Noise-Reduction for Automatically Transferred Relevance Judgments

Maik Fröbe,[1] Christopher Akiki,[2] Martin Potthast,[2] Matthias Hagen[1]

[1] Martin-Luther-Universität Halle-Wittenberg
[2] Leipzig University

Abstract The TREC Deep Learning tracks used MS MARCO Version 1 as their official training data until 2020 and switched to Version 2 in 2021. For Version 2, all previously judged documents were re-crawled. Interestingly, in the track's 2021 edition, models trained on the new data were less effective than models trained on the old data. To investigate this phenomenon, we compare the predicted relevance probabilities of monoT5 for the two versions of the judged documents and find substantial differences. A further manual inspection reveals major content changes for some documents (e.g., the new version being off-topic). To analyze whether these changes may have contributed to the observed effectiveness drop, we conduct experiments with different document version selection strategies. Our results show that training a retrieval model on the "wrong" version can reduce the nDCG@10 by up to 75%.

Keywords: MS MARCO · monoT5 · Relevance transfer.

1 Introduction

Retrieval models are usually trained and evaluated either (1) on datasets with up to several thousands of relevance judgments, carefully curated by expert annotators (e.g., for TREC tracks), or (2) on datasets with hundreds of thousands or more judgments inferred from user data. Particularly transformer-based retrieval models require many training instances to outperform traditional sparse models like BM25 [22]. One of the first datasets with a sufficiently large number of judgments was MS MARCO [7,23]. It was originally released with one positive passage-level judgment for each of 532,761 queries (i.e., only relevant instances are annotated) and later complemented by document-level judgments for Version 1. Despite some erroneous judgments [6,8,9], MS MARCO Version 1 has been the basis of training highly effective document retrieval models (e.g., monoT5 [24] which is the current state of the art[3] on Robust04). However, in 2021, the TREC Deep Learning tracks transitioned from using MS MARCO Version 1 as official training data to MS MARCO Version 2, a larger and improved version. It therefore came as a surprise that models trained on Version 2 were found to be less effective than models trained on Version 1 [9].

[3] https://paperswithcode.com/sota/ad-hoc-information-retrieval-on-trec-robust04

Table 1. Examples of differences between versions of positive training instances from MS MARCO Version 1 (crawled in 2018) and Version 2 (2021). Text fragments highlighted in blue italics indicate relevance (erroneous versions have no blue italics).

Query	**Relevant Document**		**Comment**
	Version 1 (2018)	Version 2 (2021)	
what are deposit solutions banking	Oops! There was a problem! We had an unexpected problem processing your request.	*Deposit Solutions* Crunchbase *Company Profile* ...	Crawling error in V1
what are yellow roses mean	Meaning Of A Yellow Rose ... a yellow rose *stands for joy and happiness* ...	20 Best Knockout Roses To Make Your Garden Outstanding	Redirect in V2
how much magnesium in kidney beans	Kidney Beans ... *a cup of kidney beans contains 70 mg of magnesium* ...	Magnesium Grocery List. Bring this list to the store to ...	Content change in V2

The document-level relevance judgments for MS MARCO Version 1 were transferred 1:1 from the originally crowdsourced passage-level relevance judgments [8]. The transfer was based on a URL match, assuming that a document having the same URL as one that previously contained a relevant passage included in the original MS MARCO passage dataset is still relevant for the same query. The document-level judgment transfer from Version 1 to Version 2 relied on the same heuristic. However, the MS MARCO documents were crawled one year (Version 1) and four years (Version 2) after the original passage-level relevance judgments were obtained. Thus, some of the documents' content may have changed—possibly invalidating the passage-level judgments. A preliminary analysis of a sample of 50 instances showed that Version 2 has a comparable error rate to Version 1 [9]—whereas related work on a different web crawl found that re-crawling web pages after 3 years can yield quite substantial changes [15].

To analyze why retrieval models trained on MS MARCO Version 2 were less effective in the TREC 2021 Deep Learning track than models trained on Version 1, we compare the two versions of all 325,183 positive training instances using monoT5's estimated probability of a document being relevant to the respective query. Some cases with a substantially different probability are shown in Table 1. Overall, Version 1 contains about 3,800 such potential errors but Version 2 has about 13,100 (details in Sections 3–5). Interestingly, snapshots from the Wayback Machine with an archival date closer to the date of the MS MARCO passage judgments only yield few additional cases. Finally, we compare the effectiveness of monoT5 models trained on the erroneous versions to models trained on the "correct" counterparts and observe that training on errors can reduce the nDCG@10 by up to 75% (Section 6). Our code and data are freely available.[4]

[4] https://github.com/webis-de/CLEF-22

2 Related Work

The passage-level MS MARCO relevance judgments [8] (only positive instances included) enabled the training of data-hungry transformer-based retrieval models [22] and triggered research on identifying / sampling negative training instances [16,29,37,38]. For the two document-level MS MARCO versions created about one or four years later [8,6,9], the passage-level judgments were 1:1 transferred to the documents crawled for the same URL. Still, content changes may actually have invalidated some of the transferred judgments in the training data.

Evolution of Web Pages. Even though web pages change regularly, content and links usually remain highly similar within a couple of weeks [5,12,13,26,27]. But when more time has passed, two snapshots of a page can differ a lot. For example, a study by Fröbe et al. [15] showed that about 90% of the ClueWeb09 documents judged for some topic from the TREC Web tracks had a substantially different content in the ClueWeb12 crawled three years later—invalidating any URL-based judgment transfer. For the ClueWeb corpora, actually no judgments were transferred but a similar effect might have impacted the transition from MS MARCO Version 1 to 2. Besides the actual two MS MARCO document versions, we also study Wayback Machine snapshots that are close to the potential period of the MS MARCO passage-level judgments—inspired by recent studies that successfully enriched their datasets via the Wayback Machine [15,19].

Handling Training Data Errors. The two standard approaches to deal with errors in the training data of learning-to-rank algorithms [34] are (1) robust loss functions and (2) sample selection. While modifications of popular loss functions like adaptions of the cross-entropy loss [11] or generalizations of PeerLoss [34] aim to make them "robust" on noisy click data, sample selection aims to remove erroneous training instances [14]. Sample selection has been successfully applied to click logs [4,32] but also to MS MARCO [29,31]. For instance, Qu et al. [29] and Arabzadeh et al. [1] observed that unjudged MS MARCO passages (implicitly assumed to be non-relevant) can be more relevant to a query than the actually annotated positive instance. Taking this observation into account when sampling negative training instances, Qu et al. substantially increased the effectiveness of their final model [29]. Also Rudra et al. [31] applied sample selection and used only the most relevant passage of a relevant document as a positive instance during training. They assumed that the passage with the highest monoBERT score [25] is the most relevant to a query and removed the other passages of a positive document. We expand this idea to compare multiple versions of a document but use monoT5 [24] since it is more effective than monoBERT [36].

3 Identifying Potential Errors in the Training Instances

To study potential judgment "errors" in the MS MARCO document retrieval training data caused by the different crawling dates and possible content changes,

we use monoT5 [24] to estimate the probability of a positive training document being relevant for the respective query. For each positive training instance, we compare the probabilities of the variants in Version 1 and Version 2 to identify discrepancies that may hint at errors on either side. We also use snapshots from the Wayback Machine to assess whether document versions close to the time of the MS MARCO passage-level judgments could "repair" some errors.

Overview of MS MARCO. Version 1 of the MS MARCO document-level dataset was crawled in 2018 and consists of 3,213,835 documents and 384,597 positive training instances (i.e., query–document pairs) released for the document retrieval tasks at the TREC 2019 and 2020 Deep Learning tracks [6,8]. Documents have a URL, a title field, and a body (HTML tags and boilerplate such as navigation elements removed by a proprietary approach [9]). The positive training instances were created by transferring the passage-level judgments obtained about one year earlier to the documents crawled for the same URL [8]. To somewhat assess the noise introduced by the different crawling date, Craswell et al. [9] used the NIST assessors' judgments on the 46 test queries of the 2020 track and found that for 11 of the 46 queries (i.e., 24%) the positive training instances were assessed as non-relevant—possibly hinting at crawling differences.

Version 2 of the MS MARCO document-level dataset was crawled in 2021 and consists of 11,959,635 documents and 331,956 positive training instances (i.e., query–document pairs) for the TREC 2021 Deep Learning track [9]. Documents now have a URL, title, body, and headings. The document pre-processing (i.e., identifying the body and headings) was different to Version 1, though. A proprietary query-independent approach identified the best non-overlapping passages in a document and concatenated them (mappings between the passage dataset and the document dataset were provided). The training instances were again created by transferring them on basis of the URLs.

Wayback Machine Snapshots. We use the Wayback Machine to compare the MS MARCO document versions with snapshots closer to the time of the passage-level judgments. For each training instance, we try to find one valid snapshot (i.e., successfully crawled with status code 200) from 2015, 2016, and 2017 using the Memento API.[5] If multiple snapshots are available for a year, we select the one closest to July 2nd since this day is the "middle" of the calendar year. We use the Resiliparse library [2] of ChatNoir [3] to extract the plain text and main content of the Wayback documents (this approach produces slightly different main content than the proprietary MS MARCO one, but we still deem the results as "good enough"). Overall, we found snapshots for 68,384 MS MARCO training instances (41,269 have a snapshot for all three years).

Preprocessing Steps. Since the monoT5 model that we use to identify potential errors cannot handle arbitrary input lengths, long documents need to be split into passages that are scored individually [24]. Since there is no explicit mapping to

[5] https://archive.readme.io/docs/memento

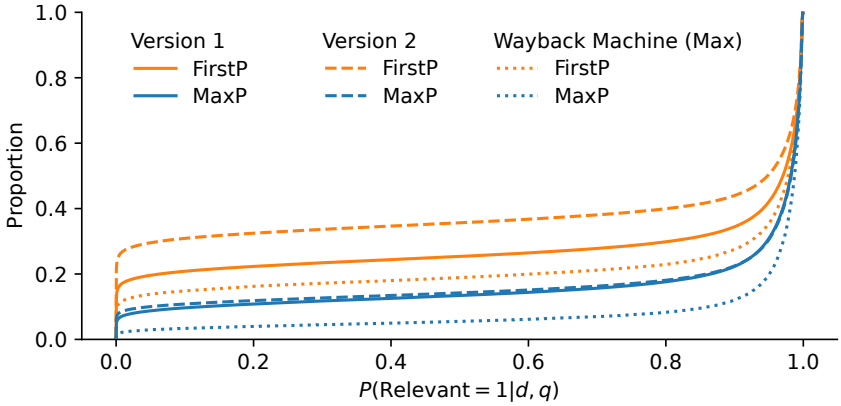

Figure 1. Cumulative distribution of the monoT5 relevance probability estimates for the positive training instances in MS MARCO Version 1 and 2, and the "best" Wayback Machine snapshot using the first or highest scoring passage (FirstP or MaxP).

passages for Version 1 documents and our snapshots from the Wayback Machine, we use the TREC CAsT tools[6] to split all document versions into passages with the same pipeline. Following suggestions of Dai and Callan [10], we concatenate a document's title and body and split documents at the sentence level into fixed-length passages of approximately 250 terms—fixed-length passages were previously reported to be superior to variable-length passages [17].

Relevance Estimation with monoT5. We use the PyGaggle[7] implementation of monoT5 with its most effective pre-trained variant[8] to estimate the relevance of a document to a query. MonoT5 is based on the sequence-to-sequence model T5 [30] and ranks documents by the probability that, given the query and the document, the decoders' output is the literal "true" [28]:

$$P(\text{Relevant} = 1|d, q)$$

which estimates the probability that a document d is relevant to a query q. We apply monoT5 to all passages of a document and use two approaches that showed high effectiveness in previous work [24,31,35] to aggregate the passage level scores to document level scores: (1) FirstP where the probability of a document being relevant to a query is approximated by monoT5's prediction for the first passage, and (2) MaxP where the probability of a document being relevant to a query is approximated by monoT5's maximum prediction for any of its passages.

Results. Figure 1 shows the cumulative distribution of the monoT5 scores (i.e., the probabilities that a document is relevant to a query, as determined by

[6] https://github.com/grill-lab/trec-cast-tools
[7] https://github.com/castorini/pygaggle
[8] https://huggingface.co/castorini/monot5-3b-msmarco

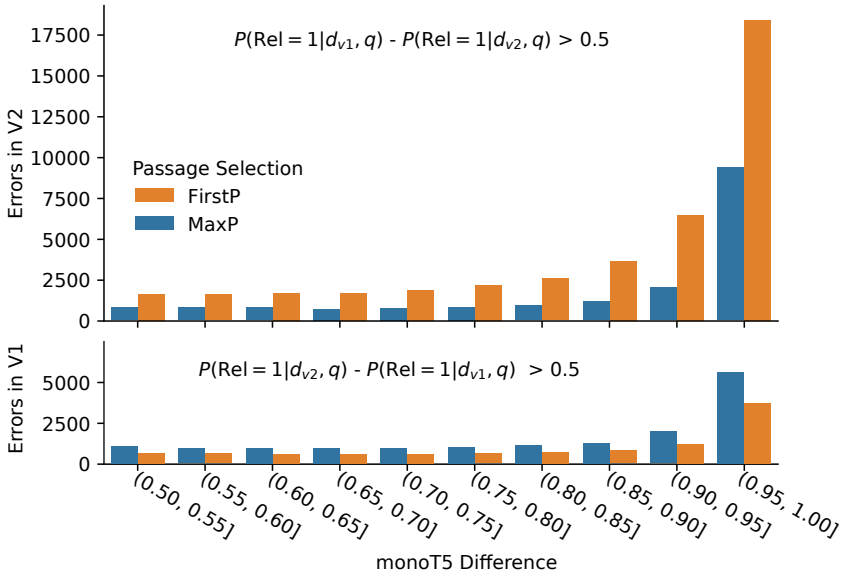

Figure 2. Distribution of the monoT5 difference of error candidates in Version 2 (upper plot) and Version 1 (bottom plot). We report the difference in the monoT5 probabilities of the two positive training instances for all pairs with high discrepancies above 0.5 for two passage selection strategies (FirstP and MaxP).

monoT5) among the positive training instances for Version 1, Version 2, and snapshots from the Wayback Machine from 2015 to 2017 for FirstP and MaxP aggregation. The cumulative distributions for both Version 1 and 2 include all training instances, whereas we only retain those 41,269 instances into the Wayback Machine snapshots that were successfully crawled in all three years. For them, we select the maximum score of the three candidates as upper bound. Given only correct positive training instances (that are all relevant to its query), an ideal monoT5 model would assign probabilities of 1 to all of them. However, we observe that monoT5 predicts that a non-negligible number of documents is not relevant for all three corpora. Version 2 has the highest proportion of such potential errors (30.59% of positive instances have a probability below 10% for FirstP, respectively 10.52% for MaxP) while our upper bound using snapshots from the Wayback Machine has the smallest proportion (14.54% of positive instances have a probability below 10% for FirstP, respectively 3.07% for MaxP) showing that selecting a positive instance out of multiple versions is promising.

4 Training Datasets with Potential Errors

To assess the reliability of monoT5's relevance probabilities, we construct five datasets with potential errors by comparing the probabilities across different versions of the documents. Figure 2 shows the distribution of positive training

instances where one version is predicted to be substantially more likely to be relevant while its counterpart is not (differences >0.5), indicating errors. Our five datasets cover cases of interest from this analysis:

- *V1≫ V2 (FirstP).* This dataset contains the 41,587 qrels where monoT5 predicts the first passage of the document in Version 1 to be substantially more relevant than the first passage of the recrawled document in Version 2 (probability of Version 1 minus probability of Version 2 is greater than 0.5). The orange boxes in the upper plot of Figure 2 show the distribution of probability differences. For 18,382 instances, the probability that the document in Version 1 is relevant is by 0.95 larger than the probability that the counterpart in Version 2 is relevant.
- *V1≫ V2 (MaxP).* This dataset contains the 17,969 qrels where monoT5 predicts the highest-scoring passage of the document in Version 1 to be substantially more relevant than that of the recrawled document in Version 2 (probability of Version 1 minus the probability of Version 2 is greater than 0.5; the blue boxes in the upper plot of Figure 2).
- *V2≫ V1 (FirstP).* This dataset contains the 9,991 qrels where monoT5 predicts the first passage of the document in Version 2 to be substantially more relevant than that of the recrawled document in Version 2 (the orange boxes in the lower plot of Figure 2).
- *V2≫ V1 (MaxP).* This dataset contains the 15,817 qrels where monoT5 predicts the highest-scoring passage of the document in Version 2 to be substantially more relevant than that of the recrawled document in Version 2 (the blue boxes in the lower plot of Figure 2).
- *Wayback Machine.* This dataset contains the 41,269 qrels where 5 versions of the positive documents are available: (1) Version 1, (2) Version 2, and (3) three snapshots from the Wayback Machine for 2015, 2016, and 2017. While the above datasets allow the assessment of the impact of errors, this one is used to assess if multiple versions are helpful on parts of MS MARCO without many errors.

Discussion. Our five datasets cover different parts of MS MARCO and are not representative of the complete corpus because they are intentionally focused on subsets of the training data that may contain many errors. Table 2 provides an overview of the datasets showing that all of them are rather dissimilar (the highest Jaccard Similarity with respect to included query IDs is 0.34 between the "V2 ≫ V1 (FirstP)" and the "V2 ≫ V1 (MaxP)" datasets). The first four have much fewer documents from Wikipedia and longer URLs with more parameters compared to "all" documents from MS MARCO. URL parameters are indicative of dynamic content and thus varying relevance, while Wikipedia articles remain topic-stable and thus relevant.

5 Review of Potential Errors in Positive Instances

We manually review 600 queries with their corresponding positive documents in Version 1 and 2 of MS MARCO, 100 each from our five datasets and a

Table 2. Overview of characteristics of positive documents in our constructed training datasets using FirstP (F) respectively MaxP (M) aggregation. We report statistics on the URL, the most similar dataset measured as Jaccard similarity on the query IDs, and the most frequent domain.

Query Set	Relevant URLs		Most Similar		Most Frequent Domain	
	Len(Path)	Parameters	Name	Sim.	Domain	Percentage
V1 ≫ V2 (F)	38.04	0.06	V1 ≫ V2 (M)	0.31	wikipedia.org	8.06
V1 ≫ V2 (M)	36.90	0.07	V1 ≫ V2 (F)	0.31	wikipedia.org	4.37
V2 ≫ V1 (F)	33.57	0.04	V2 ≫ V1 (M)	0.34	wikipedia.org	9.01
V2 ≫ V1 (M)	35.28	0.04	V2 ≫ V1 (F)	0.34	wikipedia.org	9.09
Wayback M.	32.25	0.02	V1 ≫ V2 (F)	0.07	wikipedia.org	22.72
All	35.11	0.05	–	–	wikipedia.org	15.86

random subset of 100 from all MS MARCO training queries. One annotator labeled the 600 instances in random order, and the annotations were randomly checked by two of the co-authors. For each instance, the annotator saw the query, the document in Version 1, and the document in Version 2 and labeled which of the two documents is more relevant to the query (if any). Table 3 provides an overview of the annotations for the five datasets and the random subset. Most of the document pairs of the random subset are equally relevant to the query (79 of the 100 labeled pairs have labels V1=V2=1), but still, there are some errors (e.g., for 9 document pairs, Version 1 was relevant but Version 2 non-relevant, i.e., V1=1>V2=0). The subsets that we constructed so that they contain many errors have, as expected, many errors (e.g., 73 of the labeled pairs from "V1 ≫ V2 (MaxP)" are indeed not relevant in Version 2 but in Version 1, i.e., V1=1>V2=0). To estimate the overall number of errors in both versions, we use the MaxP variants (as FirstP has cases where the relevant passage comes later in the document) and find that the precision of monoT5 differs substantially for the two versions. Errors in Version 2 are detected with a precision of 0.73 (for 73 of the 100 reviewed "V1 ≫ V2 (MaxP)" pairs, the document in Version 1 is relevant but not relevant in Version 2, i.e., V1=1>V2=0). Errors in Version 1 only have a precision of 0.25 (for 25 of the 100 reviewed "V2 ≫ V1 (MaxP)" pairs, the document in Version 2 is relevant but not relevant in Version 1, i.e., V2=1>V1=0), resulting in a precision-oriented estimation that Version 2 has 13,117 errors while Version 1 has 3,954 errors.

6 Experiments

We fine-tune monoT5 on each of our datasets to assess if the potential errors identified in MS MARCO negatively affect their effectiveness. We therefore evaluate the effectiveness of these models on three benchmarks: (1) the 100 TREC Web track topics of the ClueWeb12, (2) the 88 topics of the document retrieval

Table 3. Overview of our manual review of the relevance of positive documents for our datasets. For each dataset, we labeled 100 document pairs and report the absolute number of relevance preferences (e.g., V1=V2=1: both relevant, V1=1,V2=0: V1 relevant, V2 non-relevant, etc.) and the precision and the estimated number of errors.

Query Set		**Document Relevance**							**Prec.**	**Labels**	**Errors**
Type	Selection	V1	0	0	1	1	2	1			
		V2	0	1	0	1	1	2			
V1 ≫ V2	FirstP		1	4	48	37	6	4	0.48	41,587	19,962
	MaxP		5	3	73	11	7	1	0.73	17,969	13,117
V2 ≫ V1	FirstP		0	21	7	55	2	15	0.21	9,991	2,098
	MaxP		0	25	5	51	0	19	0.25	15,817	3,954
Random	—		4	0	9	79	3	5	—	325,183	—
Wayback M.	—		0	1	3	89	7	0	—	41,269	—

task of the TREC Deep Learning track from 2019 and 2020 (Voorhees et al. [33] recommend not to reuse the 2021 edition), and (3) all 250 topics of Robust04. Each training dataset has multiple versions of the positive document and we compare strategies to select the "best" version to demonstrate how the different versions impact effectiveness.

Trained Models. We conduct our experiments with the PyGaggle[9] implementation [18] of monoT5 as this model shows state-of-the-art effectiveness in a range of retrieval experiments [36]. Following Nogueira et al. [24], we use the base version of monoT5 and fine-tune it for one epoch on 10,000 randomly selected positive training instances from one of our five datasets, plus 10,000 randomly selected negative instances from the top-100 BM25 results on MS MARCO. This is repeated ten times using ten different seeds, thus obtaining ten fine-tuned monoT5 models per dataset. Independently of the passage aggregation strategy (FirstP or MaxP) used for the ground truth labels of each of our five datasets, five of the ten models per dataset use FirstP aggregation during training, and five use MaxP aggregation.

Using ir_datasets [20] for data-wrangling,[10] we follow previously suggested training regimes [24,25,35], and pass relevant and non-relevant instances in alternating order within the same batch to a model during training. During inference, we rerank the top-100 BM25 results of PyTerrier [21] (default configuration) using the same passage aggregation used during training a given model.

Effectiveness of MonoT5 Trained on Erroneous Positive Instances. In our first experiments, we finetune monoT5 models on the four datasets which, according to the probabilities of the pretrained monoT5 model, contain errors in the positive training instances in one version of MS MARCO while the other version

[9] https://github.com/castorini/pygaggle
[10] https://github.com/allenai/ir_datasets

Table 4. Effectiveness of monoT5-base models trained on 20,000 instances from our constructed datasets. Positive instances are selected with one of three selection strategies: (1) BM25, (2) $T5_{Min}$, and (3) $T5_{Max}$. We report Precision@10 and nDCG@10 on the ClueWeb12 (2013 and 2014), the TREC Deep Learning document retrieval task (2019 and 2020), and Robust04 (all topics). Highest nDCG@10 in bold; † marks statistically significant differences to $T5_{Min}$ at $p = 0.05$, with Bonferroni correction.

Training Data		**ClueWeb12**		**DL 19/20**		**Robust04**	
Queries	Selection	P@10	nDCG@10	P@10	nDCG@10	P@10	nDCG@10
V1 ≫ V2 (FirstP)	BM25	0.517†	0.358†	0.580†	0.512†	0.359†	0.376†
	V1=$T5_{Max}$	**0.551**†	**0.385**†	**0.649**†	**0.586**†	**0.441**†	**0.448**†
	V2=$T5_{Min}$	0.425	0.282	0.450	0.388	0.294	0.297
V1 ≫ V2 (MaxP)	BM25	0.508†	0.352†	0.542†	0.474†	0.377†	0.380†
	V1=$T5_{Max}$	**0.557**†	**0.387**†	**0.620**†	**0.562**†	**0.436**†	**0.446**†
	V2=$T5_{Min}$	0.307	0.177	0.197	0.142	0.211	0.209
V2 ≫ V1 (FirstP)	BM25	0.455	0.308	0.547	0.466	0.384	0.383
	V1=$T5_{Min}$	0.468	0.314	0.534	0.452	0.345	0.349
	V2=$T5_{Max}$	**0.499**	**0.333**	**0.559**	**0.505**†	**0.386**†	**0.385**†
V2 ≫ V1 (MaxP)	BM25	0.422†	0.278†	0.449†	0.394†	0.324†	0.319†
	V1=$T5_{Min}$	0.367	0.238	0.385	0.316	0.287	0.279
	V2=$T5_{Max}$	**0.482**†	**0.318**†	**0.530**†	**0.476**†	**0.361**†	**0.367**†
Random	BM25	**0.546**	**0.371**	0.586	0.538	0.400†	0.404†
	$T5_{Min}$	0.532	0.369	0.591	0.531	0.376	0.384
	$T5_{Max}$	0.544	0.368	**0.616**	**0.570**†	**0.410**†	**0.412**†
BM25 (Baseline)		0.439	0.298	0.563	0.507	0.438	0.449

is correct: "V1 ≫ V2" selected by FirstP or MaxP, and "V2 ≫ V1" selected by FirstP or MaxP. We compare these datasets with random training queries.

In the datasets, two versions of each positive document (Version 1 and Version 2) are found. We compare three selection strategies to select which of the two is used for finetuning: (1) $T5_{Min}$ as baseline, which selects the document with the lower pretrained monoT5 score, (2) BM25, which selects the document with the higher BM25 score, and (3) $T5_{Max}$, which selects the document with the higher pretrained monoT5 score. It turns out that $T5_{Min}$ respectively $T5_{Max}$ almost unanimously select the document from Version 1 respectively Version 2 of MS MARCO, e.g., for "V1 ≫ V2", $T5_{Max}$ always selects Version 1, and consequently $T5_{Min}$ always selects Version 2.

Table 4 shows the effectiveness measured as Precision@10 and nDCG@10 for each combination of finetuning dataset, version selection strategy, and the three benchmarks ClueWeb12, TREC Deep Learning tracks 2019/2020, and Robust04. Each score reported results from applying each of the ten fine-tuned monoT5 models available for a dataset on a given benchmark to obtain ten runs, and

Table 5. Precision@10 and nDCG@10 on three corpora for monoT5-base trained on 20,000 instances from the Wayback Machine data with 5 selection strategies: (1) BM25, (3) $T5_{Max}$, (2) $T5_{Min}$, (4) Version 1, and (5) Version 2. Highest nDCG@10 in bold; † marks statistical significance at $p = 0.05$ to $T5_{Min}$ with Bonferroni correction.

Training Data		**ClueWeb12**		**DL 19/20**		**Robust04**	
Queries	Selection	P@10	nDCG@10	P@10	nDCG@10	P@10	nDCG@10
Wayback M.	BM25	0.534	0.393	0.574	0.509	0.365	0.373
	$T5_{Max}$	0.543	**0.397**	0.620†	0.557†	0.396†	0.403†
	$T5_{Min}$	0.523	0.371	0.585	0.509	0.355	0.361
	V1	**0.562**	0.388	**0.641**†	**0.597**†	**0.439**†	**0.445**†
	V2	0.518	0.359	0.542†	0.472†	0.307†	0.316†
BM25 (Baseline)		0.439	0.298	0.563	0.507	0.438	0.449

then applying five-fold cross-validation over the benchmark's topics using the ten runs, as implemented by PyTerrier [21].

Many erroneous positive training instances can have a very dramatic impact on the effectiveness of ranking models. For the two "V1 $\gg$ V2" training datasets for which our monoT5 heuristic predicted that the positive document in Version 2 is not relevant while the positive document in Version 1 is relevant, we observe that BM25 and $T5_{Max}$ selection outperform the $T5_{Min}$ baseline statistically significant on all three benchmarks. The model trained on the positive instance selected with $T5_{Max}$ achieves an nDCG@10 of 0.562 on the TREC Deep Learning document retrieval task, while the model trained on positive instances selected with $T5_{Min}$ achieve only an nDCG@10 of 0.142. This behavior on the two "V1 $\gg$ V2" training data sets supports our manual review (cf. Section 5) that there is a substantial portion of positive training documents that were relevant to its query in Version 1 (selected by the $T5_{Max}$ strategy), which became non-relevant in Version 2 (selected by the $T5_{Min}$ strategy). Interestingly, many such cases can already be resolved by just using the version of the document with the higher BM25 score. Table 4 shows that training on erroneous positive instances from Version 2 of MS MARCO is very ineffective and that this effect is larger for the "V1 $\gg$ V2 MaxP" dataset than it is for the "V1 $\gg$ V2 FirstP" dataset. This is consistent with our manual review in Section 5, where the MaxP variant identified more errors in positive instances. Also the opposite direction, where the positive instance in Version 1 is not relevant to its query but the version of the document in Version 2 is more relevant can be confirmed by the effectiveness of models trained on the two "V2 $\gg$ V1" datasets: Selecting always the document from Version 2 for training achieves the most effective models, however, these effects are only significant for "V2 $\gg$ V1 (MaxP)", which is again consistent with our manual review from Section 5. The results for the random training queries show that selecting the better positive document out of Version 1 and Version 2 for training also increases model effectiveness, but only slightly because the random selection is less prone to noise compared to our other training datasets.

Using Snapshots from the Wayback Machine. To complement our experiments, we assess whether more versions of positive instances covering a wider time period may improve the effectiveness of finetuned models. We use our Wayback Machine dataset with 41,269 qrels having five versions of each positive document (Version 1, Version 2, 2015, 2016, and 2017, extracted from the Wayback Machine; cf. Section 4). We apply the same training procedure as above. We compare 5 strategies to select the positive instance out of the 5 versions of the positive document: (1) always using Version 1, (2) always using Version 2, and (3) $T5_{Min}$, (4) $T5_{Max}$, and (5) BM25.

Table 5 shows the effectiveness of monoT5 models trained on the Wayback Machine dataset for the five selection strategies on the three benchmarks. The overall picture is similar to the previous experiments: selecting the positive document with $T5_{Max}$ yields more effective models than the BM25 selection which is, in turn, again more effective than the $T5_{Min}$ selection. Interestingly, selecting always Version 1 is even more effective than $T5_{Max}$ and selecting always Version 2 is less effective than the $T5_{Min}$ strategy. The fact that the $T5_{Max}$ and $T5_{Min}$ selection strategies do not produce the most (respectively least) effective models shows that monoT5's probabilities are not suitable to distinguish among mostly correct positive documents and erroneous ones. The Wayback Machine dataset in Table 3 shows that only four out of 100 reviewed queries had incorrect positive documents, likely because "stable" domains like Wikipedia are overrepresented in the Wayback Machine dataset, as shown in Table 2. Hence, only substantial differences in the monoT5 relevance probabilities between versions are reliable. Switching to versions with a slightly higher monoT5 relevance probability does not improve the effectiveness of trained models.

7 Conclusion

Inspired by the effectiveness drop observed in the TREC 2021 Deep Learning track for models trained on MS MARCO Version 2 instead of Version 1, we have compared monoT5's estimated probabilities of judged documents being relevant for their queries in the two versions. Since the judgments were simply transferred after re-crawling documents for Version 2, larger differences in the probabilities might hint at major content changes. Our precision-oriented estimation predicts 13,100 such problems in Version 2—and only 3,800 in Version 1. In experiments, we show that models trained on the "wrong" document versions are highly ineffective. These cases thus probably contribute to the observed effectiveness drop.

Interesting directions for future work include a further investigation of other factors that may influence a model's effectiveness, such as the different preprocessing pipelines used for Versions 1 and 2, or the fact that Version 2 is larger than Version 1 (but same number of judgments). In addition, a more fine-grained classification of possible content changes might help to identify issues that can be neglected and issues that should be fixed during training dataset creation.

References

1. Arabzadeh, N., Vtyurina, A., Yan, X., Clarke, C.: Shallow Pooling for Sparse Labels. CoRR **abs/2109.00062** (2021)
2. Bevendorff, J., Potthast, M., Stein, B.: FastWARC: Optimizing Large-Scale Web Archive Analytics. In: Proc. of OSSYM 2021. OSF (2021)
3. Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: Elastic ChatNoir: Search Engine for the ClueWeb and the Common Crawl. In: Proc. of ECIR 2018. Springer (2018)
4. Cen, R., Liu, Y., Zhang, M., Zhou, B., Ru, L., Ma, S.: Exploring Relevance for Clicks. In: Proc. of CIKM 2009. pp. 1847–1850. ACM (2009)
5. Cho, J., Garcia-Molina, H.: The Evolution of the Web and Implications for an Incremental Crawler. In: Proc. of VLDB 2000. pp. 200–209 (2000)
6. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2020 Deep Learning Track. In: Proc. of TREC 2020. NIST (2020)
7. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Lin, J.: MS MARCO: benchmarking ranking models in the large-data regime. In: Proc. of SIGIR 2021. pp. 1566–1576. ACM (2021)
8. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.: Overview of the TREC 2019 Deep Learning Track. In: Proc. of TREC 2019. NIST (2019)
9. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2021 Deep Learning Track. In: Voorhees, E.M., Ellis, A. (eds.) Notebook. NIST (2021)
10. Dai, Z., Callan, J.: Context-Aware Document Term Weighting for Ad-Hoc Search. In: Proc. of WWW 2020. pp. 1897–1907. ACM (2020)
11. Feng, L., Shu, S., Lin, Z., Lv, F., Li, L., An, B.: Can Cross Entropy Loss Be Robust to Label Noise? In: Proc. of IJCAI 2020. pp. 2206–2212. ijcai (2020)
12. Fetterly, D., Manasse, M., Najork, M.: On the Evolution of Clusters of Near-Duplicate Web Pages. In: Proc. of (LA-WEB 2003. pp. 37–45 (2003)
13. Fetterly, D., Manasse, M., Najork, M., Wiener, J.: A Large-Scale Study of the Evolution of Web Pages. In: Proc. of WWW 2003. pp. 669–678 (2003)
14. Frénay, B., Verleysen, M.: Classification in the Presence of Label Noise: A Survey. IEEE Trans. on Neural Networks Learn. Syst. **25**(5), 845–869 (2014)
15. Fröbe, M., Bevendorff, J., Gienapp, L., Völske, M., Stein, B., Potthast, M., Hagen, M.: CopyCat: Near-Duplicates Within and Between the ClueWeb and the Common Crawl. In: Proc. of SIGIR 2021. pp. 2398–2404. ACM (2021)
16. Gao, L., Dai, Z., Fan, Z., Callan, J.: Complementing lexical retrieval with semantic residual embedding. CoRR **abs/2004.13969** (2020)
17. Kaszkiel, M., Zobel, J.: Passage Retrieval Revisited. In: Proc. of SIGIR 1997. pp. 178–185. ACM (1997)
18. Lin, J., Ma, X., Lin, S., Yang, J., Pradeep, R., Nogueira, R.: Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In: Proc. of SIGIR 2021. pp. 2356–2362. ACM (2021)
19. MacAvaney, S., Macdonald, C., Ounis, I.: Reproducing Personalised Session Search over the AOL Query Log. In: Proc. of ECIR 2022 (2022)
20. MacAvaney, S., Yates, A., Feldman, S., Downey, D., Cohan, A., Goharian, N.: Simplified Data Wrangling with ir_datasets. In: Proc. of SIGIR 2021. pp. 2429–2436. ACM (2021)
21. Macdonald, C., Tonellotto, N., MacAvaney, S., Ounis, I.: PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. In: Proc. of CIKM 2021. pp. 4526–4533. ACM (2021)

22. Mokrii, I., Boytsov, L., Braslavski, P.: A Systematic Evaluation of Transfer Learning and Pseudo-labeling with BERT-based Ranking Models. In: Proc. of SIGIR 2021. pp. 2081–2085. ACM (2021)
23. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A human generated machine reading comprehension dataset. In: Proc. of CoCo@N(eur)IPS 2016. CEUR, vol. 1773. CEUR-WS.org (2016)
24. Nogueira, R., Jiang, Z., Pradeep, R., Lin, J.: Document ranking with a pretrained sequence-to-sequence model. In: Findings of EMNLP 2020. pp. 708–718. ACL (2020)
25. Nogueira, R., Yang, W., Cho, K., Lin, J.: Multi-Stage Document Ranking with BERT. CoRR **abs/1910.14424**, 1–13 (2019)
26. Ntoulas, A., Cho, J., Olston, C.: What's new on the Web? The Evolution of the Web from a Search Engine Perspective. In: Proc. of WWW 2004. pp. 1–12. ACM (2004)
27. Olston, C., Pandey, S.: Recrawl Scheduling Based on Information Longevity. In: Proc. of WWW 2008. pp. 437–446. ACM (2008)
28. Pradeep, R., Nogueira, R., Lin, J.: The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. CoRR **abs/2101.05667**, 1–23 (2021)
29. Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W., Dong, D., Wu, H., Wang, H.: Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In: Proc. of NAACL 2021. pp. 5835–5847 (2021)
30. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020)
31. Rudra, K., Anand, A.: Distant supervision in bert-based adhoc document retrieval. In: Proc. of CIKM 2020. pp. 2197–2200. ACM (2020)
32. Singla, A., White, R.: Sampling High-Quality Clicks from Noisy Click Data. In: Proc. of WWW 2010. pp. 1187–1188. ACM (2010)
33. Voorhees, E., Craswell, N., Lin, J.: Too Many Relevants: Whither Cranfield Test Collections? In: Proc. of SIGIR 2022. ACM (2022)
34. Wu, X., Liu, Q., Qin, J., Yu, Y.: PeerRank: Robust Learning to Rank With Peer Loss Over Noisy Labels. IEEE Access **10**, 6830–6841 (2022)
35. Yates, A., Arora, S., Z., X., Yang, W., Jose, K., Lin, J.: Capreolus: A toolkit for end-to-end neural ad hoc retrieval. In: Proc. of WSDM 2020. pp. 861–864. ACM (2020)
36. Yates, A., Nogueira, R., Lin, J.: Pretrained Transformers for Text Ranking: BERT and Beyond. In: Proc. of SIGIR 2021. pp. 2666–2668. ACM (2021)
37. Zhan, J., Mao, J., Liu, Y., Guo, J., Zhang, M., Ma, S.: Optimizing dense retrieval model training with hard negatives. In: Proc. of SIGIR 2021. pp. 1503–1512. ACM (2021)
38. Zhan, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: Repbert: Contextualized text embeddings for first-stage retrieval. CoRR **abs/2006.15498** (2020)