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 BERT\textsubscript{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 BERT\textsubscript{DOT} model and re-rank the outcome with our knowledge distilled BERT\textsubscript{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 BERT\textsubscript{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 BERT\textsubscript{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{align}
\hat{q} &= BERT([CLS; q_{1:m}]) \\
\hat{p} &= BERT([CLS; p_{1:n}]) \\
BERT_{DOT}(q_{1:m}, p_{1:n}) &= \hat{q} \cdot \hat{p}
\end{align}
We propose a simple training process for compressing a dense retriever to a single shared layer function. We run the following steps:

1. Train the \( BERT_{\text{DOT}} \) 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 \( BERT_{\text{DOT}} \) (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:

\[
\bar{q} = BERT([CLS; q_{t,n}]) \ast W \\
\bar{p} = BERT([CLS; p_{t,n}]) \ast W
\]

In this study, we use the Standalone and TAS-Balanced trained instances of \( BERT_{\text{DOT}} \), 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 data. For the Podcast track we used the TREC-Podcast collection. 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.

### 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.

### 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

<table>
<thead>
<tr>
<th>Run</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TUW_tasb192_ann</td>
<td>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).</td>
</tr>
<tr>
<td>TUW_tasb_cat</td>
<td>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.</td>
</tr>
<tr>
<td>TUW_hybrid_cat</td>
<td>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.</td>
</tr>
<tr>
<td>TUW_hybrid_ws</td>
<td>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.</td>
</tr>
</tbody>
</table>

Re-Ranking Task | Approach
--- | ---
Entertaining | We utilize a pre-trained BERT-based emotions classifier\(^3\) trained on the GoEmotions dataset \[3\]. We use the \(1 - \text{neutral score}\) 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 guaradnial 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 guaradnial of minus 5 points of the respective model’s nDCG@30 considering the test set of TREC-Podcast'20.

Discussion
Not participated.

---

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:

**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\(_{CAT}\) re-ranking model.

---

<table>
<thead>
<tr>
<th>Run/Model</th>
<th>nDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@10</td>
</tr>
<tr>
<td>BM25</td>
<td>.2486</td>
</tr>
<tr>
<td>TUW_tasb192_ann</td>
<td>.3082</td>
</tr>
<tr>
<td>TUW_tasb_cat</td>
<td>.3255</td>
</tr>
<tr>
<td>TUW_hybrid_cat</td>
<td>.3234</td>
</tr>
<tr>
<td>TUW_hybrid_ws</td>
<td>.3205</td>
</tr>
</tbody>
</table>
we generate a dense (FAISS) index using our TAS-B trained dense retrieval and re-ranking approach (compare row 3 to row 4 and row 5 in Table 4). 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 Entertaining 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”.

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.

---

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

<table>
<thead>
<tr>
<th>Run/Model</th>
<th>nDCG</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25</td>
<td>.1104</td>
<td>.1420</td>
</tr>
<tr>
<td>TUW_tashb192_ann</td>
<td>.1366</td>
<td>.1443</td>
</tr>
<tr>
<td>TUW_tashb_cat</td>
<td>.1353</td>
<td>.1514</td>
</tr>
<tr>
<td>TUW_hybrid_cat</td>
<td>.1437</td>
<td>.1582</td>
</tr>
<tr>
<td>TUW_hybrid_ws</td>
<td>.1207</td>
<td>.1481</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Run/Model</th>
<th>nDCG</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25</td>
<td>.1971</td>
<td>.2187</td>
</tr>
<tr>
<td>TUW_tashb192_ann</td>
<td>.2605</td>
<td>.2501</td>
</tr>
<tr>
<td>TUW_tashb_cat</td>
<td>.2533</td>
<td>.2657</td>
</tr>
<tr>
<td>TUW_hybrid_cat</td>
<td>.2433</td>
<td>.2600</td>
</tr>
<tr>
<td>TUW_hybrid_ws</td>
<td>.2556</td>
<td>.2691</td>
</tr>
</tbody>
</table>

---

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).
Table 7: Comparing out-of-pool (2020) vs. in-pool (2021) ad-hoc retrieval evaluation for dense retrieval on TREC-Podcast.

<table>
<thead>
<tr>
<th>Run</th>
<th>TREC-Podcast 2020 (out-of-pool)</th>
<th>TREC-Podcast 2021 (in-pool)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>J@10 nDCG@10</td>
<td>J@10 nDCG@10</td>
</tr>
<tr>
<td>1 BM25 (Pyserini)</td>
<td>98% .380 91% .412</td>
<td>100% .249 93% .273</td>
</tr>
<tr>
<td>2 TUW_tash192_ann</td>
<td>53% .294 38% .291</td>
<td>100% .308 77% .297</td>
</tr>
<tr>
<td>3 TUW_hybrid_ws</td>
<td>90% .403 80% .437</td>
<td>100% .321 93% .328</td>
</tr>
<tr>
<td>4 TUW_tash_cat</td>
<td>66% .355 57% .381</td>
<td>100% .326 84% .329</td>
</tr>
<tr>
<td>5 TUW_hybrid_cat</td>
<td>67% .355 57% .381</td>
<td>100% .323 88% .336</td>
</tr>
</tbody>
</table>

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 its trails BM25 by 9 points nDCG@10. Additionally, re-ranking with BERTCAT (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, BERTCAT does not further help – this could be our first confirmed limitation in the zero-shot scenario, as we expected BERTCAT to outperform BERTDOT strongly. Once we observe nDCG@30, we again fall into the problem of pool-bias, as the judgment rate between BM25 and BERTDOT 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


