutoff Loss Function

The flexibility of treating the scores and document statistics as a non-parametric distribution allows any metric C if there is an appropriate corresponding loss function $\mathcal{L}_C$ for minimization objective to be chosen over any ranked list. To illustrate the flexibility of this approach, two C functions are proposed and evaluated. However, any standard IR metric can be used, as well as context sensitive user models such as learning an inverse reward function for dialogue systems.

Recall-Precision: Precision and Recall are two fundamental metrics in IR field. Due to the trade-off between them, the F1 metric is commonly adopted to balance these metrics when evaluating a system. To reflect BiCut's ability to handle different metrics, we use F1 as the user defined metric C, referred to as C_{F1}, and adopt a weighted loss function that leverages a joint loss function with an α hyperparameter that controls the impact of false positives and false negatives. We apply the loss,

$$\mathcal{L}_{C_{F1}} = \sum_{d_i \in D_j} (\alpha \mathbb{I}(y_i = 0) \frac{p_i}{1-r} + (1-\alpha) \mathbb{I}(y_i = 1) \frac{1-p_i}{r}) \quad (1)$$

where y_i is the relevance label, $\mathbb{I}$ is an indicator function, r is the estimation of proportion of relevant documents in the list, D_j is a partition of D up to document d_j, and α is a hyperparameter between 0 and 1 for modifying the balance between two terms. In the loss function, the first term is for false positive instances and the second is for false negative instances. r acts as a normalization value for the ratio of relevant and non-relevant documents. In this paper, we estimate r by the proportion of relevant documents in the training data.

Negative Cumulative Impact: To represent a case where C is a non-linear user cost such as in mobile search or dialogue systems, we consider a negative cumulative impact (NCI) user model where there exists a significant increase in cost resulting from including non-relevant documents in the truncated list. C_{NCI} is modeled as a discounted utility gain when a relevant document is included, and a linearly increasing negative utility for including a non-relevant document. C_{NCI}, and the loss of the cost function used to train BiCut is shown below:

$$C_{NCI} = \sum_{d_i \in D_j} \left(\frac{\mathbb{I}(y_i = 1)}{\log_2(i+1)} - \mathbb{I}(y_i = 0) * \frac{i}{\alpha}\right) \tag{2}$$

$$\mathcal{L}_{C_{NCI}} = \sum_{d_i \in D_j} \left(-\frac{\mathbb{I}(y_i = 1)p_i}{\log_2(i+1)} + \mathbb{I}(y_i = 0)p_i * \frac{i}{\alpha}\right) \tag{3}$$

4 EXPERIMENTS

4.1 Dataset

We analyze truncation methods on the TREC collection *Robust04*. This collection was used in the TREC Robust Track 2004. It contains 250 queries, 528k news articles, and approximately 70 relevant documents for each query. In this dataset, each query has a title, rough description and detailed narrative. In this paper, the title is used as the query for retrieval.

We randomly select 80% of the queries as training set, and the remaining 20% as testing set. For hyperparameter tuning, we conduct a grid search using a 5-fold cross-validation over the training set.

4.2 Baselines

A number of baseline truncation methods are evaluated. The first policy uses a fixed k. This is a common method for evaluation of ranking functions [7, 11, 13] as past work has evaluated their models by metrics with a fixed truncation, such as Precision@10 and nDCG@10. Second, the greedy choice of k based on training data is used as a case of a fixed k method by choosing a k such that C is maximized over the training data. Lastly, Arampatzis et al. [2] is evaluated as a competitive method for dynamic list truncation that relies on a parametric view of the data.

4.3 Setting

The effectiveness of a dynamic truncation model is analyzed over two retrieval approaches. The first is a traditional *tf.idf* based model, BM25, and the second represents a deep neural retrieval model, DRMM [7]. We retrieve the top 300 documents from each retrieval model and pass it through the truncation methods. Although this process removes a significant number of documents with respect to the entire collection, it is much larger than the general cut-off adopted by ranking evaluations such as P@10 or R@20. Thus, while it does simplify the dynamic truncation task, it provides a relatively unbiased and challenging environment to sufficiently validate BiCut.

Table 1: Truncation results over F1 and NCI costs. † marks a statistically significant difference against the best baseline ($p < 0.05$ for paired t-test) for each C.

	BM25		DRMM	
	C_{F1}	C_{NCI}	C_{F1}	C_{NCI}
Fixed k (5)	0.1621	1.403	0.0974	1.518
Fixed k (10)	0.1995	2.027	0.1560	2.141
Fixed k (50)	0.2241	2.915	0.2646	3.436
Greedy choice	0.2292	3.038	0.2601	3.038
Arampatzis et al. [2]	0.2227	-	0.1660	-
BiCut(f)	0.2349†	3.099†	0.2740†	3.487†
BiCut (f,s)	**0.2412**†	**3.108**†	**0.2836**†	**3.511**†

We conduct two experiments to validate the BiCut model. First, we examine the performance of BiCut by dynamically choosing a cutoff on the 300 length ranked list for the desired metric, C. Second, we perform an ablation study by examining the role of the document statistic distribution on BiCut's ability to estimate the confidence of the retrieval model by only using $f(q, d)$ as input. The ablation study also facilitates a fair comparison between Arampatzis et al [2] and BiCut by reducing the input space to that of just the retrieval score, $f(q, d)$, as the assumption-based method does not exploit individual document statistics, $s(d)$.

5 RESULTS AND DISCUSSION

5.1 F1 Cutoff Performance

As seen in Table 1, BiCut achieves a significant increase in performance over all baselines in the target F1 metric. We include precision and recall performance to provide insight into the F1 value as seen in Figure 2, where the model achieves optimal F1 performance at $\alpha = 0.65$. This suggests that unlike the actual F1 metric, which precision and recall are equally important, the loss is more effective when it focuses on precision over recall. We consider that the estimation of r from training data inevitably causes bias, but the mechanism of a tunable α can still help to find an effective weighting for C_{F1}. Furthermore, we also see that different values of α result in a precision-oriented (larger α) or recall-oriented (smaller α) predictor. Lastly, in Figure 3, BiCut aims to approximate the optimal distribution of cutoff value, and truncates the list well before the 300 document limit, indicating that the initial cutoff did not impact the model's results. However, the inability to properly produce the optimal k when k is greater than 200 suggests that the model learns a conservative approach to the truncation task.

Of particular note is the performance of Arampatzis et al. [2]. As their model relies on certain distribution assumptions over the collection and scores to fit a parametric normal-exponential distribution, it struggles in the situations where this assumption is violated. While the normal-exponential approach was developed for *tf.idf* score distributions, the significant decrease in performance

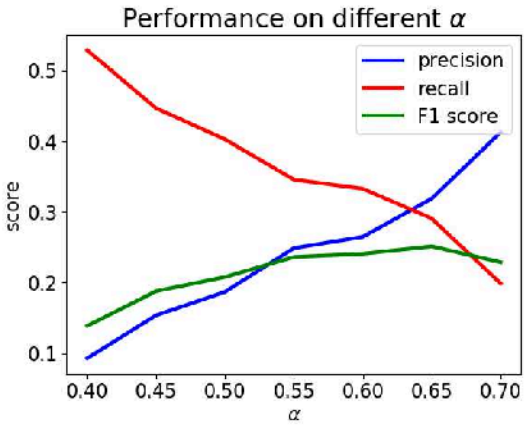

Figure 2: The curve of precision, recall and F1 score over adjusting α for the Recall-Precision loss.

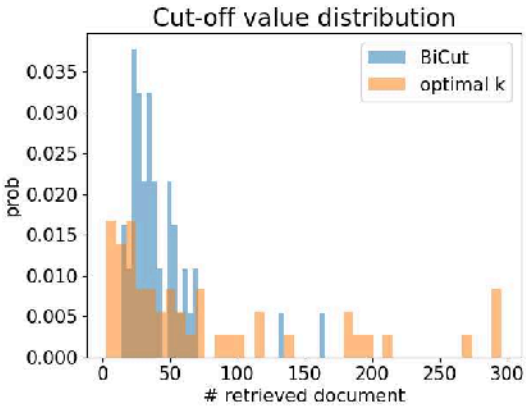

Figure 3: The distribution of truncation values of testing queries for Recall-Precision loss. Optimal k represents the best cut-off searched by brute-force

compared to a greedy-k approach over the DRMM ranked list suggests that neural model ranked lists follow a different distribution. This is further enforced by BiCut's improvement due to BiCut's assumption-free characteristics of the distribution.

Examining the impact of using document information, BiCut(f, s) reveals that the majority of information with respect to the confidence of the retrieval model is captured directly in the score distribution. The caveat is that in the case of a neural model where document statistics is not inherently part of the score as it is for BM25, there exists a greater change in performance between BiCut(f) and BiCut(f, s) with an increase in 3.5% compared to 2.6% of the BM25 experiment.

5.2 Negative Cumulative Impact Performance

Following a similar trend to the F1 performance, the BiCut approach outperforms the traditional baselines. Arampatzis et al. [2] is not included as their model is not adaptable for other cost functions. The small difference between BiCut(f) and BiCut(f, s) may be attributed to the non-linear decreasing reward, where only in certain situations does the change in document distribution warrant a truncation on a ranked list over that of original IR model's score. This same non-linear behaviour also reflects the close performance between the best greedy choice, BiCut(f) and BiCut(f, s) as the average k values were respectively 45, 47.98, and 48.90. Although it is close to the performance of the greedy method, BiCut is able to consistently find a k that further improves performance with respect to C.

6 CONCLUSION AND FUTURE WORK

We demonstrate the efficacy of an RNN-based model combined with a flexible cost function to dynamically truncate lists on a per query basis. This approach is robust to the initial retrieval model's score distribution due to its assumption-free property. Furthermore, in the case of neural retrieval where a model can produce a relevance score for each document in a collection, the BiCut model can significantly reduce the workload of human judgment, especially for recall-oriented retrieval tasks.

While not addressed within the scope of this paper, this approach can be incorporated into a cascade framework where the cutoff value k for each ranker is chosen by a BiCut inspired model rather than tuned using a parametric approach or through brute force methods. The C function then would be the retrieval cost of the next ranker.

As there exists ranked lists with a large number of documents, such as the collection used in Arampatzis et al. [2], potential future work would examine the efficacy of adapting a neural truncation approach to larger sequences where BiCut would fail due to the nature of LSTMs struggling to capture the information contained over long sequences [12].

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) via AFRL contact #FA8650-17-C-9116 under subcontract #94671240 from the University of Southern California. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] ACM 2018. *SWIRL 2018*. ACM.
[2] Avi Arampatzis, Jaap Kamps, and Stephen Robertson. 2009. Where to stop reading a ranked list?: threshold optimization using truncated score distributions. In *SIGIR '09, Boston, MA, USA, July 19-23, 2009*. 524–531.
[3] Stephen Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *SIGIR '02, August 11-15, 2002, Tampere, Finland*. 299–306.
[4] J. Shane Culpepper, Charles L. A. Clarke, and Jimmy Lin. 2016. Dynamic Cutoff Prediction in Multi-Stage Retrieval Systems. In *Proceedings of the 21st Australasian*

Document Computing Symposium (ADCS '16). ACM, New York, NY, USA, 17–24.

[5] Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). 2014. *SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*. ACM.

[6] Alex Graves. 2013. Generating Sequences With Recurrent Neural Networks. *CoRR* abs/1308.0850 (2013).

[7] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM '16*. 55–64.

[8] Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. In *Neural Networks*. 251–257.

[9] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

[10] R. Manmatha, T. Rath, and F. Feng. 2001. Modeling Score Distributions for Combining the Outputs of Search Engines. In *SIGIR '01*. ACM, New York, NY, USA, 267–275.

[11] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Cross Temporal Recurrent Networks for Ranking Question Answer Pairs. In *AAAI '18, New Orleans, Louisiana, USA, February 2-7, 2018*.

[12] Trieu H. Trinh, Andrew M. Dai, Thang Luong, and Quoc V. Le. 2018. Learning Longer-term Dependencies in RNNs with Auxiliary Losses. In *ICML '18, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 4972–4981.

[13] Di Wang and Eric Nyberg. 2015. A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering. In *ACL '15, July 26-31, 2015, Beijing, China*. 707–712.

[14] Lidan Wang, Jimmy Lin, and Donald Metzler. 2011. A Cascade Ranking Model for Efficient Ranked Retrieval. In *SIGIR '11*. 105–114.

[15] Hamed Zamani, W. Bruce Croft, and J. Culpepper. 2018. Neural Query Performance Prediction using Weak Supervision from Multiple Signals. In *SIGIR '18*.