CQADupStack: Gold or Silver?

Doris Hoogeveen[1,2] Karin M. Verspoor[2] Timothy Baldwin[1,2]
[1]NICTA
[2]Department of Computing and Information Systems
The University of Melbourne, VIC
Australia
dhoogeveen@student.unimelb.edu.au karin.verspoor@unimelb.edu.au tb@ldwin.net

ABSTRACT

In this paper we analyse the quality of a recently-released dataset for community question-answering (cQA) research, `CQADupStack` [8]. This set contains cQA data from the Stack Exchange community, annotated by the forum users with duplicate question information. While there is the expectation that the labels are of high quality, no previous analysis has been done to determine the quality of these annotations in terms of precision or recall. We provide evidence for the reliability of the existing duplicate question labels (precision) and identify in a sample that around 1% of the question pairs is missing a duplicate label (recall). This may extrapolate to as many as 39 million question pairs. Two strategies for sampling question pairs for possible annotation to boost recall are presented. Our results suggest that we can increase the number of duplicates by around 45%, by annotating only 0.0003% of all the question pairs in the data set.

1. INTRODUCTION

For a fair comparison of different methods proposed for a given task, it is important to have high quality test sets. It is nearly impossible to make such test sets perfect, but as long as the limitations are known, they can still be of high value to the research community. One such test set, `CQADupStack` [8],[1] has recently been released for community question-answering (cQA) research, targeting question retrieval in particular. While an active field of research, question retrieval has long suffered from the lack of a publicly available set of cQA data with duplicate question annotations, hampering comparison of retrieval methods. `CQADupStack` consists of twelve Stack Exchange[2] subforums and comes with predefined splits, to ensure maximum reproducibility and comparability of studies using it. The duplicate question labels in `CQADupStack` come directly from the users of the subforums. Users can close a question by marking it as a duplicate of an existing one, to avoid duplicated effort from the community.

Another way of identifying duplicate questions would be find question pairs that are answered by the same answer, but such information is unfortunately not available in `CQADupStack`. In the Stack Exchange subforums, which are the source of the data in `CQADupStack`, the usual way of labeling two questions as duplicates is by linking a question to another question. While it is possible to label a question as a duplicate of an answer, this happens far less often, possibly because the meaning of such a label is less well-defined. The usual scenario would be to mark the question as a duplicate of the archived question that received the relevant answer. We will therefore limit our analyses to such links or labels only.

No previous analysis has been done to establish how trustworthy these labels are, and to estimate the volume of missing labels; in other words, the quality of the data in terms of precision and recall of the provided relevance judgements is yet to be examined in detail. In this sense, the data should currently be considered a silver standard rather than a gold standard. In this paper we present a number of analyses and experiments aimed at quantifying this label uncertainty, and determining how close to a gold standard `CQADupStack` is.

2. RELATED WORK

Constructing large, realistic test sets for information retrieval (IR) research has long been recognised as a difficult task [2, 11, 15, 19]. One of the main problems is obtaining relevance judgements for all indexed documents, across a sizeable number of queries. It is generally accepted that as test sets become larger, not all documents will have received a manual relevance judgement. A common approach is to use methods that are less expensive and comprehensive than exhaustive expert manual annotation, to construct a set that is "good enough" rather than perfect. Given that the primary goal of test sets is to compare the performance of different systems, if this goal is achievable without perfect annotation, then perfection is not necessary [4]. Example approaches to achieve this are the following: crowdsourcing to cheaply source relevance judgements, via platforms such as Amazon's Mechanical Turk [1, 7],[3] or pooling to selectively sample the documents that require relevance judgements, as originally proposed in the context of TREC [17, 18].

It has been shown that the quality of the relevance judgements can have a large effect on system rankings [3, 5, 10]. One way to mitigate the effect of bad annotations is to use multiple annotators and apply a majority vote to obtain the final label. In this way, an annotated set can be made, with labels that are of comparable quality to those applied by experts [12]. This method can be improved by using methods such as the EM algorithm to jointly estimate the document relevance and the annotation quality [9]. Our work differs from the work above in that our labels are all supplied by

[1]http://nlp.cis.unimelb.edu.au/resources/cqadupstack/
[2]http://stackexchange.com/
[3]https://www.mturk.com/mturk/

Doris Hoogeveen, Karin Verspoor and Timothy Baldwin (2016) CQADupStack: Gold or Silver? In Proceedings of the SIGIR 2016 Workshop on Web Question Answering Beyond Factoids (WebQA 2016), Pisa, Italy.

experts rather than by the crowd, and majority voting has already been applied by the forum itself (see Section 3).

3. ESTIMATING THE LABEL PRECISION

In this section we try to quantify the quality of the data. A logical way to do this would be to take a random sample of the data, let an annotator provide judgements on whether question pairs are duplicates or not, and compare these to the labels in the data, to see how well they correlate. This could give us insights into both the quality of the existing labels and how many labels are missing. However, the data is heavily skewed towards the non-duplicate class (99.99% based on the original annotations), which causes a bias problem: a random sample of question pairs would consist predominantly (or only) of non-duplicate pairs.

The solution is to focus our analysis on label precision in the first instance, in artificially constraining the sample to contain a high density of duplicate question pairs. We do this by splitting the data of each subforum into a set of question pairs that have been labelled as duplicates, and a set of question pairs that have not been labelled as duplicates. We then randomly sample 50 duplicate question pairs and 50 non-duplicate question pairs from these sets (1200 question pairs in total, for all the subforums together). We sampled the questions randomly to avoid biasing the data toward particular methods as much as possible. The non-duplicate questions are added to the set to allow us to measure agreement with the existing labels more sensibly, and also, as we will see later, to give us a first insight into the recall.

The sampled questions were randomly shuffled and presented to our annotator (the first author of this paper), who was initially not provided with access to the labels. A second annotator (the third author of this paper) was presented with 200 randomly sampled question pairs from the same pool of 1200 question pairs the first annotator annotated, to see how much of an influence the bias in the set would have on the annotations. The inter-annotator agreement was 92%, with a Kappa coefficient [6] of 0.84. This shows that even though both of the annotators were aware of the 50/50 split of duplicate and non-duplicate pairs in the overall dataset, there were nonetheless occasional disagreements.

Table 1 shows the results of the first annotator's judgements in the form of false positives (question pairs that have a duplicate label in the dataset, but which the annotator judges as incorrect), false negatives (question pairs that do not have a duplicate label in the dataset, but which the annotator deems to be duplicates), and the corresponding precision and recall. That is, we treat the annotator's judgement as the ground truth and look at how well the labels in the data match these judgements.

As can be seen, the recall is very high. This means that there are not many question pairs that are duplicates according to the annotator, but which do not have a duplicate label in the dataset. The story is different for the precision however. For some subforums (e.g. `mathematica`) as many as 20% of the labelled duplicate pairs were not recognised as such by the annotator. Therefore, we next revealed the original labels to the annotator and asked her whether she wished to reconsider any of her labels. The false positives all but disappeared: only 2 remained that the annotator felt should not have a duplicate label. This would suggest that forum members are highly adept at recognising duplicates, even when they are not obvious at first sight.

One of the two question pairs that had been labelled as a duplicate by the forum users, but which our annotator disagreed with, is the following (from the `android` subforum):

> Q1: How can I install the Google Apps Package (Play Store, ...) on my Android device? [...]

> Q2: How to sync android market account if I have the app but the market isn't in my add account? [...]

One question asks about the installation of the app, while the other asks about linking an account to an already installed app.

The other duplicate question pair which the annotator disagreed with is the following, from the `wordpress` subforum:

> Q1: How Do I Configure Automatic Updates in WordPress 3.7? WordPress 3.7 added automatic updates. How does this work, and how can I configure this feature?

> Q2: Add pagination to my custom loop. Any tips on fitting pagination into my loop below and I owe you a beer or two. I've worked out how to pull the top rated articles by vote from a custom plugin, and lay it out with a count, but pagination has me stumped. [...]

There is one type of duplicate question which is particularly difficult to recognise without knowledge of the subforum community. This occurs when that community creates a wiki-style "super-post" that groups all questions about a certain topic together. A user posting any related question may be referred to this super-post and have their question marked as a duplicate. An example can be found in the `webmasters` subforum, where all questions regarding how to find suitable hosting arrangements are labelled as a duplicate of such a post (with its answers), regardless of the differences in the particular hosting needs of the user. In these cases, the users' questions are not semantically equivalent, and as such not strictly duplicates. However, this is part of the reality of the context, and such threads form a great source of information. Super-posts are created when many users ask questions about one topic that are strongly related in some way. This means such super-posts contain valuable information for many users, and ignoring them means excluding posts that may contain answers to many users' questions. Retrieval strategies should therefore handle such "duplicates" too. However, this structure and/or a lack of expertise may explain why the human annotator did not recognise some of the duplicate question pairs as being duplicates.

To understand the high precision that we have observed, it is worth understanding how the flagging of duplicate questions works in Stack Exchange. Not all users of the Stack Exchange subforums can close questions and mark them as duplicates of an archived question; only users with a high number of reputation points, i.e. users that have proven themselves to be knowledgeable and engaged in the forum, have this privilege. Out of the 26 different levels of privilege that one can achieve based on the number of reputation points earned, the level beyond which it is possible to tag duplicate questions is 21. In other words, one needs a strong reputation to unlock this privilege. This is to ensure that the duplicate labels are of a high quality. Even so, some duplicate labels may still be wrong, as we have seen.

	FP		FN		P	R
android	3	(1)	1	(0)	0.94 (0.98)	0.98 (1.00)
english	1	(0)	0	(0)	0.98 (1.00)	1.00 (1.00)
gaming	3	(0)	1	(1)	0.94 (1.00)	0.98 (0.98)
gis	8	(0)	0	(0)	0.84 (1.00)	1.00 (1.00)
mathematica	10	(0)	1	(0)	0.80 (1.00)	0.98 (1.00)
physics	8	(0)	1	(1)	0.84 (1.00)	0.98 (0.98)
programmers	2	(0)	2	(0)	0.96 (1.00)	0.96 (1.00)
stats	9	(0)	3	(2)	0.82 (1.00)	0.93 (0.96)
tex	9	(0)	2	(0)	0.82 (1.00)	0.95 (1.00)
unix	6	(0)	2	(1)	0.88 (1.00)	0.96 (0.98)
webmasters	4	(0)	1	(0)	0.92 (1.00)	0.98 (1.00)
wordpress	9	(1)	2	(1)	0.82 (0.98)	0.95 (0.98)
TOTAL	72/600	(2/600)	16/600	(6/600)	0.89 (1.00)	0.97 (0.99)

Table 1: An evaluation of 50 duplicate and 50 non-duplicate question pairs, manually annotated and compared to the original benchmark labels ("FP" = false positives and "FN" = false negatives, both relative to the original labels; "P" = precision, and "R" = recall; numbers in brackets show the revised evaluation after the annotator was shown the labels in the data set and given the opportunity to change her verdicts).

Another mechanism to ensure the quality of the duplicate labels is to allow other reputable users to cast votes for or against a duplicate label. The average number of votes for each duplicate label in CQADupStack is 3.5 votes. We can conclude from this that the duplicate labeling is really a community effort, and the labels are quite reliable, as confirmed by our analysis.

It can also happen that a question is flagged as a duplicate, but other users do not agree, and the label is removed again, or changed into a related question label. In CQADupStack, this happened 329 times out of the 24455 times a question received a duplicate label.[4] That is, for 98.7% of duplicate questions, there is implicit community agreement on the fact that two questions are duplicates. It is therefore not surprising that the annotator was in 598 out of 600 cases convinced of the correctness of the existing labels. Since the forums are dynamic environments, the two remaining duplicate labels which the annotator did not agree with may still be disputed by other community members and be removed. One of them (from the wordpress subforum) seems to be a genuine mistake (which persists in the live Stack Exchange data). The other one (from the android subforum) can be explained by looking at the answers: a question relating to adding account details in the Google Market app is marked as a duplicate of a question with details of how to delete and reinstall the Google Market app, implicitly suggesting that the answer is to delete and reinstall the app (despite the specific question being quite different, as seen above).

4. ESTIMATING THE LABEL RECALL

Our focus in the previous section was estimation of label precision, with the expectation that, given the biased sampling of question pairs, recall would be 100%. In practice, even after giving our annotator access to the original labels,[5] 1% of the randomly-selected *non-duplicate* question pairs were identified as *duplicates* (FNs in Table 1). Because the sample is small, we calculated a confidence interval using a bootstrap test to see how far we can generalise this 1% to the full collection. At a 95% confidence level, the confidence interval is $[0.0, 0.03]$, indicating that there may be no significant number of duplicates in the collection, or alternatively there could be well over 100 million pairs, with the expectation that the true number lies somewhere between these two extremes.

Manual annotation of all the question pairs that are currently not labelled as duplicates is infeasible, but we can still estimate the number of missing labels, and propose methods to reduce the annotation effort needed to uncover as many of them as possible. In this section we aim to do exactly that: to more systematically measure the label recall, and also suggest how we could increase the label recall of the dataset with relatively little manual annotation effort.

4.1 Label completeness

The data in CQADupStack is chronologically ordered, and this time element is preserved in the predefined splits. For any given question, only questions that have been posted earlier in time can be retrieved as duplicates. A natural consequence of this is that duplicate labels are unidirectional. If multiple questions are labelled as duplicates of a newer one, however, an interesting question arises: if questions A, B and C are posted one after the other — with A being the oldest, and C the newest — and question C is a duplicate of questions A and B, must question B also be a duplicate of question A? That is, are connected sub-graphs complete under the duplicate relation?

An analysis of the completeness of connected sub-graphs,

[4] 24455 is the number of times a question in CQADupStack received a duplicate label. However, in Section 4.2 we mention that CQADupStack contains 7040 questions with a duplicate label. This second number includes only the questions in the test and development set of the retrieval splits that are predefined in CQADupStack, because those are the only ones that are relevant to making the set a true gold standard. Table 4 provides an overview of these corpus statistics.

[5] On the forums, this is exactly the setup that users work with. One user flags a question as a duplicate of another one. Other users can see this and can indicate whether they agree with it or not.

and more specifically the number of *incomplete* graphs, provides us with a means to estimate, and ultimately increase, the recall. To analyse this, we identify questions which have been labelled as the duplicate of multiple other questions, and recursively form a "question cluster" from it in the form of a connected sub-graph. For each such cluster, we then exhaustively generate all question pairs.

Table 2 shows the results of this analysis. Based on the original duplicate labels, for only 212 out of the 1358 question clusters (15.6%) does completeness hold. These clusters generate 12763 potential duplicate pairs. The `webmasters` subforum in particular has a handful of questions with many duplicates: one with 54 duplicates and one with 106(!) duplicates.[6] These two questions are responsible for generating 6996 (out of a total of 8066) candidate duplicate question pairs. If we leave these out, only 1070 remain, resulting in a total of 5767 for all subforums (instead of 12763), of which 3176 (55.1%) are labelled as duplicates, leaving 2591 potential duplicate pairs. We manually inspected 112 pairs selected randomly from this set, and annotated them with a three-level annotation scheme ("duplicate", "not duplicate", and "possible duplicate", with the final class reserved for cases where the annotator lacked the domain expertise to definitely say that there was no duplication). This revealed that around 68% are missing a duplicate label, 21% are not, and the remaining 11% should possibly have one.

The question pairs that should not have a duplicate label are clearly related, but ask for slightly different things. Here is an example from the `mathematica` subforum:

> Q1: How to plot an ellipse? I'm new to Mathematica, and I'm finding it difficult to plot an ellipse. [...]

> Q2: Coloring a shape according to a function. I would like to create a 2D shape, say an ellipse, where each point in the ellipse is colored according a RGB function, [...]

4.2 Analysis of likely false negatives

The second strategy we used to try to identify missing duplicate labels consists of indexing the training question data from `CQADupStack`, and for each test question, retrieving the top-N relevant questions in the corpus, using three different ranking functions. We then take the intersection of the three result sets. The intuition behind this method is to maximise the likelihood of a given document in the result set being a duplicate of the query. We could add more ranking functions to increase this likelihood even further.

The retrieval methods we used were TF-IDF, BM25 [16], and a language model with Dirichlet smoothing (LMD) [14, 13]. We used question titles and descriptions as queries, and full threads (questions and all their answers) as indexed documents. The motivation for this was to minimise the lexical gap problem that exists between many similar questions. The documents we retrieve are therefore questions (plus their answers), rather than answers on their own.

We first looked at the retrieval results for queries that have duplicates in the original dataset, to see how many of these duplicates are retrieved in the top-N results. This is meant as a sanity check to confirm that at least some of the duplicates are retrieved using our method. Table 3 lists the average intersection of the top 1, 5 and 10 rankings

[6]These are both examples of the wiki-style duplicate question clusters mentioned in Section 3.

	Question clusters (complete/clusters)	Question pairs (dup/pairs)
`android`	8/81	226/592
`english`	69/364	828/1155
`gaming`	12/65	150/222
`gis`	2/17	36/51
`mathematica`	7/96	206/303
`physics`	20/193	415/597
`programmers`	42/173	424/601
`stats`	3/25	54/78
`tex`	27/238	520/777
`unix`	19/66	151/198
`webmasters`	0/20	114/8066
`wordpress`	3/20	52/123
TOTAL	212/1358	3176/12763

Table 2: An overview of the completeness of question clusters for each subforum. The question cluster column signifies the number of questions which are duplicates of multiple questions ("clusters"), and the number of these where all question pairs are labelled as duplicates ("complete"). This definition of *question cluster* is the same as in Section 4.1. The question pair column signifies the number of question pairs across all question clusters ("pairs"), and the number of these which are labelled as duplicates ("dup").

of the retrieval methods for these queries, and the average percentage of duplicates in that intersection. For example, for the queries in the `android` subforum, on average 3.26 of the returned results in the top 5 are the same for the three ranking functions. On average, 6.1% of these 3.26 questions are labelled as duplicates. Consequently, 93.9% of the intersecting results are not labelled as a duplicate.

There is a high degree of overlap in the top results of the three methods: 55% in the top 1 (0.55/1), 65% in the top 3 (3.28/5), and 68.5% in the top 10 (6.85/10). Interestingly, the table also shows that a large fraction of this overlap consists of non-duplicates. Most of the duplicate percentages in parentheses in Table 3 are quite low, signifying that the percentages of non-duplicates are high. This can be explained by the fact that, on average, questions with duplicates have only fractionally more than 1 duplicate, meaning that a ranking of ten documents will usually contain many non-duplicate questions. However, it also means that there is a large pool of unlabelled potential duplicates for us to explore.

In the remainder of our analysis of top-ranked documents, we took all of the queries into consideration, whether they were labelled as having a duplicate in the original dataset or not. Out of a total of 43133 queries, 24732 had a non-empty result set under intersection, and 24565 of those documents in rank 1 did not have a duplicate label.

There were some problematic cases where a certain extremely long question ended up at the top of the ranking for many queries. For the `wordpress` subforum, for instance, 3869 out of the 5659 queries have the same question ranked at the top for all three retrieval methods. It is unlikely

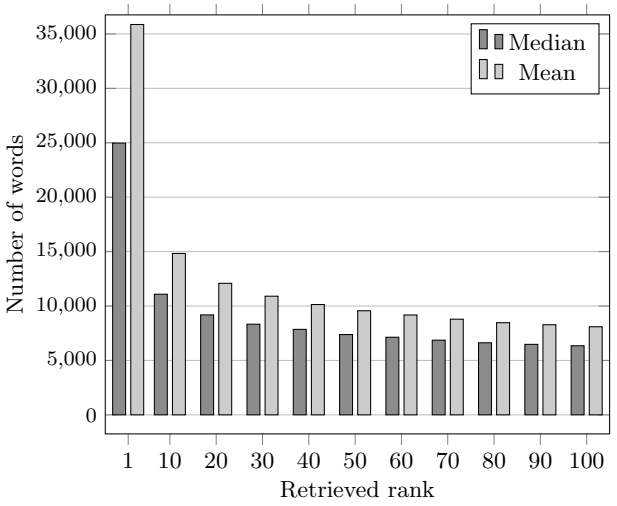

Figure 1: The mean and median length of questions retrieved at different ranks using BM25. The other two retrieval models we used (TF-IDF and a language model with Dirichlet smoothing) showed the same trend.

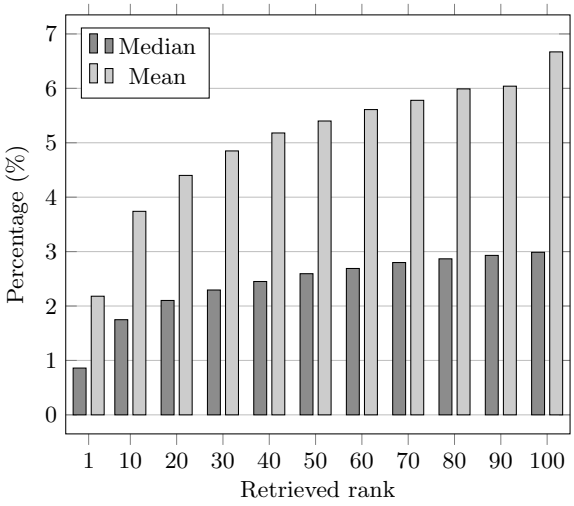

Figure 2: The query length compared to the retrieved document length of questions retrieved at different ranks using BM25. 100% means the query and retrieved document are the same length. 20% means the retrieved document is five times longer than the query. The lower the number, the bigger the difference in length between the query and the retrieved documents. The other two retrieval models we used (TF-IDF and a language model with Dirichlet smoothing) showed the same trend.

that this question really is a duplicate of all these queries. Figure 1 shows that long questions are likely to end up in higher ranks. The problematic question was between 3.5 and 4 times longer than the average question in rank 1. Figure 2 shows that the difference in length between the query and the retrieved documents is highest in the top ranks, and decreases further down the ranking. This shows that the long retrieved documents in the top ranks cannot be explained by the length of the queries. We decided to filter out results that appeared at the top of the ranking for more than 50 queries, in an attempt to mitigate this effect of question length. In this way we could remove 15188 potential duplicate pairs, leaving us with 9377 candidates. A manual evaluation of a random subset of 100 of these pairs resulted in an estimate of 15% being actual potential duplicates, translating into an additional 1407 duplicates.

When performing the same manual analysis for the top 5 rather than the top 1 results, we found a slightly lower percentage of duplicates (8%), which translates to an estimate of 2743 additional duplicates. The cost for finding these is high, however: 42489 more pairs need to be examined to find them, which is beyond the realistic bounds of our annotation resource. We therefore propose to limit ourselves to the top 1 rankings.

The strategy above identifies 9377 question pairs to examine, of which 1407 (15%) are estimated to be duplicates. Likewise, the graph completeness experiment in Section 4.1 finds 2431 question pairs to examine, of which 1730 are estimated to be duplicates (68% plus 11%). In total, therefore, we expect to be able to identify around 3137 new duplicate question pairs on the basis of annotating 11,809 question pairs. `CQADupStack` currently contains 3,916,532,896 question pairs, of which 7040 have a duplicate label. This means that we could potentially increase the number of duplicates by around 45%, by annotating only 0.0003% of all the question pairs in the set (11,809/3,916,532,896). The proportion of false negatives in Table 1 suggests that up to another

	top1	top5	top10
`android`	0.54 (18.5)	3.26 (6.1)	6.66 (3.8)
`english`	0.50 (17.6)	3.01 (7.3)	6.54 (4.1)
`gaming`	0.60 (37.0)	3.24 (12.3)	6.49 (6.2)
`gis`	0.60 (20.6)	3.53 (7.8)	7.10 (5.0)
`mathematica`	0.67 (2.4)	3.41 (1.4)	7.02 (1.6)
`physics`	0.59 (11.8)	3.25 (6.3)	6.78 (4.4)
`programmers`	0.83 (0.0)	2.67 (0.0)	5.83 (0.0)
`stats`	0.57 (17.1)	2.98 (7.1)	6.50 (3.8)
`tex`	0.47 (5.0)	3.45 (2.3)	7.18 (1.5)
`unix`	0.66 (9.6)	3.32 (4.3)	6.80 (2.6)
`webmasters`	0.61 (13.3)	3.47 (5.6)	7.04 (3.6)
`wordpress`	0.80 (5.9)	3.43 (2.2)	6.88 (2.1)
AVERAGE	0.55 (12.4)	3.28 (5.0)	6.85 (3.1)

Table 3: The average intersection (in absolute numbers) of the results in the top 1, 5 and 10 of three different retrieval methods (BM25, TF-IDF and a language model with Dirichlet smoothing); the percentage of duplicates is given in parentheses.

1% of question pairs (or 39,165,329) may be missing a duplicate label. Annotating 3.9 billion question pairs to find these is infeasible. We therefore need to find more ways to reduce the number of annotations needed to find these and hope to do that in future work. Table 4 recaps the numbers presented above.

5. DISCUSSION AND FUTURE WORK

Nr of threads/questions	470,344
Nr of test and dev questions	116,544 (25%)
Nr of indexed questions	353,800 (75%)
Nr of test and dev question pairs	3,916,532,896
Nr of test and dev pairs with dup label	7,040 (0.0001%)
Nr of test and dev pairs without dup label	3,916,525,856 (99.9998%)
Nr of pairs with dup label in full set	24045
Estimated missing dups in test and dev	39,165,329 (1% of all pairs)
Estimated nr of new dups in test and dev	3,137 (45% increase over 7,040)
- via transitive closure	1,730 / 2,190 (79%)
- via false negatives (top 1)	1,407 / 9,377 (15%)
- via false negatives (top 5)	4,150 (1,407 + 2,743) / 51,866 (9,377 + 42,489) (8%)
Nr of annotations needed	11,809 (0.0003% of 3,916,532,896)

Table 4: Descriptive statistics of `CQADupStack` and the estimates derived from the experiments in this paper. To make `CQADupStack` a gold standard, our aim is to identify the missing duplicate labels for the questions in the predefined test and development splits that come with the dataset, rather than *all* the missing labels.

In this paper we analysed the precision and recall of the duplicate question labels in `CQADupStack` with the goal of determining how close to a gold standard for question retrieval research it is. We have shown that the precision is high; in other words, the existing duplicate labels are very reliable. At the same time we have uncovered a problem with the recall and estimated that a little over 1% of the question pairs in the dataset lack a needed duplicate question label. This corresponds to around 39 million question pairs out of 3.9 billion. We can conclude from this that the current version of CQADupStack is a silver standard rather than a gold one, in terms of the quality of the relevance judgements.

To work towards making it (more) gold, we presented two methods to increase the number of labelled duplicate question pairs by around 45%, through manually annotating only 0.0003% of the question pairs in the data set. This number was estimated by identifying question clusters and annotating a portion the questions without a duplicate label, that would complete the transitive closure of these clusters. Additionally we used three well established retrieval methods and looked at the intersection of the results in the top 1 returned by all of them. Again we annotated a sample of the returned questions that did not have a duplicate label yet, to form an estimate of how many of these should have a duplicate question label.

In future work we would like to engage the Stack Exchange community in the annotation of the likely duplicate candidates we have identified. As we have established in this study, the Stack Exchange users have the necessary knowledge for this task, and would therefore be ideal annotators.

6. ACKNOWLEDGMENTS

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

7. REFERENCES

[1] O. Alonso, D. E. Rose, and B. Stewart. Crowdsourcing for Relevance Evaluation. *ACM SIGIR Forum*, 42(2):9–15, 2008.

[2] P. Bailey, N. Craswell, and D. Hawking. Engineering a Multi-Purpose Test Collection for Web Retrieval Experiments. *Inf Process Manag*, 39(6):853–871, 2003.

[3] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance Assessment: Are Judges Exchangeable and Does it Matter. In *Proc. 31st SIGIR*, pages 667–674, 2008.

[4] B. Carterette, J. Allan, and R. Sitaraman. Minimal Test Collections for Retrieval Evaluation. In *Proc. 29th SIGIR*, pages 268–275, 2006.

[5] B. Carterette and I. Soboroff. The Effect of Assessor Error on IR System Evaluation. In *Proc. 33rd SIGIR*, pages 539–546, 2010.

[6] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Education and Psychological Measurement*, 20(1):37–46, 1960.

[7] C. Grady and M. Lease. Crowdsourcing Document Relevance Assessment with Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 172–179, 2010.

[8] D. Hoogeveen, K. M. Verspoor, and T. Baldwin. CQADupStack: A Benchmark Data Set for Community Question-Answering Research. In *Proc. 20th ADCS*, page 3, 2015.

[9] M. Hosseini, I. J. Cox, N. Milić-Frayling, G. Kazai, and V. Vinay. On Aggregating Labels from Multiple Crowd Workers to Infer Relevance of Documents. In *Advances in Information Retrieval*, pages 182–194. Springer, 2012.

[10] K. A. Kinney, S. B. Huffman, and J. Zhai. How Evaluator Domain Expertise Affects Search Result Relevance Judgments. In *Proc. 17th CIKM*, pages 591–598, 2008.

[11] C. Macdonald and I. Ounis. The TREC Blogs06 Collection: Creating and Analysing a Blog Test Collection. *Department of Computer Science, University of Glasgow Tech Report TR-2006-224*, 1:3–1, 2006.

[12] S. Nowak and S. Rüger. How Reliable are Annotations

via Crowdsourcing: a Study about Inter-Annotator Agreement for Multi-Label Image Annotation. In *Proc. ICMR*, pages 557–566, 2010.

[13] J. M. Ponte. *A Language Modeling Approach to Information Retrieval*. PhD thesis, University of Massachusetts Amherst, 1998.

[14] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *Proc. 21st SIGIR*, pages 275–281, 1998.

[15] A. Ritchie, S. Teufel, and S. Robertson. Creating a Test Collection for Citation-Based IR Experiments. In *Proc. NAACL*, pages 391–398, 2006.

[16] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at TREC-3. *TREC-3*, pages 109–126, 1995.

[17] I. Soboroff, C. Nicholas, and P. Cahan. Ranking Retrieval Systems Without Relevance Judgments. In *Proc. 24th SIGIR*, pages 66–73, 2001.

[18] I. Soboroff and S. Robertson. Building a Filtering Test Collection for TREC 2002. In *Proc. 26th SIGIR*, pages 243–250, 2003.

[19] E. M. Voorhees and D. M. Tice. Building a Question Answering Test Collection. In *Proc. 23rd SIGIR*, pages 200–207, 2000.