# TU Wien at TREC DL and Podcast 2021: Simple Compression for Dense Retrieval

Sebastian Hofstätter
TU Wien
s.hofstaetter@tuwien.ac.at

Mete Sertkan
TU Wien
mete.sertkan@tuwien.ac.at

Allan Hanbury
TU Wien
hanbury@ifs.tuwien.ac.at

## ABSTRACT

The IR group of TU Wien participated in two tracks at TREC 2021: Deep Learning and Podcast segment retrieval. We continued our focus from our previous TREC participations on efficient approaches for retrieval and re-ranking. We propose a simple training process for compressing a dense retrieval model's output. First, we train it with full capacity, and then add a compression, or dimensionality reduction, layer on top and conduct a second full training pass. At TREC 2021 we test this model in a blind evaluation and zero-shot collection transfer for both Deep Learning and Podcast tracks.

For our participation at the Podcast segment retrieval track, we also employ hybrid sparse-dense retrieval. Furthermore, we utilize auxiliary information to re-rank the retrieved segments by entertainment and subjectivity signals.

Our results show that our simple compression procedure with approximate nearest neighbor search achieves comparable in-domain results (minus 2 points nDCG@10 difference) to a full TAS-Balanced retriever and reasonable effectiveness in a zero-shot domain transfer (Podcast track), where we outperform BM25 by 6 points nDCG@10.

## 1 INTRODUCTION

The IR group of TU Wien participated in two tracks at TREC 2021: Deep Learning (DL) and Podcast segment retrieval. We continued our focus from our previous TREC participations [6, 8] on efficient approaches for retrieval and re-ranking. At the DL track, we tested our TAS-Balanced [7] training approach against a standalone dense retrieval baseline, and a compressed version of TAS-Balanced dense retriever, trained with a simple dimensionality reduction technique, which we present in this paper.

We propose a simple training process for compressing a dense retrieval model's output, usable with any training approach as it does not alter the input interface or loss function. Our compression pipeline is summarized as follows:

(1) Train the $\text{BERT}_{\text{DOT}}$ model with full capacity (for DistilBERT [21] this is 768 dimensions) with a training method of your choice – we use our TAS-Balanced approach;
(2) Add a randomly initialized compression, or dimensionality reduction, layer after the CLS pooling to the fully trained model (we settled on 192 dimensions, a 4x reduction);
(3) Conduct a second full training pass, of the training method of your choice, without freezing any weights or training length constraints.

With this approach, we reduced the storage cost by 4x and only loose 1% of effectiveness on MSMARCO-V1 compared to our best TAS-Balanced model. While one could use a post-hoc compression approach, we chose to incorporate the compression directly into the model, as it allows us 1) to publish the model with compressed output as one unit on the HuggingFace model hub and 2) anyone using this checkpoint automatically receives smaller but equal qualitative vectors without adding more complexity to their system.

At TREC 2021, we test this model in a blind evaluation and zero-shot collection transfer for Deep Learning and Podcast tracks. The DL track focuses on the feasibility of using DR models on a much larger scale, with a slight test collection shift and strong size increase compared to the training data we used (we trained on MSMARCO v1). The Podcast track represents a zero-shot transfer scenario, without any domain-specific training data – the queries are much shorter, and the passages are much longer than in the MSMARCO collection.

Additionally, for the ad-hoc retrieval task of the podcast track, we apply: 1) Our full TAS-B trained $\text{BERT}_{\text{DOT}}$ model and re-rank the outcome with our knowledge distilled $\text{BERT}_{\text{CAT}}$ [5]; 2) Sparse-dense retrieval [12] using our full TAS-B trained model and a standard BM25 approach [18]; and 3) Our full TAS-B trained model, merge the outcome with BM25 rankings (omitting duplicates), and re-rank the top-1000 with our knowledge distilled $\text{BERT}_{\text{CAT}}$.

For the *Re-Rank Entertaining* sub-task of the podcast segment retrieval task, we use the output of a BERT-based emotions classifier, fine-tuned on GoEmotions dataset [3], as an additional signal to re-rank the outcome of our retrieval approaches. For the *Re-Rank Subjective* sub-task of the podcast segment retrieval task, we combine the scores of RoBERTArg – a pre-trained RoBERTA base model fine-tuned on an argument mining dataset [22] – and a simple dictionary-based subjectivity classifier. We use the final score as an additional signal to re-rank the outcome of our retrieval approaches.

We used our PyTorch [16] implementations available at: *github.com/sebastian-hofstaetter/matchmaker* furthermore we will make the trained & compressed dense retrieval model available on the HuggingFace model hub at: *huggingface.co/sebastian-hofstaetter*

## 2 BACKGROUND

In the following we give a quick overview of the methodology; we refer to the respective papers for more details. In our runs, we use the $\text{BERT}_{\text{DOT}}$ model as the dense retrieval system. It uses two independent BERT computations (each time pooling the `CLS` vector output) to obtain the query $q_{1:m}$ and passage $p_{1:n}$ representations. It then computes the retrieval score based on the dot product similarity of the two representations:

$$\begin{aligned} \vec{q} &= \text{BERT}([\text{CLS}; q_{1:m}]) \\ \vec{p} &= \text{BERT}([\text{CLS}; p_{1:n}]) \\ \text{BERT}_{\text{DOT}}(q_{1:m}, p_{1:n}) &= \vec{q} \cdot \vec{p} \end{aligned} \tag{1}$$

Table 1: Summary of our submitted TREC-DL'21 passage runs

| Run | Description |
|---|---|
| **TUW_DR_Base** | This is a baseline dense retrieval model (based on DistilBERT) trained on the MSMARCO-V1 training triples (using BM25 negative samples) and a simple RankNet loss with a batch size of 32 using the binary relevance labels, without any knowledge distillation. For inference we use ONNX runtime and BERT optimizations with fp16 (resulting vectors are also fp16). |
| **TUW_TAS-B_768** | We use our publicly available checkpoint of our TAS-Balanced trained DistilBERT dense retrieval model in a brute-force search configuration. For inference we use ONNX runtime and BERT optimizations with fp16 (resulting vectors are also fp16). |
| **TUW_TAS-B_ANN** | This TAS-Balanced trained model (based on DistilBERT) uses a compression layer at the end to produce 192 dimensional embeddings in fp16 (an 8x reduction to a default 768 dim output in fp32), we then indexed the vectors with HNSW (using 96 neighbors per vector). For inference we use ONNX runtime and BERT optimizations with fp16 (resulting vectors are also fp16). |

This architecture decouples the costly encoding from the search. For direct vector-based retrieval, we can store every passage in an (approximate) nearest neighbor index $I$. The retrieval of the top $k$ hits for a given query $q$ is then formalized as:

$$\text{top}_k \left\{ \vec{q} \cdot \vec{p} \,\middle|\, \vec{p} \in I \right\} \tag{2}$$

In this study, we use the *Standalone* and *TAS-Balanced* trained instances of $\text{BERT}_{\text{DOT}}$, developed by Hofstätter et al. [7]. The *Standalone* version is trained with binary relevance labels from MS MARCO [1]. The *TAS-Balanced* retriever is trained with pairwise and in-batch negative knowledge distillation using topic-aware sampling to compose batches.

We trained all our models on MSMARCO-v1 data and for the DL track evaluated it with the new MSMARCO-v2 collection. While stemming from the same query distribution, the v2 collection does have different passage selections and a drift in the crawl-time of the data. For the Podcast track we used the TREC-Podcast collection. In both cases we concatenated the page or episode title with the respective passage.

## 3 SIMPLE COMPRESSION

We propose a simple training process for compressing a dense retrieval model's output as part of the model, usable with any training approach as it does not alter the input interface or loss function. We run the following steps:

(1) Train the $\text{BERT}_{\text{DOT}}$ model with full capacity (for DistilBERT [21] this is 768 dimensions) with a training method of your choice – we use our TAS-Balanced approach;
(2) Add a randomly initialized compression, or dimensionality reduction, layer after the CLS pooling to the fully trained model (we settled on 192 dimensions, a 4x reduction);
(3) Conduct a second full training pass of the training method of your choice without freezing any weights or training length constraints.

Step (2) is formalized as follows: we adapt $\text{BERT}_{\text{DOT}}$ (Eq. 1) with a single shared layer $W$ with dimensions $\mathbb{R}^{b \times c}$, where $b$ is the output dimension of BERT and $c$ is our target compression dimension:

$$\begin{aligned} \vec{q} &= \text{BERT}([\text{CLS}; q_{1:m}]) * W \\ \vec{p} &= \text{BERT}([\text{CLS}; p_{1:n}]) * W \end{aligned} \tag{3}$$

Table 2: Official TREC-DL'21 passage retrieval results.

| | Run | nDCG@10 | MRR@100 | MAP@100 |
|---|---|---|---|---|
| 1 | **TUW_DR_Base** | 0.4991 | 0.6768 | 0.1540 |
| 2 | **TUW_TAS-B_768** | 0.5619 | 0.7333 | 0.2093 |
| 3 | **TUW_TAS-B_ANN** | 0.5426 | 0.7015 | 0.1932 |

Concurrent related works have also tackled the output compression of dense retrieval models: Zhan et al. [26] created a training procedure to optimize the product quantization of dense retrieval output vectors. More closely to our procedure, Ma et al. [15] used a similar dimensionality reduction layer for a DPR training procedure [10]. However, interestingly they came to different conclusions than we did: that adding a single linear layer on top of a dense retriever does not work well and is easily outperformed by post-hoc PCA. We, on the other hand, find it to work quite well (even though we do not present thorough ablation studies in this technical report). We believe this might be attributed to the following differences in the workflows: 1) We use a more robust training procedure including knowledge distillation (TAS-Balanced vs. binary DPR), 2) We train our dense retriever first with full capacity first 3) While they also had a 2-step version Ma et al. [15] froze the BERT layers for the second training round.

## 4 DEEP LEARNING TRACK

We summarize our submitted DL track runs in Table 1. They are all pure dense retrieval results without costly re-ranking. We are mainly interested in answering two specific research questions. The first carefully tests our TAS-Balanced training method:

**RQ-DL-1** Does TAS-Balanced improve over a standalone trained dense retriever?

To answer this RQ, we compare rows 1 and 2 in Table 2. Both runs were created by the same: architecture, parameter count, inference, and indexing setups. The only difference is the training method: Standalone (row 1) vs. TAS-Balanced (row 2). The results clearly show a substantial difference in all metrics, with a 6 point margin in nDCG@10. This confirms our observations and ablation studies conducted as part of our TAS-Balanced paper.

Table 3: Summary of our submitted TREC-Podcast'21 runs

| Run | Description |
|---|---|
| **TUW_tasb192_ann** | This TAS-Balanced trained model (based on DistilBERT) uses a compression layer at the end to produce 192-dimensional embeddings in fp16 (an 8x reduction to a default 768-dim output in fp32); we then indexed the vectors with HNSW (using 128 neighbors per vector). |
| **TUW_tasb_cat** | We use our publicly available checkpoint of our TAS-Balanced trained DistilBERT dense retrieval model[1] in a brute-force search configuration. We apply a knowledge distilled $BERT_{CAT}$ re-ranking model[2] to generate the final ranking. |
| **TUW_hybrid_cat** | We use our TAS-Balanced trained DistilBERT model[1] (trained on MS MARCO passage collection v1) to encode the segments and generate a faiss index. We generate a BM25 sparse index (Pyserini [11]). Using both indices, we follow a hybrid sparse-dense retrieval approach. |
| **TUW_hybrid_ws** | We combine a BM25 (Pyserini [11]) run and our full TAS-B[1] run (both top-1000) and then apply a knowledge distilled $BERT_{CAT}$ re-ranking model[2] to generate the final ranking. |
| **Re-Ranking Task** | **Approach** |
| **Entertaining** | We utilize a pre-trained BERT-based emotions classifier[3] trained on the GoEmotions dataset [3]. We use the $1 - neutral\_score$ as a signal for entertainment. We generate a final score, and thus a ranking, using a weighted sum over entertainment and relevance scores. We tune the weights by setting a guardrail of minus 5 points of the respective model's nDCG@30 considering the test set of TREC-Podcast'20. |
| **Subjective** | We utilize RoBERTArg[4], which is trained on an argument/non-argument labeled dataset [22]. We take the arithmetic mean of the argument score and a simple dictionary-based subjectivity score[5]. Our final re-ranking score is a weighted sum over the final subjectivity score and relevance score. We tune the weights by setting a guardrail of minus 5 points of the respective model's nDCG@30 considering the test set of TREC-Podcast'20. |
| **Discussion** | *Not participated.* |

[1]https://huggingface.co/sebastian-hofstaetter/distilbert-dot-tas_b-b256-msmarco
[2]https://huggingface.co/sebastian-hofstaetter/distilbert-cat-margin_mse-T2-msmarco
[3]https://huggingface.co/monologg/bert-base-cased-goemotions-original
[4]https://huggingface.co/chkla/roberta-argument
[5]https://textblob.readthedocs.io

For our next RQ, we utilize our TAS-B training process and apply our output compression technique as well as an approximate nearest neighbor indexing technique and answer:

**RQ-DL-2** Does our simple compression with approximate nearest neighbor search keep up with a full TAS-B retriever?

To answer this RQ, we compare rows 2 and 3 in Table 2. Unfortunately, we do not have a spotless ablation setup. In row 2, we mixed our compressed to 192 dimensions model with HNSW approximate nearest neighbor search to form a closer-to-realistic-production system. However, we can still evaluate it as a lower-bound for the compression and a lower-bound for the ANN search compared to the full TAS-B (row 2). The blind evaluation results follow the path of our internal validation on MSMARCO-v1: We do lose roughly 2 points nDCG@10 compared to the full + uncompressed search. We see this as a good result, as we are still comfortably in front of a standalone baseline (row 1) with more than 4 points nDCG@10 gain.

## 5 PODCAST TRACK

We summarize our submitted TREC-Podcast'21 runs in Table 3. For all runs, we consider the concatenation of episode title and podcast segment as documents, and we only take the query field of the TREC-topics as queries. Our TAS-B trained retrieval models are trained on MSMARCO-V1. However, queries in TREC-Podcast are shorter and differently structured, and documents are longer and transcribed from speech. Thus, we investigate:

Table 4: TREC-Podcast'21 ad-hoc retrieval results.

| | Run/Model | nDCG @10 | nDCG @30 | nDCG @1K | P@10 |
|---|---|---|---|---|---|
| 1 | **BM25** | .2486 | .2725 | .4467 | .3080 |
| 2 | **TUW_tasb192_ann** | .3082 | .2970 | .4286 | .3720 |
| 3 | **TUW_tasb_cat** | .3255 | .3289 | .4952 | .3860 |
| 4 | **TUW_hybrid_cat** | .3234 | .3358 | .5315 | .3860 |
| 5 | **TUW_hybrid_ws** | .3205 | .3283 | .5255 | .3840 |

**RQ-P-1** To what extent does our compressed TAS-B trained dense retriever, trained on MSMARCO-V1, generalize to the TREC-Podcast'21 retrieval task?

We evaluate our runs on the official TREC-Podcast'21 qrels, and present the results in Table 4. Our compressed TAS-B trained dense retriever substantially outperforms BM25 and shows a margin of 6 points in nDCG@10. Furthermore, it shows comparable results to our full TAS-B trained retrieval with knowledge distilled $BERT_{CAT}$

Table 5: TREC-Podcast'21 *Re-Rank Entertaining* results.

| | Run/Model | nDCG @10 | nDCG @30 | nDCG @1K | P@10 |
|---|---|---|---|---|---|
| 1 | **BM25** | .1104 | .1420 | .2705 | .1175 |
| | Runs w/ re-ranking (as submitted) | | | | |
| 2 | **TUW_tasb192_ann** | .1366 | .1443 | .2273 | .1175 |
| 3 | **TUW_tasb_cat** | .1353 | .1514 | .2691 | .1450 |
| 4 | **TUW_hybrid_cat** | .1437 | .1582 | .3065 | .1500 |
| 5 | **TUW_hybrid_ws** | .1207 | .1481 | .2869 | .1325 |
| | Runs w/o re-ranking | | | | |
| 6 | **TUW_tasb192_ann** | .1549 | .1689 | .2475 | .1425 |
| 7 | **TUW_tasb_cat** | .1443 | .1700 | .2858 | .1550 |
| 8 | **TUW_hybrid_cat** | .1430 | .1746 | .3176 | .1525 |
| 9 | **TUW_hybrid_ws** | .1332 | .1716 | .3078 | .1500 |

Table 6: TREC-Podcast'21 *Re-Rank Subjective* results.

| | Run/Model | nDCG @10 | nDCG @30 | nDCG @1K | P@10 |
|---|---|---|---|---|---|
| 1 | **BM25** | .1971 | .2187 | .4093 | .2350 |
| | Runs w/ re-ranking (as submitted) | | | | |
| 6 | **TUW_tasb192_ann** | .2605 | .2501 | .3940 | .3200 |
| 7 | **TUW_tasb_cat** | .2533 | .2657 | .4565 | .2775 |
| 8 | **TUW_hybrid_cat** | .2433 | .2600 | .4884 | .2675 |
| 9 | **TUW_hybrid_ws** | .2556 | .2691 | .4847 | .2925 |
| | Runs w/o re-ranking | | | | |
| 2 | **TUW_tasb192_ann** | .2765 | .2687 | .4064 | .3350 |
| 3 | **TUW_tasb_cat** | .2572 | .2720 | .4577 | .3075 |
| 4 | **TUW_hybrid_cat** | .2453 | .2660 | .4867 | .2950 |
| 5 | **TUW_hybrid_ws** | .2733 | .2903 | .4996 | .3275 |

re-ranking (row 3), with only -2 points loss in nDCG@10. This demonstrates the great potential of our efficient yet effective compressed TAS-B trained dense retriever.

Previous work has shown that sparse and dense signals are complementary, and thus, a hybrid approach usually yields effectiveness gains [11]. Therefore, we study:

**RQ-P-2** To what extent does combining our TAS-B dense retriever with a BM25 sparse retriever improve the performance on the TREC-Podcast'21 retrieval task?

We follow two different approaches to combine sparse and dense retrieval. In our first approach, we merge top-1000 retrieved documents of BM25 and our TAS-B trained dense retriever and skip duplicates. Then we re-rank the outcome with our knowledge distilled $BERT_{CAT}$ to obtain the final ranking. For our second approach, we generate a dense (FAISS) index using our TAS-B trained dense retriever and a sparse index using BM25. Then we apply weighted interpolation on the individual results as described and implemented in Pyserini [11]. Both approaches show similar performance and substantially outperform BM25 (compare row 1 to row 4 and row 5 in Table 4) with a margin of 8 points in nDCG@10. However, they do not show gains over our dense retrieval and re-ranking approach (compare row 3 to row 4 and row 5 in Table 4).

Besides ad-hoc retrieval, the segment retrieval task of TREC-Podcast'21 also contains *Re-Rank Entertaining*, *Re-Rank Subjective*, and *Re-Rank Discussion* tasks. Reddy et al. [17] highlight the relation of linguistic style to peoples' engagement with podcasts. Following this line of research in the re-ranking tasks we investigate:

**RQ-P-3** To what extent does incorporating auxiliary information, i.e., emotion and argument-mining scores, to the retrieval scores improve the performance on the TREC-Podcast'21 *Re-Rank Entertainment* and *Re-Rank Subjective* tasks?

Experiments by Reddy et al. [17] show that high engagement is related to more positive and less negative emotions and sentiment. In this work, we use a fine-grained emotions classifier fine-tuned on the GoEmotions dataset [3]. While there might be a correlation between entertainment and engagement, entertaining for the *Re-Rank Entertaining* task is described as *"amusing and entertaining to the listener, rather than informative or evaluative"*[1]. Based on this description and the lack of data for training and tuning, we only consider one minus the neutral score as a signal for re-ranking. Our submitted runs show substantial gains over BM25 with a 1-4 points margin in nDCG@10 (compare row 1 with rows 6-9 in Table 5). However, our experiments show no gains, and in fact losses if we compare the non-re-ranked models against our submitted re-ranked models (compare rows 2-5 to rows 6-9 in Table 5).

We utilize a BERT-based argument/non-argument classifier and a simple dictionary-based subjectivity classifier for the *Re-Rank Subjective* task. We combine the classification scores with the relevance scores to re-rank the top-1000 retrieved podcasts. Our submitted runs substantially outperform BM25 with a 4-6 points margin in nDCG@10 (compare row 1 with rows 6-9 in Table 6). However, following the *Re-Rank Entertaining* task, our experiments show losses if we compare the non-re-ranked models against our submitted re-ranked models (compare rows 2-5 to rows 6-9 in Table 6).

Although our re-ranking seems insufficient in contrast to our expectations, the released Podcast'21 evaluation data will further enable us to conduct proper training and evaluation to better incorporate auxiliary information in future work.

After their initial effectiveness leaps for in-domain training and evaluation of DR approaches [4, 7, 12, 25], a major question becomes the out-of-domain, or zero-shot, effectiveness of these neural models [23]. Our first participation at the TREC-Podcast track gives us an excellent opportunity to study the effects of pool bias [13, 14, 20, 24, 27], and discuss its impact on take away messages:

**RQ-P-4** What can we learn about out-of-pool evaluation for DR by comparing 2020 (out-of-pool) with 2021 (in-pool) TREC-Podcast results?

We have an ideal setup for comparing the two TREC years, as the Podcast track utilizes the same collection, and a similar-typed yet distinct set of queries for both years. We had no in-domain training data for our neural rankers (except for tuning the single sparse-dense hybrid score weighting parameter of the run TUW_hybrid_ws).

Judging from the overview paper of last year's Podcast track [9], the initial retrieval of all runs was based on term-based matching, and no dense retriever participated.

[1]https://trecpodcasts.github.io/participant-instructions-2021.html

**Table 7: Comparing out-of-pool (2020) vs. in-pool (2021) ad-hoc retrieval evaluation for dense retrieval on TREC-Podcast.** ***J@k indicates the ratio of judged passages at depth k; nDCG–J@10 refers to using the -J option on trec_eval to only evaluate judged documents for nDCG@10***

| | | TREC-Podcast 2020 (out-of-pool) | | | | | | TREC-Podcast 2021 (in-pool) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Run** | J@10 | nDCG@10 | J@30 | nDCG@30 | nDCG–J@10 | bpref | J@10 | nDCG@10 | J@30 | nDCG@30 | bpref |
| 1 | **BM25 (Pyserini)** | 98% | .380 | 91% | .412 | .385 | .432 | 100% | .249 | 93% | .273 | .227 |
| 2 | **TUW_tasb192_ann** | 53% | .294 | 38% | .291 | .443 | .420 | 100% | .308 | 77% | .297 | .267 |
| 3 | **TUW_hybrid_ws** | 90% | .403 | 80% | .437 | .429 | .480 | 100% | .321 | 93% | .328 | .262 |
| 4 | **TUW_tasb_cat** | 66% | .355 | 57% | .381 | .450 | .473 | 100% | .326 | 84% | .329 | .281 |
| 5 | **TUW_hybrid_cat** | 67% | .355 | 57% | .381 | .465 | .541 | 100% | .323 | 88% | .336 | .288 |

In Table 7, we present the evaluation results for our 2021-runs for both TREC-Podcast years using common approaches to tackle pool bias [2, 19]. The evaluation using the 2020 judgments is completely out-of-pool (meaning our results did not participate in the pooling process for the judgments). Let us assume we only observe nDCG@10 values for TREC-Podcast 2020 without looking at the judged ratios. This scenario would conclude that zero-shot retrieval with TAS-B (row 2) completely fails, as it trails BM25 by 9 points nDCG@10. Additionally, re-ranking with $\text{BERT}_{\text{CAT}}$ (rows 4 & 5) looks like a failure with -3 points nDCG@10 compared to BM25. Only the hybrid BM25 + TAS-Balanced (row 3) shows a slight improvement over BM25.

Now, once we take judgment ratios into account, we see that these results might not represent a valid conclusion. We observe that the out-of-pool setting has an enormous impact on the ratio of judged (relevant or non-relevant) passages on our neural retrieval runs (rows 2, 4, 5). TAS-Balanced drops to 53% of judged passages at depth 10. All while BM25 is almost fully judged with 98%.

Turning to the 2021 results, the takeaway message turns completely: TAS-Balanced (row 2; still zero-shot) outperforms BM25 (row 1) by 6 points nDCG@10 in a fully judged setting. This is a 15 point nDCG@10 change. Furthermore, the sparse-dense hybrid (row 3) again improves over TAS-B. Interestingly, $\text{BERT}_{\text{CAT}}$ does not further help – this could be our first confirmed limitation in the zero-shot scenario, as we expected $\text{BERT}_{\text{CAT}}$ to outperform $\text{BERT}_{\text{DOT}}$ strongly. Once we observe nDCG@30, we again fall into the problem of pool-bias, as the judgment rate between BM25 and $\text{BERT}_{\text{DOT}}$ diverges substantially (as many runs probably used BM25 as their starting point, and we only have a guaranteed pooling depth < 30).

*So what can we take away from these results?* The question of the robustness and reliability of previously generated test collections is not new [13]. However, it becomes increasingly important as we – as a community – want to evaluate the new paradigm of trained dense retrieval on more than just a few web-focused collections [23]. While we do not presume to generalize from this one observation on TREC-Podcast, we see a striking divide in results between in-pool and out-of-pool evaluation of simple term-based BM25 and neural ranking approaches. We caution that other term-based-retrieval-pooled collections might show similar results. Therefore, we want to highlight the great importance and our gratitude of continuous TREC-style evaluation campaigns, which are the most robust way of evaluating this increasingly diverse set of indexing approaches.

## REFERENCES


[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew Mcnamara, Bhaskar Mitra, and Tri Nguyen. 2016. MS MARCO : A Human Generated MAchine Reading COmprehension Dataset. In *Proc. of NIPS*.

[2] Chris Buckley and Ellen M Voorhees. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 25–32.

[3] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547* (2020).

[4] Yingqi Qu Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. *arXiv:2010.08191* (2020).

[5] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. *arXiv:2010.02666* (2020).

[6] Sebastian Hofstätter and Allan Hanbury. 2020. Evaluating Transformer-Kernel Models at TREC Deep Learning 2020. In *Proc. of TREC*.

[7] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proceedings of the 44rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*.

[8] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2019. TU Wien @ TREC Deep Learning '19 – Simple Contextualization for Re-ranking. In *Proc. of TREC*.

[9] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth JF Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu. 2021. Trec 2020 podcasts track overview. *arXiv preprint arXiv:2103.15953* (2021).

[10] 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)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[11] 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 International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2356–2362. https://doi.org/10.1145/3404835.3463238

[12] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling Dense Representations for Ranking using Tightly-Coupled Teachers. *arXiv:2010.11386* (2020).

[13] Aldo Lipani. 2016. Fairness in information retrieval. In *Proc. of SIGIR*.

[14] Xiaolu Lu, Alistair Moffat, and J Shane Culpepper. 2016. The effect of pooling and evaluation depth on IR metrics. *Information Retrieval Journal* 19, 4 (2016), 416–445.

[15] Xueguang Ma, Minghan Li, Kai Sun, Ji Xin, and Jimmy Lin. 2021. Simple and Effective Unsupervised Redundancy Elimination to Compress Dense Vectors for Passage Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

[16] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, et al. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

[17] Sravana Reddy, Mariya Lazarova, Yongze Yu, and Rosie Jones. 2021. Modeling Language Usage and Listener Engagement in Podcasts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1:*

*Long Papers)*. Association for Computational Linguistics, Online, 632–643. https://doi.org/10.18653/v1/2021.acl-long.52

[18] Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation* (2004).

[19] Tetsuya Sakai. 2007. Alternatives to Bpref. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands) *(SIGIR '07)*. Association for Computing Machinery, New York, NY, USA, 71–78. https://doi.org/10.1145/1277741.1277756

[20] Tetsuya Sakai. 2008. Comparing Metrics across TREC and NTCIR: The Robustness to System Bias. In *Proc. of CIKM*.

[21] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[22] Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic Argument Mining from Heterogeneous Sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3664–3674. https://doi.org/10.18653/v1/D18-1402

[23] 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:2104.08663 [cs.IR]

[24] William Webber and Laurence A. F. Park. 2009. Score Adjustment for Correction of Pooling Bias. In *Proc. of SIGIR*.

[25] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[26] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Jointly Optimizing Query Encoder and Product Quantization to Improve Retrieval Performance. *arXiv preprint arXiv:2108.00644* (2021).

[27] Justin Zobel. 1998. How Reliable Are the Results of Large-Scale Information Retrieval Experiments?. In *Proc. of SIGIR*.