Spotify at the TREC 2020 Podcasts Track: Segment Retrieval

Yongze Yu[1], Jussi Karlgren[1], Hamed Bonab[2]*,
Ann Clifton[1], Md Iftekhar Tanveer[1], Rosie Jones[1]

[1] Spotify
[2] University of Massachusetts Amherst

In this notebook paper, we present the details of baselines and experimental runs of the segment retrieval task in TREC 2020 Podcasts Track. As baselines, we implemented traditional IR methods,i.e. BM25 and QL, and the neural re-ranking BERT model pre-trained on MS MARCO passage re-ranking task. We also detail experimental runs of the re-ranking model fine-tuned on additional external data sets from (1) crowd-sourcing, (2) automatically generated questions, and (3) episode title-description pairs.

1 Introduction

The TREC 2020 Podcasts track[1] included an an Ad-hoc Segment Retrieval task[6]. High-quality search of topical content of podcast episodes is challenging. Existing podcast search engines index the available metadata fields for the podcast as well as textual descriptions of the show and episode, but these descriptions often fail to cover the salient aspects of the content[2]. Improving and extending podcast search is limited by the availability of transcripts and the cost of automatic speech recognition. Therefore, this year's task is set to fixed-length segment retrieval: given an arbitrary query (a phrase, sentence, or set of words), retrieve topically relevant segments from the data. These segments can then be used as a basis for topical retrieval, for visualization, or other downstream purposes [5]. Figure 1 shows an example of the retrieval topics.

```
<topic>
<num>34</num>
<query>halloween stories and chat</query>
<type>topical</type>
<description>I love Halloween and I want
             to hear stories and con-
             versations about things
             people have done to celebrate
             it. I am not looking for
             information about the history
             of Halloween or generalities
             about how it is celebrated,
             I want specific stories
             from individuals.
</description>
</topic>
```

Figure 1: An example of the retrieval topics.

**Work done while at Spotify

[1]https://podcastsdataset.byspotify.com/

In this notebook paper, we describe the details of our submissions to the TREC Podcasts Track 2020. Based on the task guidelines, a segment, for the purposes of the document collection, is a two-minute chunk with one minute overlap and starting on the minute; e.g. 0.0-119.9 seconds, 60.0-179.9 seconds, 120.0-239.9 seconds, etc. This creates 3.4M segments in total from the document collection with the average word count of 340 ± 70. These segments will be used as passage units and the retrieved passage is then judged by assessors. We have implemented two baseline models as well as three experimental runs. The baselines are traditional IR models and neural re-ranking from the top-N passages. The experimental runs used the re-ranking model fine-tuned on various synthetic or external data labels from the data corpus (Table 1).

2 Methods

2.1 Information Retrieval Baselines

We implemented as baselines the standard retrieval models BM25 and query likelihood (QL), using the Pyserini package [2], built on top of the open-source Lucene[3] search library. Stemming was performed using the Porter stemmer. The models BM25 and QL are used with Anserini's default parameters.[4] We created two document indexes, one with transcript segments only and the other with title and descriptions concatenated to each transcript segment. For each topic, there is a short query phrase and a sentence-long description. The short query phrase was used as the query term for searching the index. Up to the top 1000 passages were submitted in runs on the 50 test topics.

2.2 Neural re-ranking baseline

The BERT re-ranking system is the current state-of-the-art in search [8]. It been implemented for passage and document ranking systems on many document collections, including MS-MARCO [1], TREC-CAR[4], and Robust04 [3].

The system can be described as two main stages. First, a large number of possibly relevant documents to a given question are retrieved from a corpus by a standard mechanism, such as BM25. In the second stage, passage re-ranking, each of these documents is scored and re-ranked by a more computationally-intensive method.

The job of the re-ranker is to estimate a score s_i of how relevant a candidate passage d_i is to a query q. The query and passage pair is fed into the model as sentence A and sentence B with BERT tokenization[5]. The query was truncated to have at most 128 tokens, and the passage text was truncated so that the concatenation of query, passage, and separator tokens stays within the maximum length of 512 tokens. To evaluate how much the segments could be truncated, we used the organizer-provided 8-topic 609-example "training" sets as our evaluation examples. Using topic descriptions (with avg. 39 tokens) as sentence A, 13% (82 out of 609) of segment texts are truncated and 37 ± 27 tokens are removed in 82 truncated texts.

A BERT-LARGE model was used as a binary classification model, that is, the [CLS] vector was used as input to a single layer neural network to obtain the probability of the passage being relevant. The pre-trained BERT model was fine-tuned on MS MARCO dataset with 400M tuples of a query, relevant and non-relevant passages with point-wise cross-entropy loss:

$$loss = -\sum_{j \in J} j \log(s_j) - (1-j)\log(1-s_j)) \qquad (1)$$

where J is the set of indices of the passages in top documents retrieved with BM25.

The model learned that knowledge of query-document relevance can be transferred to other

[2]https://github.com/castorini/pyserini – a Python front end to the Anserini open-source information retrieval toolkit [13] .

[3]https://lucene.apache.org

[4]BM25 parameter settings $k = 0.9, b = 0.4$; QL setting for Dirichlet smoothing $\mu = 1000$

[5]BERT tokenization is based on WordPiece.

Run	Description	Topic Inputs	Indexed fields
BM25	Standard information retrieval algorithm developed for the Okapi system	query	Transcript only
QL	Query Likelihood; Standard information retrieval	query	Transcript only
RERANK-QUERY	BM25 (using query) + BERT re-ranking model (using the query of the topic as the input)	query	Transcript only
RERANK-DESC	BM25 (using query) + BERT re-ranking model (using the description of the topic as the input)	query + description	Transcript only
BERT-DESC-S	Same as RERANK-DESC except that the re-ranking model was fine-tuned on extra crowd-sourced data	query + description	Transcript only
BERT-DESC-Q	Same as RERANK-DESC except that the re-ranking model was fine-tuned on synthetic data from generated questions	query + description	Transcript only
BERT-DESC-TD	Same as RERANK-DESC except that the re-ranking model was fine-tuned on synthetic data from episode title and descriptions	query + description	Transcript only

Table 1: Run names and short descriptions.

similar datasets. Therefore, we implement the neural baselines using the BERT re-ranking model as described above by Nogueira et. al.[8] without further parameter-tuning.

Another nontrivial question is whether we should use the phrase-like query or sentence-like description of the topic as the input to the re-ranking model. For exploration purposes, we prepared two baseline runs using query and description, denoted as RERANK-QUERY and RERANK-DESC respectively. The top 50 passages retrieved by BM25[6] were scored and re-ranked for each test topic. We submitted the top-50 re-ranked passages for test topics.

2.3 Fine-tuning

One of the the limitations of BERT re-ranking is that it is trained on MS Marco dataset, which is different in the domain and topics from podcast ranking. The models have not seen the corpus even once. Therefore, we performed more fine-tuning on the re-ranking models with external and synthetic examples as described below.

2.3.1 Crowd-sourced labels

One of challenges for a new dataset is the limited set of labeled training data. The TREC 2020 Podcasts Track organizers provided 609 training labels for 8 topics. But these examples are too few to train a reasonable model from scratch. Therefore, we developed 30 more topics in the same format as the training/test topics[7], and use the those eight "training" topic as our validation topics. We collected the union of the top-20 segments from BM25 and QL model and fed the examples to a crowd-sourcing tool Appen[8]. We thus obtained 919 relevance labels on the Excellent-Good-Fair-Bad (EGFB) scale (denoted as 3,2,1,0 respectively) from crowd-sourced annotators for those 30 development topics. The distribution of labels is shown in Figure 2. The 4-point scale labels were transformed to binary labels (EG to one, FB to zero) and then fed into the BERT binary classification model. The topic description was chosen as the sentence A in the BERT model. The model is fine-tuned from the baseline model for 10 epochs using the AdamW Optimizer[9]. The retrieval setup is similar to the neural re-ranking baselines, but we submitted the run as BERT-DESC-S with the scores computed from this fine-tuned model (the 'S' is DESC-S is for "supervised").

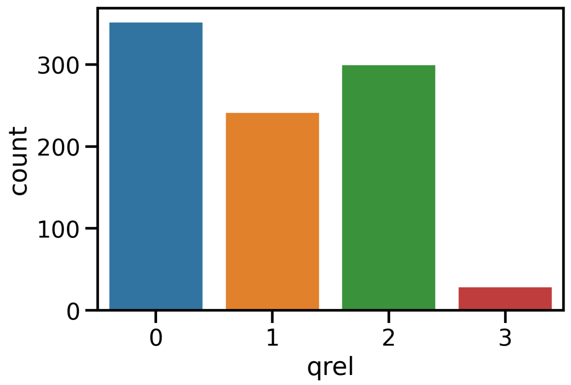


Figure 2: Counts of crowd-sourced relevance labels for 30 development topics. (3:Excellent,2:Good,1:Fair,0:Bad)

2.3.2 Automatically Generated Questions

In an attempt to generate large amounts of training data for domain adaption and further fine-tuning to our podcast corpus content, we leveraged the recently introduced *doc2query* model [10]. The authors used existing MS-MARCO dataset in the reverse order; given a document generate the query. A sequence-to-sequence model is trained using the MS-MARCO's relevant passages along with the relevant query questions. The trained model is expected to produce a list of relevant questions (or queries) given a passage. The original *doc2query* model was trained from scratch on a Transformer neural model [12]. The authors further improved the question generation model using a pre-trained

[6]Top passages were retrieved using the topic query only.

[7]To simplify the task, we included only `topical` topics, and not `known-item`.

[8]https://www.appen.com

[9]Optimizer AdamW[7] is implemented using Transformers(https://github.com/huggingface/transformers) with the initial learning rate set to 1×10^{-6} and linear decay of the learning rate

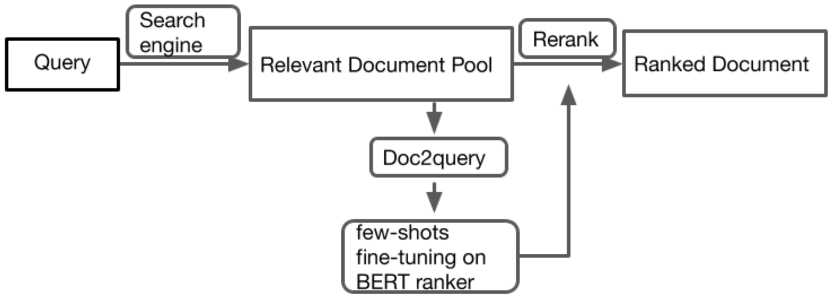


Figure 3: Scheme of re-ranking model with few-shots tuning on generate queries.

text-to-text model, named T5[11], using the same training data. It has shown better performance on downstream retrieval tasks and the model is denoted as docTTTTTquery[9].

Along with the strategy of feeding the re-ranking model with additional data from the current corpus, we propose a few-shot tuning method using the automatically generated questions as synthetic queries, with their source-passages as the corresponding relevant documents. The scheme is shown in Figure 3. For each topic, we first retrieve the top-50 segments using BM25, then we generate 5 queries or questions using the docTTTTTquery model for each segment. These question-segment pairs are treated as positive labels for fine-tuning the BERT model. The negative labels can be generated using different sampling strategies. Due to time limitations, we implemented only one strategy for this experiment. For each segment retrieved per query, we randomly sample 5 questions generated by other segments but within the same topic. The reasoning for this strategy is that the generated negative questions should be close to the positive questions but still distinguishable from one segment to another segment. An example of generated questions and labels is shown in Figure 4.

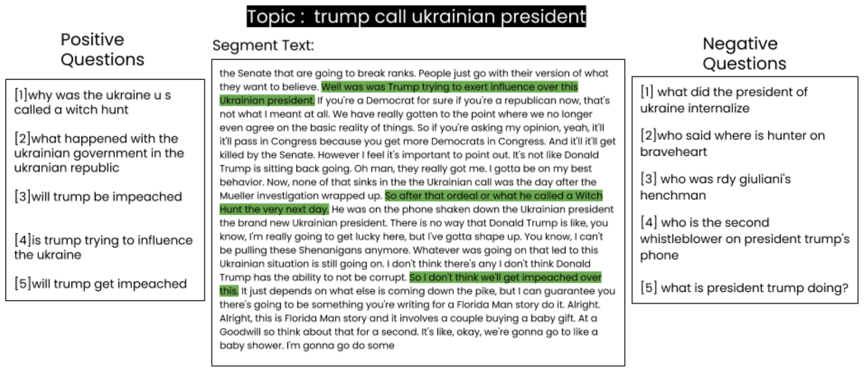
The retrieval setup is similar to neural re-ranking baselines. We fine-tuned the BERT re-ranking model using the synthetic examples as described above on the test topics. After fine-tuning, we used the scores using the topic description and segment text from this fine-tuned model and submitted the run as BERT-DESC-Q.

2.3.3 Title and Description

Unlike other corpora, the podcast dataset contains plentiful metadata which is extracted from RSS feed of the podcast episodes. The metadata, especially the text title and description, could contain important information about the episode. More importantly, many named entities which could be mistranscibed by the automatic speech recognition system are written in description. Therefore, linking the episode title and description to the transcripts could potentially help named entity matching in the re-ranking models.

We pre-process the episode title and description the same way as the topic query and description. We first cleaned non-topical content in the title and description, e.g. $Ep.5, S4E3, Episode5$ patterns in titles using a regular expression[10] as well as advertisements and links in descriptions. Then, we use the cleaned episode title as a search query, calculating the BM25 ranking score for each segment within the episode. Then, we input the episode description as sentence A to BERT re-ranking model. The top-3 ranked segments by BM25 score were used as positive labels and the bottom-3 at 50th ranked segments were used as negative labels. The model was fine-tuned on the 100K examples randomly sampled from the synthetic examples above. Similarly to other experiments, the top-50 segments by ranking score were selected and submitted for the test topics. This submission is called BERT-DESC-TD.

[10] $(Ep|Episode|Season|S|E)(.|-|)(|\s)\d+$



Topic : trump call ukrainian president

Positive Questions

[1]why was the ukraine u s called a witch hunt

[2]what happened with the ukrainian government in the ukranian republic

[3]will trump be impeached

[4]is trump trying to influence the ukraine

[5]will trump get impeached

Segment Text:

the Senate that are going to break ranks. People just go with their version of what they want to believe. Well was was Trump trying to exert influence over this Ukrainian president. If you're a Democrat for sure if you're a republican now, that's not what I meant at all. We have really gotten to the point where we no longer even agree on the basic reality of things. So if you're asking my opinion, yeah, it'll it'll pass in Congress because you get more Democrats in Congress. And it'll it'll get killed by the Senate. However I feel it's important to point out. It's not like Donald Trump is sitting back going. Oh man, they really got me. I gotta be on my best behavior. Now, none of that sinks in the the Ukrainian call was the day after the Mueller investigation wrapped up. So after that ordeal or what he called a Witch Hunt the very next day. He was on the phone shaken down the Ukrainian president the brand new Ukrainian president. There is no way that Donald Trump is like, you know, I'm really going to get lucky here, but I've gotta shape up. You know, I can't be pulling these Shenanigans anymore. Whatever was going on that led to this Ukrainian situation is still going on. I don't think there's any I don't think Donald Trump has the ability to not be corrupt. So I don't think we'll get impeached over this. It just depends on what else is coming down the pike, but I can guarantee you there's going to be something you're writing for a Florida Man story do it. Alright. Alright, this is Florida Man story and it involves a couple buying a baby gift. At a Goodwill so think about that for a second. It's like, okay, we're gonna go to like a baby shower. I'm gonna go do some

Negative Questions

[1] what did the president of ukraine internalize

[2]who said where is hunter on braveheart

[3] who was rdy giuliani's henchman

[4] who is the second whistleblower on president trump's phone

[5] what is president trump doing?

Figure 4: An example of the generated questions and labels. The sentences related to positive questions are highlighted.

2.4 Extension: Search with Full Transcript

In order to understand the benefits from searching on transcript content as compared to only on title and descriptions. We have conducted a set of experiments on a document-level retrieval task. The task is to rank the podcast episode given a search query. We convert judged query segment annotations using $qrel_d = \max(qrel_{s_1}, qrel_{s_2}, ..., qrel_{s_n})$ to document relevance, which is the maximum of the relevance score of the segments in the given document.

The episode's title, episode's title with description, episode's title with description plus show's title with description, episodes' transcript, and episode's transcript with title and description are indexed, respectively. We performed the document-level search task using 50 test topics and the standard retrieval model $BM25$. The results are shown in Table 2.

3 Results

According to the document-level search results in Table 2, the episode description entails more relevant information than episode title. Searching on combined episode title and description outperforms searching on individuals. This means episode title and description are mutually supportive and supplementary. However, adding information of show title and description does not help search result significantly. It is because we are measuring the episode level relevance in this metric, and show level information may bring false positive signals to the search results. In another word, the episodes in a relevant show may not be all relevant to the search query. Instead, the transcript containing the details of content information significantly improves the search result. The result shows the advantage of transcript search against regular search on episode title and description. When all information, transcript, title and description, are used, we can achieve the best retrieval performance.

The submitted runs on 50 test topics from the systems were evaluated by the NIST assessors. The annotation depth was 20 across different runs. Based on the annotation depth and submitted runs, we use NDCG@10 and NDCG@20 as our evaluation metrics. The results are shown in Table 3. The neural re-ranking baselines significantly outperforms traditional IR methods. Reranking on topic query and topic description get similar mean scores. However, in order to understand the distribution of score on individual topics, we plot the score for RERANK-DESC and RERANK-QUERY models in Figure 5. The ranking on topic description tends to push extreme score from RERANK-QUERY to center. In another word, even though the mean nDCG cross topics are same for both model, the variance of RERANK-DESC is lower. Reranking on topic description has more resistances to extreme queries. However, the three experimental runs do not show significant difference against the re-ranking baseline on description.

And the model fine-tuned on crowd-sourced data performed slightly better than other two. Possible reasons are (1) the ranking depth, which is 50, was set too shallow; (2) the negative sampling strategy should be better designed and evaluated.

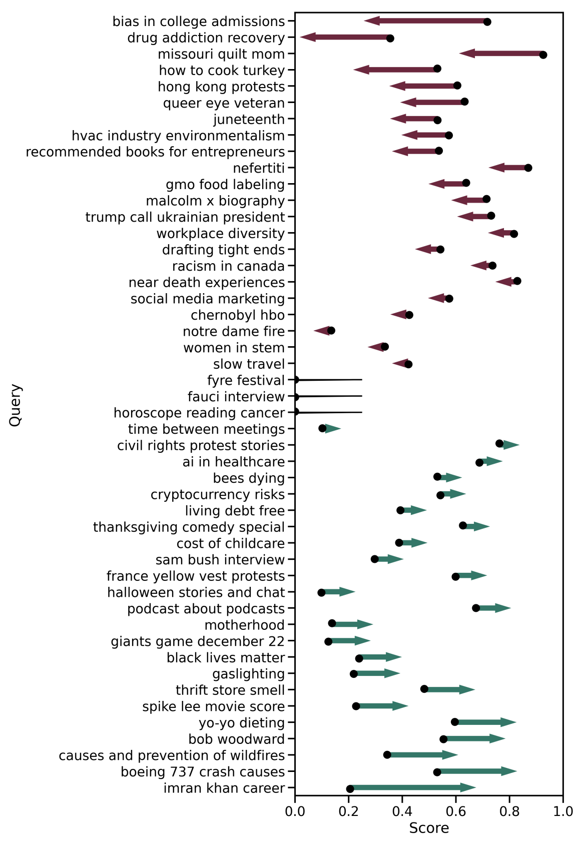


Figure 5: nDCG@20 score difference plot between RERANK-QUERY and RERANK-DESC for each individual topic. The arrow starts from RERANK-QUERY score and ends at RERANK-DESC score.

list of runs	nDCG@20	nDCG@10
BM25	0.386	0.366
QL	0.380	0.366
RERANK-QUERY	0.469	0.457
RERANK-DESC	0.469	0.459
BERT-DESC-S	**0.473**	**0.461**
BERT-DESC-Q	0.433	0.420
BERT-DESC-TD	0.464	0.456

Table 3: Results of test topics across different runs.

4 Conclusion

We have presented the details of our TREC 2020 Podcasts Track segment retrieval task submissions. Our experiments included baseline runs using traditional information retrieval methods and neural re-ranking baseline using BERT models pre-trained for a similar task. We also provide three experimental runs by fine-tuning the re-ranking with various synthetic labels. Even if the performance of the fine-tuned models did not improve compared to the neural baseline models, we have been able to calibrate our approach for more general deployment and lay the ground for experimentation on next year's task, where we plan to direct our attention not only at the re-ranking stage using language models but also at the improving the 1st-stage recall-based retrieval models.

	nDCG	nDCG @ 30	P @ 10
Episode Title	0.22	0.19	0.12
Episode Description	0.32	0.27	0.17
Episode Title and Description	0.36	0.30	0.19
Episode Title and Description with Show Title and Description	0.37	0.30	0.20
Transcript Text	0.58	0.46	0.41
Transcript Text with Episode Title and Description	0.61	0.49	0.43

Table 2: The contribution of transcripts compared to title search on document-level search results.

References

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[2] Jana Besser, Katja Hofmann, and Martha A. Larson. An exploratory study of user goals and strategies in podcast search. In Joachim Baumeister and Martin Atzmüller, editors, *Proceedings from the workshop Lernen, Wissen & Adaptivität (LWA)*. Department of Computer Science, University of Würzburg, Germany, 2008.

[3] Zhuyun Dai and Jamie Callan. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988, 2019.

[4] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. TREC Complex Answer Retrieval Overview". In *The 26th Text Retrieval Conference (TREC 2017) Proceedings*, NIST Special Publication. National Institute of Standards and Technology, 2017.

[5] Maria Eskevich, Walid Magdy, and Gareth JF Jones. New metrics for meaningful evaluation of informally structured speech retrieval. In *Proceedings of the 34th European Conference on Information Retrieval (ECIR)*. Springer, 2012.

[6] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth J. F. Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu. TREC 2020 podcasts track overview. In *The 29th Text Retrieval Conference (TREC 2020) Proceedings*, NIST Special Publication. National Institute of Standards and Technology, 2021.

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[8] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

[9] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. From doc2query to docTTTTTquery. *Online preprint*, 2019.

[10] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21, 2020.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems (NIPS 2017)*, 2017.

[13] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of Lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.