A Reproducibility Study of Question Retrieval for Clarifying Questions

Sebastian Cross(✉), Guido Zuccon, and Ahmed Mourad

The University of Queensland, St Lucia, Australia
sebastian.cross@uq.net.au, g.zuccon@uq.edu.au

Abstract. The use of clarifying questions within a search system can have a key role in improving retrieval effectiveness. The generation and exploitation of clarifying questions is an emerging area of research in information retrieval, especially in the context of conversational search.

In this paper, we attempt to reproduce and analyse a milestone work in this area. Through close communication with the original authors and data sharing, we were able to identify a key issue that impacted the original experiments and our independent attempts at reproduction; this issue relates to data preparation. In particular, the clarifying questions retrieval task consists of retrieving clarifying questions from a question bank for a given query. In the original data preparation, such question bank was split into separate folds for retrieval – each split contained (approximately) a fifth of the data in the full question bank. This setting does not resemble that of a production system; in addition, it also was only applied to learnt methods, while keyword matching methods used the full question bank. This created inconsistency in the reporting of the results and overestimated findings. We demonstrate this through a set of empirical experiments and analyses.

1 Introduction

Creating a single query that is complex and detailed enough to retrieve the required information accurately is a difficult task. Failure often requires users to recreate and rewrite the query several times to get their desired information. This issue has led to the development of systems designed to assist the user with query formulation [6,11,15,24–26]. These systems implement multiple methods, one being asking for *clarifying questions* [3]. Clarifying questions help to identify a user's information-seeking intent by identifying if their query meets an ambiguity threshold [12]. If this is the case, the system poses a clarifying question to the user, expecting their answer to clarify aspects of their query. Asking clarifying questions has been recognised as an increasingly useful feature for conversational search [1–4,12,13,20,27,29]. In this context, the search agent often can only present a limited set of results (e.g., one, a handful, or even an answer synthesised from some top results) and thus the need for clear intent-driven queries is further exacerbated.

Developing methods for asking clarifying questions has become a recent focus in information retrieval, with Aliannejadi et al.'s work [3] being a key milestone.

J. Kamps et al. (Eds.): ECIR 2023, LNCS 13982, pp. 35–50, 2023.
https://doi.org/10.1007/978-3-031-28241-6_3

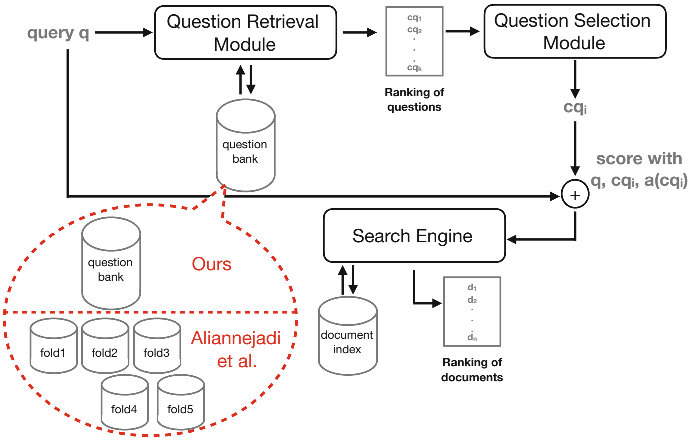

Fig. 1. Overview of a retrieval pipeline that exploits clarifying questions. Our work specifically focuses on the question retrieval module. In red, we highlight the key issue related to data preparation in the context of Aliannejadi et al.'s paper we reproduce [3]: we show the difference between our data preparation (i.e. use the whole question bank to retrieve against) and Aliannejadi et al.'s data preparation (i.e. divide the question bank into 5 folds to retrieve against). (Color figure online)

This work modelled solutions for clarifying questions as a two-step process: question retrieval and question selection. They then contributed methods that adhere to the two-step process along with an evaluation methodology and a rich dataset (Qulac). The work by Aliannejadi et al. [3] has been key for the establishment of the task of clarifying questions and has been relied upon by many others to build upon their research [2,4,10,12,13,20,27,29,31,35,39].

Our paper aims to reproduce that original work [3], and focuses specifically on the key step of *question retrieval*, see Fig. 1. The question retrieval module receives the user query as input and retrieves from a question bank a list of candidate clarifying questions. The original paper explored a number of methods for tackling this task. Methods implementation, data preparation and raw results were not made available with the original work, and we sought to then create reference implementations and data preparations for others to build upon and compare – with the intention of building ourselves our future work on this task based on these implementations. In reproducing this work, and working closely with two of the original authors who shared their data, we were able to identify a key divergence between common data preparation practice and what the authors did. Data preparation was necessary as 5-fold cross validation was used in the empirical experiments. The original authors partitioned into folds both the query set and the target data for retrieval (i.e. the index of clarifying questions), while standard practice in information retrieval is to partition the query set only, and instead maintain the index on which retrieval is performed unchanged across the folds. The data preparation they adopted was not documented explicitly in the original paper. This issue, as we show in this paper, leads to a considerable overestimation of the effectiveness of some of the methods studied in the paper.

In addition to this key issue, we then further identified other issues related to the features used for learning to rank and ties in retrieval scores – which however had less impact on results. Overall, the data preparation issue has important implications because many others have built upon Aliannejadi et al.'s milestone paper, but apparently did so without reproducing that work.

The remainder of the paper is organised as follows. In Sect. 2 we introduce the task of question retrieval for clarifying questions, along with the methods originally proposed by Aliannejadi et al., which adapted well-known information retrieval methods to the retrieval of questions from a question bank [3]. In Sect. 3 we describe the key issue regarding data preparation present in the original work, along with what we argue a more common and realistic[1] data preparation for this task should have been. Section 4 lists the experimental settings used to reproduce the original work and to study the differences in results caused by the differences in data preparation. We then report and analyse the experimental results in Sect. 5. For this, we develop the analysis along 3 main directions: (1) the effectiveness of methods on the data preparation of Aliannejadi et al., (2) their effect on what we repute as the correct data preparation for this task, and (3) an analysis of issues related to the topics and question-bank, their representation and the difference that these generate across keyword matching and learnt models.

2 Question Retrieval for Clarifying Questions

2.1 The Question Retrieval Task

Figure 1 provides an overview of a system that uses clarifying questions to improve the search effectiveness of document retrieval. In the context of this paper, we focus on the first module: the question retrieval module. In this context, the question retrieval module takes as input the original user query q and uses it to retrieve candidate clarifying questions organised as a ranking $\mathcal{R}_{cq} = <cq_1, cq_2, \ldots, cq_k>$ from a question bank $\mathcal{QB}$. We note that the question bank $\mathcal{QB}$ is not tailored a-priori to the query q: that is, $\mathcal{QB}$ contains a large set of clarifying questions, some of which applicable to any of the users' queries (or any of the queries that are deemed to needing a clarifying question). This is an important aspect to stress because, as we shall discuss in Sect. 3, it should be reflected in how the data in the $\mathcal{QB}$ should be prepared for experimentation. We further note that the question retrieval module would not be necessary if questions were generated on the fly given the input (e.g., via a generative language model), rather than retrieved from a question-bank [1,32,35,38]; this is however not the setting considered by the reference paper we aim to reproduce.

2.2 Methods for Question Retrieval

In the original paper, Aliannejadi et al. adapted the question retrieval task methods from three broad families of retrieval models: keyword-matching models [23, 36], learning to rank models [16,19], and transformer-based models [17,30].

[1] In that it resembles what a production system may look like.

As keyword matching models, they considered the common Query Likelihood (QL) with Dirichlet smoothing [36], BM25 [23], and the use of RM3 pseudo-relevance feedback method on top of QL [14].

As learning to rank models, they considered LambdaMART and RankNet [16]; both are pairwise models. As features to represent a query-question pair, they used the QL, BM25 and RM3 scores. This representation choice should be kept in mind, as we further analyse the implication of this later when examining the results: while this specific choice is not the focus of our paper, we do argue that this choice is problematic if considering the broader generalisation of these learning to rank methods for question retrieval.

As transformer model, the original authors introduced BERT-LeaQuR, which constituted one of the key original contributions of that work. From the description of BERT-LeaQuR, we understand that the model structure is similar to that of a typical cross-encoder ranker like monoBERT [22], but where the pre-trained language model is directly fine-tuned on the target dataset for question retrieval (see Sect. 4 for a description of the dataset). To clarify the implementation of BERT-LeaQuR we contacted the original authors to also acquire the corresponding source code. We were however told that one of their follow-up work yielded a stronger transformer model [1,2], and we were advised to use that for reproduction instead of BERT-LeaQuR.

3 Issues with Data Preparation

The experiments in the original paper used the Qulac question bank. According to the original paper, this question bank contained 2,649 clarifying questions, but the question bank made available by the authors in the repository associated with the paper contained 2,593 questions; we used this available question bank. These questions were assembled through crowdsourcing tasks and with respect to 198 target queries from the TREC 2009, 2010, and 2011 Web Track collection [9,28]. For each topic, only a small subset of the clarifying questions in the question bank is relevant to the specific topic.

3.1 Early Investigation of Reproducibility

In our early attempts to reproduce the question retrieval methods from the original paper, we were failing to obtain similar results as in the original experiments for the learnt models (LabdaMART, RankNet, BERT). On the other hand, we were able to obtain similar results for the keyword-matching models. In particular, our results on some of the learnt models had lower effectiveness compared to the keyword matching models: an unexpected result, especially in light of the results reported in the original paper. This triggered an in-depth analysis of the dataset and runs. Yet, we could not identify specific faults in our implementations or use of toolkits such as RankLib[2].

[2] https://sourceforge.net/p/lemur/wiki/RankLib.

We then decided to contact the authors for advice. While they also could not identify why we were unable to reproduce the results, with a genuine collaborative and supportive spirit, they were able to retrieve from back-ups and share the feature files and the saved models they created for learning to rank. We then turned to examine these files. We started by running their saved models on their test files, which returned similar results to those originally reported. We then retrained the learning to rank models using our settings and their dataset files (train, validation and test files) – obtaining the same models and results of the original experiment. Yet, we were unable to reproduce the results if we changed to our dataset files (train, validation and test files).

While we expected minor differences in effectiveness due to different random splitting[3] of topics into train, validation and test files (the dataset was split into 5 folds to allow for 5-fold cross-validation), we could not reconcile this being the reason for the remarkable drop in effectiveness, rendering the trends we observed being widely different from those in the original work. This then triggered a review of the train, validation and test files for the learning to rank models.

3.2 Analysis of Data Preparation and Differences Identified

The train, validation and test files for learning to rank contained a list of query-question pairs for several topics. Each pair was represented by three features: the BM25, QL and RM3 scores. A binary label was associated with each pair: 1 if the clarifying question was for that query-question pair, 0 otherwise. For this data, pairs originated from the Qulac question bank.

When we examined and compared the original and our train, validation and test files we identified two differences.

The first was a minor difference. Our features were the BM25, QL and RM3 scores as computed by the models (or, more precisely, by the Anserini toolkit[4]). This meant for example that if a candidate clarifying question did not contain any query term, the BM25 score we assigned to the question and therefore we used for the corresponding feature was 0. This was not the case however in the files given to us by Aliannejadi et al. In these files, retrieval scores appeared to have been smoothed – we believe by adding an ad-hoc $\epsilon \neq 0$ value (akin to Laplace smoothing in language modelling [37]). While this smoothing has, in practice, no effect on the learning to rank models that were created, this highlighted as most of the query-question pairs for a topic had a feature representation that was zero-valued (in our file) – so many pairs had the same exact representation. We comment on this aspect in Sect. 5.3.

The second instead was a major difference. The experimental setup used by Aliannejadi et al. required to train and test learnable models (learning to rank, transformers) using a 5-fold cross-validation setup (60% train, validation 20%,

[3] Note this was true in early experiments, but in the experiments reported in this paper, we were able to reproduce the exact split of topics into folds as they had.

[4] We note that different information retrieval toolkits follow different reference implementation of some of the keyword matching methods, e.g. of BM25.

test 20%). Therefore, when we prepared the data, we created 5 folds by dividing the topics. However, we did not divide the question bank into folds. This meant that, for example, the test file for a fold contained 40 topics (queries). For each topic, the file contained 2,593 candidate query-question pairs, i.e. all the possible clarifying questions in the question bank. This is akin to the common practice in information retrieval when performing n-fold cross-validation: topics are divided into n folds, but retrieval occurs over the whole candidate set (the entire index[5]).

The setup we produced in our data preparation mimics that of a production system. In this case, a question bank would not be limited to a set of queries. Attempts would be made instead to source questions that can cover a large portion of queries that users would issue. Thus, when experimenting with methods for question retrieval, the entire set of candidate clarifying questions should be considered, i.e. retrieval should take place from the whole question bank.

However, when examining the train, validation and test files from the original work, we noticed that: (1) each of the train, test, and validation not only contained a subset of topics, as expected, but it also contained just a subset of all candidate clarifying questions, (2) these subsets always contained all the relevant questions for a given topic, (3) the folds did not contain 2,593 clarifying questions as in their released question bank, but 2,609, and we could not exactly map one to the other because of different identifiers been used. In other words, while our data files contained, for each topic, the same 2,593 clarifying questions, their data files contained on average 1,558.8, 521.8, and 521.8 clarifying questions for train, validation and test files, respectively. These statistics are clearly compared side by side in Table 1. This difference, also visualised in Fig. 1, had two implications, which we discuss in Sect. 3.3.

We already highlighted the difference between the question bank statistics reported in the original paper (size: 2,639 questions), the learning to rank files provided to us (2,609), and the question bank made available publicly in their repository[6] (2,593). When asked, the authors recalled that they modified the data after publication of the original paper. Our experiments, including when examining both our and their data partition, are based on the question bank with 2,593 questions. We, therefore, expect *minor* differences in evaluation metrics' absolute values when compared to the results reported in the original paper.

3.3 Implications of the Differences in Data Preparations

The first implication is that at *training time*, for any given topic, we provide to the model an average of 13.1 relevant clarifying questions and 2,579.9 non-relevant clarifying questions. In Aliannejadi et al.'s experiments, instead, because

[5] Note that commonly in learning to rank, feature files are created for the top-k candidate documents. This however is not because retrieval only considers k documents. Learning to rank is unfeasible for large collections, and is therefore part of a cascade pipeline where full index retrieval occurs first with a cheaper model, and then learning to rank is applied to the top-k. Yet, retrieval considers the full index, not an arbitrary subset that – what the chances – contains all relevant documents.

[6] https://github.com/aliannejadi/qulac.

Table 1. Statistics of the original data preparation (folds) by Aliannejadi et al. [3] compared to Ours. The number of topics for train/validation/split are the same across the two preparations. Differences are found with regards to the number of clarifying questions per topic. Note that Aliannejadi et al.'s data preparation statistics computed on the learning to rank data they provided differ from those reported in their paper: they reported having 2,639 questions in total in their question-bank, while we could count only 2,609; yet the question bank released in their repository contained 2,593.

	Average number of clarifying questions per topic		Average number of topics per fold	
	Aliannejadi et al.	Ours	Aliannejadi et al.	Ours
Train	1,558.8	2,593	118.8	118.8
Validation	521.8	2,593	39.6	39.6
Test	521.8	2,593	39.6	39.6

of the different way of preparing the data, they provided the same number of *relevant* clarifying questions as we do, but *far less non-relevant* clarifying questions (on average, 523.9). This may make our learnt models weaker than theirs because our training data is much more imbalanced. However, we believed this not to be the case, because we observed most of the non-relevant clarifying questions have a feature representation that consists of zero-valued weights (i.e. there are no matching keywords in the questions).

The second and most important implication is that at *retrieval time (test)*, our learnt model has to rank 2,593 candidate clarifying questions per topic, of which, on average, only 13.1 are relevant. The learnt model in Aliannejadi et al.'s experiments instead had to rank only 537 candidate clarifying questions per topic, despite the number of relevant candidate clarifying questions per topic being the same. This means that our ranker is more likely to obtain lower effectiveness than their ranker. But this is not necessarily because it is a worse ranker. In fact, given a ranker $\mathcal{R}$ and an equal number of relevant candidate questions across two candidate sets S_1 and S_2, with $S_2 \supseteq S_1$ (and thus also $|S_1| < |S_2|$, where $|S_1|$ is the size of S_1), $\mathcal{R}$ is more likely to produce a ranking with higher effectiveness when applied to S_1than when applied to S_2, a (significantly, in the case of these experiments) larger superset of candidate questions than S_1. This is obvious for example when considering a relevant question with a zero-valued feature representation. In this case, $\mathcal{R}$ ranks the question at the bottom of the ranking[7]. In our data preparation, this means this relevant question is ranked at rank 2,593. However, in Aliannejadi et al.'s data preparation this same question, when ranked at the bottom of the ranking, would be at rank ≈521.8. This difference would translate into a sizable difference in effectiveness as measured for example by MAP. In fact, in the case of our data preparation, the MAP's gain [7] contributed by this question is 3.8565×10^{-4}, while in the case of Aliannejadi et al. is 1.9164×10^{-3} – one order of magnitude larger contribution to MAP.

[7] Possibly tied with other questions that also have a zero-valued feature representation, which, in the dataset considered, are the majority of them.

Given these differences in data preparations, we asked ourselves if this could be the reason why we could not reproduce the original results, or at least trends. To investigate these aspects and the empirical differences induced by the two data preparations, we set up the experiments described and analysed in Sect. 5.

4 Experimental Settings

Datasets. The Qulac dataset used for the question bank and the topics have already been described in Sect. 3. Topics were split into folds for 5-fold cross-validation following those supplied by Aliannejadi et al.[8]. Additionally, we created two question bank preparations: one containing all clarifying questions from the question bank (referred to as Ours), and one following the division of clarifying questions into separate folds (referred to as Aliannejadi et al.).

Evaluation Metrics. We chose to use the same evaluation metrics as the original work. For question retrieval, these are: mean average precision (MAP) and recall for the top 10, 20, and 30 retrieved questions (Recall@10, Recall@20, Recall@30). In addition, we also report Success@1 and Precision@5: this is to understand the suitability of the rankings produced by question retrieval if questions were issued to users (without further refinement from the question selection model). In such a case, it is likely that 1 to 5 questions are asked to a user in a conversational or web setting.

We used the widespread `trec_eval` tool for computing metrics. However, `trec_eval` has an odd treatment of items with a tied score: the rank position information in the result file is discarded, and tied items are ordered alphanumerically [5,18,21,34]. This is often ignored in information retrieval experiments, however, this also arises as a (minor) issue in the experiments in this paper. We explain why this is the case in Sect. 5.3. To avoid ties, we post-process all results, including those from learning to rank, to assign to each question a unique score such that decreasing ranks correspond to decreasing scores.

For statistical significance analysis, we use a paired two-tails t-test with Bonferroni correction and regard a difference as significant if $p < 0.05$; however no significant differences were found in our experiments.

Models Implementation. For the keyword matching models, the original authors used the implementations from the Galago search engine toolkit [8]. In our reproduction, we instead use the implementation of these methods from the Anserini/Pyserini toolkit [33]. The use of these different libraries is likely to have caused minor differences in the runs produced, e.g., due to implementations, parameter settings[9], stemming and stop-listing[10]. Because of this difference in

[8] Once we obtained the feature files for learning to rank, we knew which topics were grouped together in which fold, and thus could recreate the same topic-wise division.

[9] Ours: (BM25) $k_1 = 0.9$, $b = 0.4$, (QL) $\mu = 1000$, (RM3) $fb_{terms} = 10$, $fb_{docs} = 10$ $original\_query\_weigh = 0.5$. They do not report parameter values.

[10] We used Porter Stemmer and Anserini's default stop-list. They do not report their settings.

toolkits, it is important to not directly compare the absolute numbers obtained by the methods in the original work vs. in our reproduction: comparison should instead take place with respect to the trends that are observed when comparing across models.

For the learning to rank models, we used the RankLib toolkit, as did Aliannejadi et al.[11]. They do not indicate if feature normalisation was performed. We experimented with the three normalisations made available in the toolkit (zscore, sum, and linear) and no normalisation. We used those that gave us the best effectiveness and were closer to the original values: no normalisation for LambdaMART and zscore for RankNet. As per features, we directly used the scores of QL, BM25 and RM3, with no further processing (e.g., smoothing) as no further processing was reported by Aliannejadi et al. Regardless, we found in testing that smoothing did not affect results.

For the BERT model, we used the `bert-base-uncased`[12] checkpoint made available by the Huggingface library. The architecture of the model resembled that of a monoBERT cross-encoder ranker [22], with the difference that inputs were pairs of query-question rather than query-document. The checkpoint was then fine-tuned on the Qulac dataset; fine-tuning occurred on the training portion of a fold. The implementation of this method was made publicly available[13] by Aliannejadi et al. in a separate publication [1,2].

5 Experiments and Results Analysis

Next, we report and analyse the experimental results obtained when attempting to reproduce the question retrieval component from Aliannejadi et al. For this, we develop the analysis along 3 main directions, as indicated in Sect. 1.

5.1 Experiment 1: Aliannejadi et al.'s Data Preparation

We start by attempting to replicate the results reported by Aliannejadi et al., using their data preparation based on the splitting of the question bank into subsets across train, validation and test. Our results should be compared with Table 2 of the original paper. We do not expect to obtain the same exact values of evaluation measures: (i) we know minor differences would be present because of Galago vs. Anserini – this may influence absolute values, but not the trends, (ii) their BERTleaQuR and the BERT cross-encoder we implemented may have minor differences, (iii) they reported their question-bank having 2,639 but the one we have access to has 2,593. However, we expect to observe the same trends, i.e. that learning to rank methods are superior to keyword-matching methods, and that the BERT-based method largely outperforms all others. We believe they have a mismatch in the data in the paper and associated repository, but we confirmed they ran their experiments on the data they gave us. Also, if there was

[11] We used version 2.17; Aliannejadi et al. did not report the version.

[12] https://huggingface.co/bert-base-uncased.

[13] https://github.com/aliannejadi/ClariQ.

Table 2. Question retrieval results for Aliannejadi et al.'s data preparation, which splits the question bank into 5 folds. These results are the replication of the results reported by Aliannejadi et al.'s original work in their Table 2 [3].

Aliannejadi et al. data preparation						
Method	MAP	Recall@10	Recall@20	Recall@30	Success@1	Precision@5
QL	0.7183	0.6426	0.7376	0.7394	0.9795	0.9329
BM25	0.7198	0.6426	0.7376	0.7393	0.9795	0.9380
RM3	0.7198	0.6426	0.7376	0.7393	0.9795	0.9380
LambdaMART	0.7274	0.6299	0.7253	0.7323	0.9697	0.9364
RankNet	0.7406	0.6352	0.7372	0.7498	0.9697	0.9354
BERT	0.8352	0.6868	0.8345	0.8673	0.9848	0.9657

a mismatch, it would likely affect only a handful of queries – regardless, it would be expected to impact absolute values but not trends. Results are reported in Table 2; we make the following observations:

1. QL, BM25, RM3: we were able to obtain consistently higher effectiveness than that reported in the original paper, across all metrics (e.g., for MAP they reported QL: 0.6714, BM25: 0.6715, RM3: 0.6858 [3]). While explanations for this could be because of points (i) and (iii) above, we do not believe these are the core reasons. Instead, we believe Aliannejadi et al. did not execute the keyword matching retrieval against the same data preparations (and thus, subdivisions of the question bank) they use for the learnt models (i.e. the ones used in these experiments). Instead, we believe the results they reported for keyword matching methods were obtained against the whole question bank: this is the setup we argue should have been used to evaluate *all* methods. We investigate this setup in Sect. 5.2. We show that in that setup we obtain effectiveness values for keyword-matching methods that are much closer to the ones they originally reported.
2. LambdaMART and RankNet: we were not able to obtain the same effectiveness reported in the original paper, but values are close (e.g., the reported MAP for LambdaMART is 0.7218, for RankNet is 0.7304 [3]). Differences may likely be due to the feature files they used for the paper containing more questions than the ones they gave us. The absence in our question bank of these additional questions made that effectiveness higher: intuitively this is because most of them are not relevant for most topics, and thus removing them improves rankings if they appeared before a relevant question.
3. BERT: we obtained values that are close to the ones they reported in the original paper (e.g., MAP 0.8349 [3]). While there were minor differences, we ascribe these differences mainly to points (ii) and (iii) above.

Overall, with minor discrepancies, we were able to obtain similar results to the ones reported in the original paper for the learned models (LambdaMART, RankNet, BERT), but not for the keyword-matching models (QL, BM25, RM3). Trends across models were as they reported: BERT is the best model, followed by

Table 3. Question retrieval results for our data preparation. These results strongly differ from those reported by the original work of Aliannejadi et al. in their Table 2 [3] for the learned methods, i.e. LambdaMART, RankNet, BERT.

Our data preparation						
Method	MAP	Recall@10	Recall@20	Recall@30	Success@1	Precision@5
QL	0.6975	0.6152	0.7218	0.7238	0.9442	0.9177
BM25	0.6979	0.6167	0.7201	0.7321	0.9492	0.9187
RM3	0.6979	0.6167	0.7201	00.7321	0.9492	0.9187
LambdaMART	0.6728	0.5882	0.6947	0.7068	0.9394	0.8889
RankNet	0.6851	0.6028	0.7051	0.7171	0.9293	0.9020
BERT	0.7512	0.6349	0.7686	0.7979	0.9596	0.9131

the learning to rank methods, with the keyword matching models being the worst – though gains over keyword matching models were not as large as those they reported. We believe they however incorrectly reported the values for keyword-matching models. Specifically, we believe values for keyword matching models were obtained when retrieving on the whole question bank, rather than the smaller splits they created for the learnt methods, see next.

5.2 Experiment 2: Our Data Preparation

We now consider our data preparation, where question retrieval occurs against a unique question bank, which contains all possible clarifying questions for all topics. Results are reported in Table 3; we make the following observations:

1. QL, BM25, RM3: the results we obtained when searching on the whole question bank appear to be more akin to those Aliannejadi et al. reported for their experiments (e.g., for MAP they reported QL: 0.6714, BM25: 0.6715, RM3: 0.6858 [3]) than in the data preparation setup of Sect. 5.1. Differences could be ascribed to tools (Anserini vs. Galago), model parameters, and question bank size.
2. LambdaMART and RankNet: our data preparation setup led to learning to rank methods achieving lower effectiveness than keyword matching models. This is the opposite trend of that reported in the original work, and also is opposite to what we found for Aliannejadi et al.'s data preparation in Sect. 5.1.
3. BERT: we found that on our data preparation, BERT performed worst than on theirs. While BERT was still the best method across all those considered, the gains over keyword matching were sensibly lower, e.g., for MAP a 7.64% gain in ours vs. 24.33% in theirs compared to BM25.

Overall, the results of these experiments (Table 3) differ greatly from those reported for Aliannejadi et al. 's data preparation (Table 2). Specifically, we found that learning to rank models cannot outperform keyword matching when retrieval occurs on the whole question bank, and in this setting, BERT does provide improvements over keyword matching, but not at the rate reported. Importantly,

Table 4. Statistics of the data preparations by Aliannejadi et al. and ours. While both data preparations have the same number of relevant documents and relevant documents with keyword matching score equal to zero, they consistently differ in terms of the number of non-relevant documents with keyword matching score above zero.

Statistic	Aliannejadi et al	Ours
avg. # relevant questions per query	13.1	13.1
avg. # relevant questions per query with zero score	3.5	3.5
avg. # non-relevant questions per query with score > 0	1.9	13.3

these improvements are not statistically significant. This result empirically demonstrates that the two data preparations lead to different estimations of effectiveness and different overall findings.

5.3 Further Analysis: Zero Scores, Ties

Next, we analyse two aspects of the data and experiments of this paper: the use of keyword matching scores as only features in learning to rank, and the presence of tied scores in the rankings.

Zero Score. The scores of QL, BM25 and RM3 are used in the learning to rank methods as only features for representing query-question pairs. This resulted in two characteristics arising: (1) there were a number of pairs with the same non-zero representation, (2) many pairs had a representation that was zero-valued for all three features. Characteristic 2 occurred often for non-relevant questions, but sporadically it occurred also for relevant questions: in fact on average each query had 3.5 relevant questions that had their features being all zeros (see Table 4). This fact, combined with the fact that items that had a non-zero feature representation often had their representation been the same as another item, meant that the learning to rank methods often ended up assigning to pairs at test time one of two scores: 0 or 1. This caused many ties in the ranking (see below). The analysis of the features files also revealed another problem when comparing Aliannejadi et al.'s data preparation and ours: in theirs on average there was only 1.9 non-relevant questions that had features that were not all zeros, while in ours there were 13.3 – and this would have made ranking harder in our data preparation.

Ties. As mentioned above, the keyword matching methods and the learning to rank methods produced rankings with a large number of ties. The `trec_eval` tool behaves oddly when ties are present (see Sect. 4), while RankLib considers the actual rank position recorded in the ranking file. We are unsure which tool Aliannejadi et al. used to report their results, and if mixing `trec_eval` for keyword matching and BERT methods and RankLib for learning to rank, the evaluation would have been inconsistent. We show this in Table 5 for LambaMART. In our evaluations we transformed scores as a function of their rank and used `trec_eval`, so that no ties were present.

Table 5. Differences in MAP between evaluation tools when analysing the LambdaMART run on our data preparation: RankLib evaluation, `trec_eval`, and `trec_eval` with ties removed by converting scores to a function of rank.

RankLib eval	0.6728
`trec_eval`	0.7233
`trec_eval` no ties	0.6728

6 Conclusions

The use of clarifying questions within a search system can have a key role in improving retrieval effectiveness and user interaction with the system, especially if this is a conversational search system. In this paper, we attempted to reproduce the work by Aliannejadi et al., which is a key milestone in the area of clarifying questions for search. Working closely with the original authors and thanks to their sharing of data, we identified a fundamental issue related to data preparation. In particular, their practice of dividing the question bank containing clarifying questions into folds is, we believe, unrealistic for a production system, and is also different from standard information retrieval experimentation practice. Throughout our experiments and analyses, we have shown how this issue affected the results reported in the original work.

We found that learning to rank models cannot outperform keyword matching when retrieval occurs on the whole question bank. We also found that while BERT does outperform keyword matching methods in this setting, it does so with much smaller gains than what was originally reported and, importantly, these differences are not statistically significant. We do not believe this is a generalisable result. Specifically, we believe this result is due to: (i) the amount of training data being too little for those models, especially for BERT, and (ii) the feature representation being particularly poor for learning to rank models, where most questions had identical representation. We would expect that if these two points were addressed, learnt models would provide consistently better results than keyword matching, as it often occurs in other information retrieval tasks.

This work demonstrates that it is critical to be able to communicate and share resources among researchers to facilitate the reproduction of methods and results and the identification of possible factors that may have influenced results beyond the intentions of the original researchers.

We make code, data preparations, run files and evaluation files publicly available at www.github.com/ielab/QR4CQ-question-retrieval-for-clarifying-questions.

Acknowledgments. This work was partially supported by Australian Research Council DECRA Research Fellowship (DE180101579).

References

1. Aliannejadi, M., Kiseleva, J., Chuklin, A., Dalton, J., Burtsev, M.: ConvAI3: Generating Clarifying Questions for Open-Domain Dialogue Systems (ClariQ). arXiv:2009.11352 (2020)
2. Aliannejadi, M., Kiseleva, J., Chuklin, A., Dalton, J., Burtsev, M.: Building and evaluating open-domain dialogue corpora with clarifying questions. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 4473–4484 (2021)
3. Aliannejadi, M., Zamani, H., Crestani, F., Croft, W.B.: Asking clarifying questions in open-domain information-seeking conversations. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 475–484 (2019)
4. Bi, K., Ai, Q., Croft, W.B.: Asking clarifying questions based on negative feedback in conversational search. In: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 157–166 (2021)
5. Cabanac, G., Hubert, G., Boughanem, M., Chrisment, C.: Tie-breaking bias: effect of an uncontrolled parameter on information retrieval evaluation. In: Agosti, M., Ferro, N., Peters, C., de Rijke, M., Smeaton, A. (eds.) CLEF 2010. LNCS, vol. 6360, pp. 112–123. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15998-5_13
6. Cai, F., De Rijke, M., et al.: A survey of query auto completion in information retrieval. Found. Trends® Inf. Retrieval **10**(4), 273–363 (2016)
7. Carterette, B.: System effectiveness, user models, and user utility: a conceptual framework for investigation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 903–912 (2011)
8. Cartright, M.A., Huston, S.J., Feild, H.: Galago: a modular distributed processing and retrieval system. In: Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval, pp. 25–31 (2012)
9. Clarke, C.L., Craswell, N., Soboroff, I.: Overview of the TREC 2009 web track. In: Proceedings of TREC (2009)
10. Dubiel, M., Halvey, M., Azzopardi, L., Anderson, D., Daronnat, S.: Conversational strategies: impact on search performance in a goal-oriented task. In: The Third International Workshop on Conversational Approaches to Information Retrieval (2020)
11. Fails, J.A., Pera, M.S., Anuyah, O., Kennington, C., Wright, K.L., Bigirimana, W.: Query formulation assistance for kids: what is available, when to help & what kids want. In: Proceedings of the 18th ACM International Conference on Interaction Design and Children, pp. 109–120 (2019)
12. Kim, J.K., Wang, G., Lee, S., Kim, Y.B.: Deciding whether to ask clarifying questions in large-scale spoken language understanding. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 869–876. IEEE (2021)
13. Krasakis, A.M., Aliannejadi, M., Voskarides, N., Kanoulas, E.: Analysing the effect of clarifying questions on document ranking in conversational search. In: Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval, pp. 129–132 (2020)
14. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: ACM SIGIR Forum, vol. 51, pp. 260–267. ACM, New York (2017)

15. Lee, C.-J., Lin, Y.-C., Chen, R.-C., Cheng, P.-J.: Selecting effective terms for query formulation. In: Lee, G.G., et al. (eds.) AIRS 2009. LNCS, vol. 5839, pp. 168–180. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04769-5_15
16. Li, H.: Learning to rank for information retrieval and natural language processing. Synth. Lect. Hum. Lang. Technol. **7**(3), 1–121 (2014)
17. Lin, J., Nogueira, R., Yates, A.: Pretrained transformers for text ranking: BERT and beyond. Synth. Lect. Hum. Lang. Technol. **14**(4), 1–325 (2021)
18. Lin, J., Yang, P.: The impact of score ties on repeatability in document ranking. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1125–1128 (2019)
19. Liu, T.Y., et al.: Learning to rank for information retrieval. Found. Trends® Inf. Retrieval **3**(3), 225–331 (2009)
20. Lotze, T., Klut, S., Aliannejadi, M., Kanoulas, E.: Ranking clarifying questions based on predicted user engagement. In: MICROS Workshop at ECIR 2021 (2021)
21. McSherry, F., Najork, M.: Computing information retrieval performance measures efficiently in the presence of tied scores. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 414–421. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78646-7_38
22. Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)
23. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: BM25 and beyond. Found. Trends® Inf. Retrieval **3**(4), 333–389 (2009)
24. Russell-Rose, T., Chamberlain, J., Shokraneh, F.: A visual approach to query formulation for systematic search. In: Proceedings of the 2019 Conference on Human Information Interaction and Retrieval, pp. 379–383 (2019)
25. Scells, H., Zuccon, G., Koopman, B.: A comparison of automatic boolean query formulation for systematic reviews. Inf. Retrieval J. **24**(1), 3–28 (2021)
26. Scells, H., Zuccon, G., Koopman, B., Clark, J.: Automatic boolean query formulation for systematic review literature search. In: Proceedings of the Web Conference 2020, pp. 1071–1081 (2020)
27. Sekulić, I., Aliannejadi, M., Crestani, F.: Towards facet-driven generation of clarifying questions for conversational search. In: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 167–175 (2021)
28. Soboroff, I.M., Craswell, N., Clarke, C.L., Cormack, G., et al.: Overview of the TREC 2011 web track. In: Proceedings of TREC (2011)
29. Tavakoli, L.: Generating clarifying questions in conversational search systems. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 3253–3256 (2020)
30. Tonellotto, N.: Lecture notes on neural information retrieval. arXiv preprint arXiv:2207.13443 (2022)
31. Vakulenko, S., Kanoulas, E., De Rijke, M.: A large-scale analysis of mixed initiative in information-seeking dialogues for conversational search. ACM Trans. Inf. Syst. (TOIS) **39**(4), 1–32 (2021)
32. Wang, J., Li, W.: Template-guided clarifying question generation for web search clarification. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 3468–3472 (2021)
33. Yang, P., Fang, H., Lin, J.: Anserini: reproducible ranking baselines using lucene. J. Data Inf. Qual. (JDIQ) **10**(4), 1–20 (2018)
34. Yang, Z., Moffat, A., Turpin, A.: How precise does document scoring need to be? In: Ma, S., et al. (eds.) AIRS 2016. LNCS, vol. 9994, pp. 279–291. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48051-0_21

35. Zamani, H., Dumais, S., Craswell, N., Bennett, P., Lueck, G.: Generating clarifying questions for information retrieval. In: Proceedings of the Web Conference 2020, pp. 418–428 (2020)
36. Zhai, C.: Statistical language models for information retrieval. Synth. Lect. Hum. Lang. Technol. **1**(1), 1–141 (2008)
37. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. (TOIS) **22**(2), 179–214 (2004)
38. Zhao, Z., Dou, Z., Mao, J., Wen, J.R.: Generating clarifying questions with web search results. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 234–244 (2022)
39. Zou, J., Kanoulas, E., Liu, Y.: An empirical study on clarifying question-based systems. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2361–2364 (2020)