Combine and Re-Rank:
The University of Maryland at the TREC 2020 Podcasts Track

Petra Galuščáková[1], Suraj Nair[1,2], and Douglas W. Oard[1,3]

[1]UMIACS
[2]Computer Science Department
[3]College of Information Studies
University of Maryland, College Park
galuscakova@gmail.com

Abstract

The University of Maryland submitted five runs to Task 1 of the TREC 2020 podcasts track. The best results, achieved by combining seven system variants and then re-ranking with using combinations of two neural models, achieved a 27% improvement in NDCG over a simple Indri baseline in the official evaluation.

1 Introduction

The TREC Podcasts track [5] continues the tradition of speech retrieval tracks that started with the Spoken Document Retrieval track and continued with the TREC Video Track (which now continues as TRECVID). We submitted five runs for the Ad Hoc Segment Retrieval Task, focusing on three research questions:

- *Q1*) Can neural re-ranking improve over ranking based on lexical evidence?
- *Q2*) Would the combination of multiple re-rankers be more effective than a single re-rankers?
- *Q3*) When combining re-rankers, is it more effective to re-rank before or after system combination?

All of our experiments were based on Automatic Speech Recognition (ASR) transcripts that were provided by the organizers; we made no direct use of the audio. We are making the code used to generate our runs freely available.[1]

2 The Test Collection

The main focus of the TREC 2020 Podcasts Track[2] is being able to effectively find information in a relatively large collection of podcasts. The access to the information was studied in two tasks: Ad-hoc Segment Retrieval (*Task 1*) and Summarization (*Task 2*). We participated only in Task 1, results for which we describe in this paper. The focus of this task was on finding the relevant 2 minutes long segments of the podcasts, which can be sometimes several hours long. The attention thus needs to be paid to both – the quality of the returned information and the starting point provided to the user.

[1]`https://github.com/galuscakova/podcasts`
[2]`https://podcastsdataset.byspotify.com`

The topics, documents, and relevance judgments used in the track were all provided by Spotify [3]. The Train set contained 8 training topics for which a total of 609 segments had been judged for relevance to some topic, 217 of which were judged as relevant to their associated topic to some degree. The test set contained 50 topics, with no relevance judgments. There were three types of topics: *topical*, *refinding* and *known item*. An example of each topic type is shown in the Table 1. The majority of the topics in the Test set (35) were topical, 8 were refinding topics, 7 were known-item topics. The Train set contained 6 topical topics, one refinding topic and one known-item topic. Each topic consists of a *title* that can be interpreted as a Web-like query and a more expressive *description* of the information need, similar to what might be said in conversation.

Type	Title	Description
Topical	black hole image	In May 2019 astronomers released the first-ever picture of a black hole. I would like to hear some conversations and educational discussion about the science of astronomy, black holes, and of the picture itself.
Refinding	story about riding a bird	I remember hearing a podcast that had a story about a kid riding some kind of bird. I want to find it again.
Known-item	daniel ek interview	Someone told me about a podcast interview with Daniel Ek, CEO of Spotify, about the founding and early days of Spotify. I would like to find the show and episode that contains that interview. Other interviews with Ek are relevant as well.

Table 1: Examples of different topic types from the Train set.

Graded relevance judgments, ranging from 0 to 4, were provided for every allowable replay start point (which for Task 1 was defined as each full minute from the beginning of the podcast). The highest relevance grade (4) could be only assigned to a known-item or refinding topic; it was defined to indicate the '*segment that is the earliest entry point into the one episode that the user is seeking*'. The highest relevance grade for a topical topic (3) was defined to indicate that '*the segment conveys highly relevant information, is an ideal entry point for a human listener, and is fully on topic*'. Judged segments could start at any full minute, and were two minutes long (or possibly shorter, for the final segment of a podcast).

The same set of podcasts was searched for Train and Test. It consisted of 105,360 podcast recordings totalling over 47,000 hours of audio, and it is thus one of the largest available speech retrieval test collections. One-best transcripts were provided for the full collection by Spotify. These transcripts include timing information for each word, automatically detected speaker turns, and an automated estimate of the transcription quality for each utterance. Of these, we used only the transcribed words in our experiments. Additionally, metadata recording the title and description of the podcast and the title and description of the episode was also provided. We did use this additional metadata.

3 Experiments

We submitted five runs for the Ad-hoc Segment Retrieval Task (Task 1). For all the runs, the podcasts were first split into 2 minute segments[3], starting a new segment each minute. The resulting index includes only these pre-processed segments. Standard Indri normalization, which for example lowercases all characters, was used, but no other text normalization was done.

Sequential Dependence:[4] As our baseline system, we used Sequential Dependence (SD) model [2], with stemming, using the concatenation of the topic's title and description fields as the query. The Indri [12]

[3]Except the *unstemmed LM 5 min* setup which is a part of the Combine Re-Rank Combine run.
[4]This run is referred to as Run 2 in the official task results.

query is created as follows:

$$\#\text{weight}(0.8\ \text{q}\ 0.1\ \#2(\text{q})\ 0.1\ \#\text{uw8}(\text{q})) \tag{1}$$

where q represents the concatenation of topic title and description fields. This query represents a weighted combination of

- individual query terms with weight 0.8
- query terms within an ordered window of size 2 (i.e. bigrams) (*#2*) with weight 0.1
- query terms within an unordered window of size 8 (*#uw8*) with weight 0.1

As Table 2 shows, this model, marked as *stemmed LM + SD*, achieves the best NDCG on the Train set among the lexical models that we tried.

SD Re-Ranked by T5:[5] In order to answer *Q1*, we apply a transformer-based model to re-rank the top 1000 results from the SD model. For this purpose, we use the base version of T5 model [11] fine-tuned on the MS MARCO [1] passage retrieval collection. We selected this model because we believed that the passage retrieval task to be similar to the podcast retrieval task in which also the relevant passage needs to be retrieved. We used the publicly released fine-tuned model provided by the University of Waterloo[6]. The query used for the SD stage before re-ranking used the concatenated title and description fields. Following the setup of Nogueira et al. [9], we construct the input sequence to the T5 model as follows:

$$\text{Query: q Document: d Relevant:} \tag{2}$$

Here, q stands for the description field of the topic and d represents the 2 minute text segment of the podcast. Given this input sequence, the fine-tuned model on MS MARCO is trained to produce token '*true*' following the token 'Relevant:' if the document d is relevant to query q's topic otherwise it produces token '*false*'. We use the softmaxed score of the token 'true' as the score for the document d for query q. We follow the same setup for T5 model in the subsequent runs.

Combine Re-Rank Combine:[7] The small size of the Train set results in weak estimates of relative preference among systems, so we elected to try system combination in an effort to create a run that would work well on Test. We combined the following systems:

- Retrieval of the **unstemmed** content using Indri language model (*unstemmed LM*). The query was created by placing all words from the topic title and description fields in the Indri [12] *#combine* operator.
- Retrieval of the **unstemmed** content using Indri language model with Word2vec [7] expansion applied to the topic title (*unstemmed LM + word2vec QE*). The centroid was created from the words used in the topic title and 10 terms closest to this centroid in the embeddings space were concatenated with the topic. The query was created by placing all words from the topic title and the 10 expansion terms in the Indri *#combine* operator.
- Retrieval of the **unstemmed** content using TFIDF relevance model (*unstemmed TFIDF*). Terms from topic title and description fields were used and no Indri operator was used in this case.
- Retrieval of the **unstemmed** content using Indri language model, but with segments of 5 minutes used as the indexed passages (*unstemmed LM 5 min*). As these segments are longer than those required in the task, we first locate the middle point in the segment and then find the closest possible starting point of the segment occurring after it in the recording. This means that the returned starting point is typically exactly in the third minute after the beginning of the 5-minute long passage. The query was created by placing all words from the topic title and description fields in the Indri *#combine* operator.

[5]This run is referred to as Run 1 in the official task results.
[6]`https://huggingface.co/castorini/monot5-base-msmarco`
[7]This run is referred to as Run 3 in the official task results.

- Retrieval of the **stemmed** content using Indri language model with Sequential Dependence Model applied to all words from title and description (*stemmed LM + SD*). The Indri query is created as described in Eqn. 1 using all the words from the topic title and description fields.

- Retrieval of the **stemmed** content using Indri language model with stopwords removal applied (*stemmed weighted LM + stopwords*). Indri *#weight* operator is used with the words from the title having a weight of 1 and the words from the description having a weight of 0.5 .

- Retrieval of the **stemmed** content using Indri language model (*stemmed LM + metadata*). All passages in the collections are expanded with the metadata description of the podcast. The transcribed passage in the text form is concatenated with the the additional information consisting of the show name, show description, publisher, episode name and episode description. The query was created by placing all words from the topic title, and description fields in the Indri *#combine* operator.

Results for these individual systems on the Train set are shown in Table 2. Different systems leverage different evidence, so we expect the combination of these systems to outperform any single system. We balanced the number of stemmed and unstemmed systems with an eye toward improving robustness. We first combine these systems using the reciprocal rank combination from TREC tools [10]. This combination is then re-ranked using two types of transformer models: BERT-Large [4] and T5-Base. Both of these models are publicly released and are fine-tuned on the MS MARCO passage retrieval task.[8][9] To further improve robustness, the T5 model is re-ranked separately using only the description or only the title field of the topic, while the BERT model is re-ranked using the concatenation of title and description topic fields, which was the best performing setup on the Train set. Finally, all the three re-ranked outputs are combined together with the original (unre-ranked) system combination, again using reciprocal rank combination. This process is illustrated in Figure 1.

Lexical Run	No re-ranking	Re-ranked
unstemmed LM	**0.4757**	**0.5462**
unstemmed LM + word2vec QE	0.3836	0.4886
unstemmed TFIDF	0.4640	0.4974
unstemmed LM 5 min	0.4070	0.5298
stemmed LM + SD	0.4692	0.4999
stemmed weighted LM + stopwords	0.4519	0.4734
stemmed LM + metadata	0.4159	0.4976

Table 2: NDCG scores of the lexical approaches used in the Run3 and Run4. The re-ranked scores correspond to the re-ranking approach and setup used in the Run4. The highest scores with and without re-ranking are highlighted.

Re-Rank Combine:[10] In this run, we used the same seven lexical systems as those used in the previous run. However, each of these systems was first re-ranked using one of the two re-ranking models (T5-Base or BERT-Large). Again to increase the diversity, some of the models we re-ranked using only the topic title, some using only the description and some using both. The precise setups were selected based on the performance of different setups with the respect to diversity of the used systems:

- unstemmed LM re-ranked with T5-Base using only description field. The input sequence is constructed as described in Eqn 2 with q as the description field and d as the 2 minute podcast text segment.

[8] `https://huggingface.co/castorini/monot5-base-msmarco`
[9] `https://huggingface.co/castorini/monobert-large-msmarco`
[10] This run is referred to as Run 4 in the official task results.

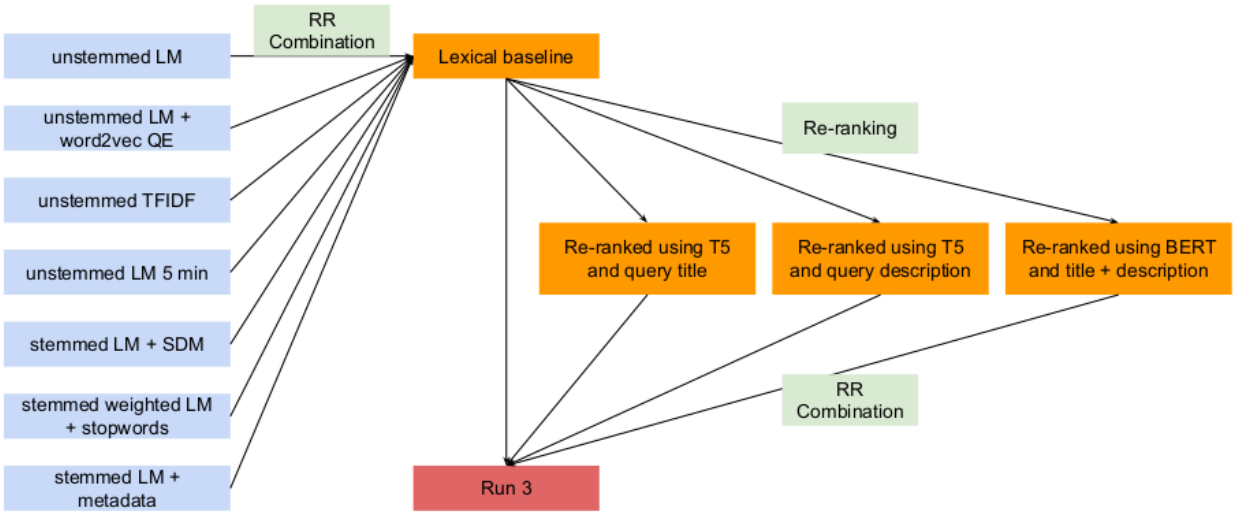

Figure 1: *Combine Re-Rank Combine* system diagram (Run 3).

- unstemmed LM + word2vec QE re-ranked with BERT-Large using only title field. The input sequence to the BERT-Large model is as follows:

$$\text{[CLS] q [SEP] d [SEP]} \tag{3}$$

  where q stands for the title field and d represents the 2 minute podcast text segment. We follow the same setup to construct the input sequence for BERT-Large model in the subsequent systems.

- unstemmed TFIDF model re-ranked with BERT-Large using only description field. The input sequence is constructed as described in Eqn 3 with q as the description field and d as the 2 minute podcast text segment.

- unstemmed LM 5 min re-ranked with BERT-Large using concatenation of title and description fields. The input sequence is constructed as described in Eqn 3 with q as the concatenation of title and description field and d as the 5 minute podcast text segment.

- stemmed LM + SD re-ranked with T5-Base using only description field. The input sequence is constructed as described in Eqn 2 with q as the description field and d as the 2 minute podcast text segment.

- stemmed weighted LM + stopwords re-ranked with BERT-Large using concatenation of title and description fields. The input sequence is constructed as described in Eqn 3 with q as the concatenation of the title and description fields and d as the 2 minute podcast text segment.

- stemmed LM + metadata re-ranked with T5-Base using only title field. The input sequence is constructed as described in Eqn 2 with q as the title field and d as the 2 minute podcast text segment.

Performance of these systems on the training set is displayed in Table 2 and it is possible to directly compare these system with their non-ranked variants. The re-ranked variants are then combined together using the reciprocal ranks-based combination. This system is thus complimentary to the previous Run 3. While in Run 3 we first created a single strong baseline by combining diverse lexical approaches and then we re-ranked that, in this run we reversed the procedure, first applied the re-ranking and then combined the re-ranked runs. Comparison of the Run 3 and Run 4 thus should help us to answer the *Q3*.

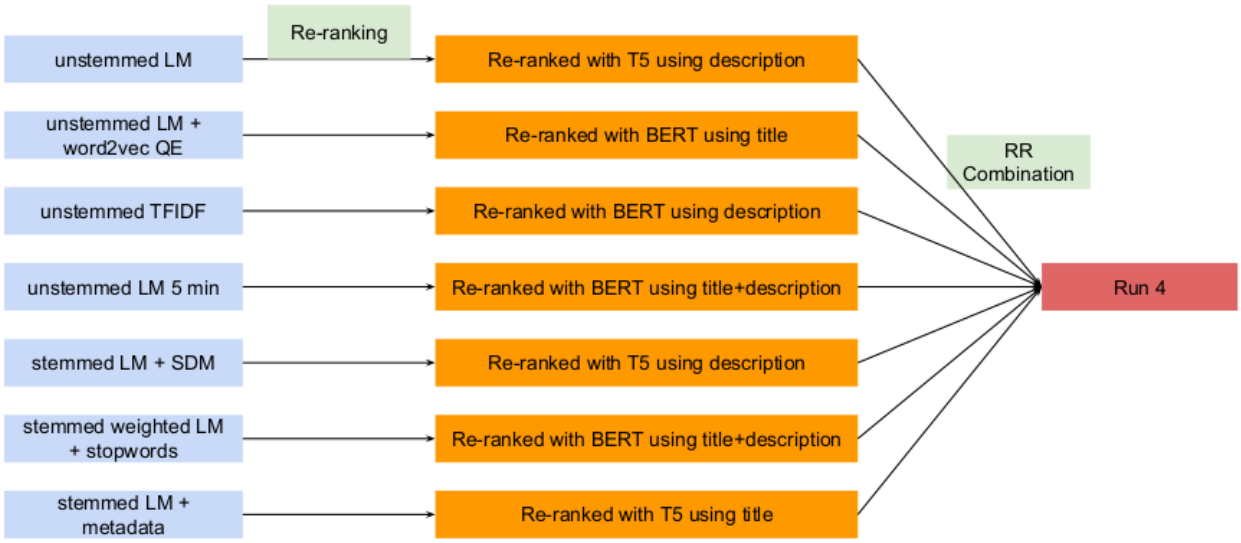

Figure 2: *Re-Rank Combine* system diagram (Run 4).

Three-Stage Combination:[11] Finally, we tried combining all four of our other submitted runs to produce a combination of combinations, one of which itself was a combination of combinations. If these comblex combinations were sufficiently diverse, then this third level of system combination might yield further benefits. Rather than using RR combination, in this case we first normalized the scores produced by each of the four systems using sum-to-one normalization and then combined the results using CombMNZ [13].

4 Results

The official Test results provided by the track organizers are shown in Table 3 together with corresponding results computed locally on the Train set.

Run	Test			Train		
	NDCG	MAP	P10	NDCG	MAP	P10
Sequential Dependence (SD)	0.5271	0.3665	0.5125	0.4657	0.2087	0.3125
SD Re-Ranked by T5	0.6182	0.3524	0.5271	0.5797	0.2617	0.3875
Combine Re-Rank Combine	**0.6682**$^{*\star\bullet}$	**0.4624**$^{*\star\circ\bullet}$	**0.5958**$^{\star\circ}$	**0.6282**	**0.3363**	**0.4500**
Re-Rank Combine	0.6563**	0.4283**	0.5583*	0.5954	0.2932	0.4375
Three-Stage Combination	0.6467**	0.4342**	0.5833*	0.5864	0.3030	0.4250

Table 3: Results for submitted runs. Test results are the official Podcast Track results, Train results are the corresponding scores obtained locally on the Train set. The highest score for each collection and measure are highlighted. The significance test results are calculated using paired t-test at $p < 0.05$, *means that the run is better than SD, *than SD Re-Ranked by T5, $^{\circ}$than Re-Rank Combine or $^{\bullet}$than Three Stage Combination.

[11] This run is referred to as Run 5 in the official task results.

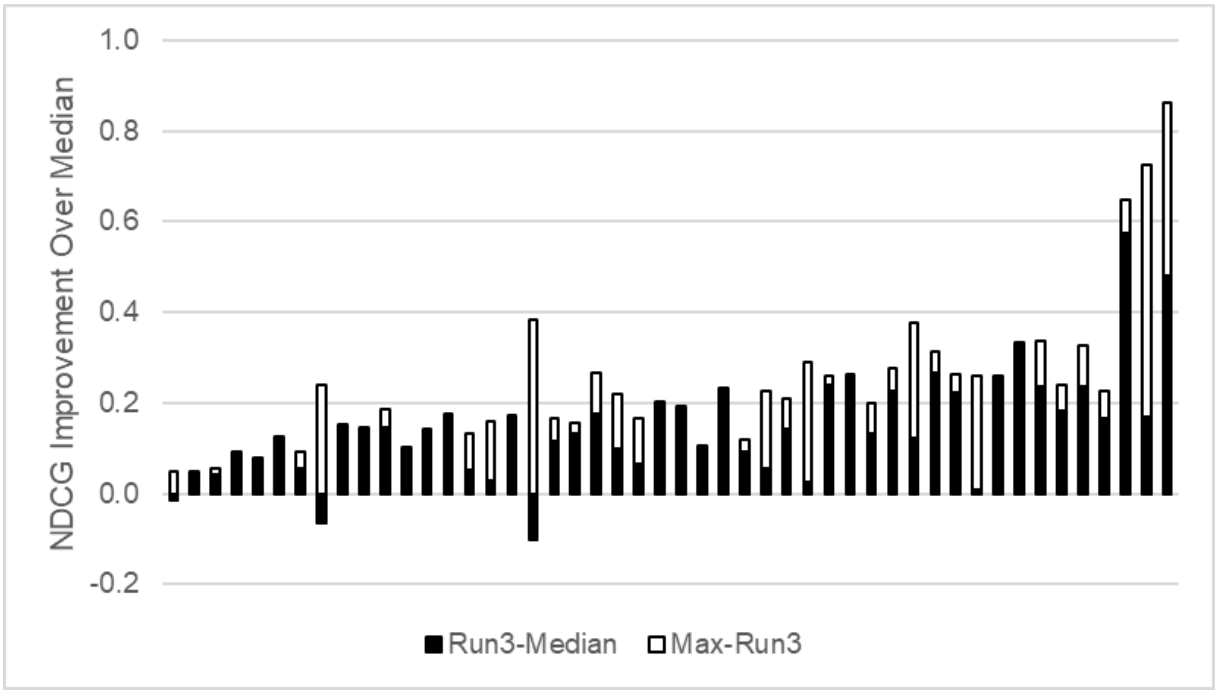

Figure 3: Improvement of Combine Re-Rank Combine run (Run 3) over Median (outline above that shows further improvement of Max over Run 3). Topics are sorted in order of decreasing Median NDCG.

5 Discussion

The results in Table 3 help to answer the research questions stated in the introduction:

- *Q1*) The re-ranking of the results of the lexical retrieval using transformer model had been recently shown to be helpful on a range of tasks [6, 8, 14, 15]. We also see that a neural re-ranking model (re-ranking by T5) runs outperforms the strong lexical SD model by both NDCG and P10. The biggest difference between these runs is for the 'thrift store smell' topic (topic 49) in which the re-ranked run also performs better than the SD run. However, the differences between the runs are not statistically significant. The performance of the re-ranked system is also consistently better on the small Train set, as shown in Table 2.
- *Q2*) All combination models which also use re-ranking (Combine Re-Rank Combine, Re-Rank Combine and Three-Stage Combination) are better than the single re-ranking model (SD Re-Ranked by T5). Differences in the NDCG and MAP scores are statistically significant.
- *Q3*) Our setup in which we first create a strong baseline using a combination of the lexical approaches performs better than the setup in which we first apply the re-ranking on each run and only then combine them. Though these setups are not directly comparable, our results might support the belief that the strong baseline is needed for the re-ranking to perform well.

Surprisingly, the results on the training set and test set exhibit the same trends, even though there are a very small number of topics available in the training set. Also in contrast to our initial expectations, the Three-Stage Combination run did not yield any further improvement over these combinations.

Comparing the NDCG of our best performing Combine Re-Rank Combine run (Run 3) with the median NDCG of all the runs submitted to the task[12], our system performs worse than the median values only for

[12]Minimum, median and maximum NDCG values for each topic were provided by the organizers.

3 topics. We achieved the best reported NDGC results (i.e., the NDCG score of our run is the maximum achieved score over all rund) for 11 topics. As there are no relevance judgements for two topics, this is almost a quarter of the available topics. This trend is visible in the Figure 3, which compares the performance of the Combine Re-Rank Combine and the median and maximal achieved per-topic scores. The scores in the figure are sorted according to the median scores, with the 'easier' (higher-median) topics on the left and the 'harder' (lower-median) topics on the right. The improvement of our best run over the median seems to increase somewhat with topic difficulty, which makes sense since these are absolute improvements over the median, and larger absolute improvements are possible for topics with lower median NDCG values. Where our run achieved the maximum reported NDCG, that happened more often on easier and mid-range topics than on hard topics. The biggest improvement over the median was achieved on the 'fyre festival' topic (topic 42). The biggest shortfall of our system when compared to the maximum occurred for the 'horoscope reading cancer' topic (topic 31). Our largest underperformance of the median was for the 'missouri quilt mom' topic (topic 46). For each of our runs, the NDCG on topical topics is much better than our NDCG for the other two categories (refinding and known item topics); those differences are smallest, however, for our Combine Re-Rank Combine run. In the case of the refinding and known item topics we also see larger differences between our best tun and the maximum NDCG for that topic. We thus plan to further explore specifically refinding and known item topics in our following work.

6 Conclusion

In this first year of the TREC Podcasts Track our focus was on achieving robust results using system combination and neural re-ranking. Our results are promising, achieving a statistically significant 27% relative improvement in NDCG and a 16% relative improvement in precision at 10 documents over a strong lexical matching baseline built using Indri language model and Sequential Dependence model. We look forward to the discussions at TREC, and to the second year of the track!

Acknowledgements

This research has been supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of 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 therein.

References

[1] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, et al. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268v3*, 2018.

[2] M. Bendersky, D. Metzler, and W. B. Croft. Learning concept importance using a weighted dependence model. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, New York, NY, USA, 2010.

[3] A. Clifton, S. Reddy, Y. Yu, A. Pappu, R. Rezapour, H. Bonab, M. Eskevich, G. J. F. Jones, J. Karlgren, B. Carterette, and R. Jones. 100,000 Podcasts: A Spoken English Document Corpus. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, Barcelona, Spain, 2020.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] R. Jones, B. Carterette, A. Clifton, M. Eskevich, G. J. F. Jones, J. Karlgren, A. Pappu, S. Reddy, and Y. Yu. TREC 2020 Podcasts Track Overview. In *The 29th Text Retrieval Conference (TREC) notebook*. NIST, Gaithersburg, MD, USA, 2020.

[6] J. Lin. The neural hype, justified! A recantation. In *ACM SIGIR Forum*, volume 53, 2019.

[7] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*. 2013.

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

[9] R. Nogueira, Z. Jiang, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*, 2020.

[10] J. Palotti, H. Scells, and G. Zuccon. TrecTools: an open-source Python library for Information Retrieval practitioners involved in TREC-like campaigns. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 2020.

[12] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, volume 2, McLean, VA, USA, 2005.

[13] S. Wu. *Data Fusion in Information Retrieval*. Springer Publishing Company, Incorporated, 2012.

[14] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, Minnesota, 2019.

[15] W. Yang, H. Zhang, and J. Lin. Simple applications of BERT for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*, 2019.