Semi-Automated Prevention and Curation of Duplicate Content in Social Support Systems

Igor A. Podgorny
Intuit, Inc.
San Diego, USA
igor_podgorny@intuit.com

Chris Gielow
Intuit, Inc.
San Diego, USA
chris_gielow@intuit.com

ABSTRACT
TurboTax AnswerXchange is a popular social Q&A system supporting users working on U.S. federal and state tax returns. Based on a custom-built duplicate scoring model, 35% of AnswerXchange questions have been found to be near-duplicates responsible for 56% of AnswerXchange document views. This degrades the user experience for both the asker who is unable to find an answer amid duplicates, and the answerer who is unable to efficiently answer at scale. The duplicate questions tend to form micro-clusters that grow via preferential attachment and, once exceeding some 25 questions in size, start morphing into mega-clusters with a complex network topology. This behavior can be leveraged to design semi-automated content curation systems to detect whether a newly posted question is a duplicate and, if so, which duplicate cluster it belongs to. In order to improve user experience in AnswerXchange, we explore how human and artificial intelligence can be jointly employed and then present several data-driven intelligent user interfaces. The duplicate scoring models can be utilized as elements of question-posting and answering experiences, unanswered question queueing and answer bots. These approaches can be extended to any social support Q&A system where duplicate posting negatively impacts search relevance and content consumption.

Author Keywords
TurboTax; AnswerXchange; CQA; community question answering; social question answering; duplicate clusters; content deduplication.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
Social Q&A systems provide a convenient self-support option for tax and financial software applications where personalized long-tail content generated by the users can supplement curated knowledge base answers. Users often prefer self-help to assisted measures (e.g. phone support or online chat) and are often able to find and apply their solution faster. This also reduces the load on assisted channels, ensuring they remain available to those who need it. AnswerXchange (http://ttlc.intuit.com) is a social Q&A site where customers can learn and share their knowledge with other TurboTax customers while preparing U.S. federal and state tax returns and also find step-by-step instructions on using the TurboTax application [5, 6]. As the users step through the TurboTax interview pages, they can ask questions about software and tax topics (Figure 1) and receive answers in a matter of minutes. AnswerXchange has generated millions of questions and answers that have helped tens of millions of TurboTax customers since launching in 2007.

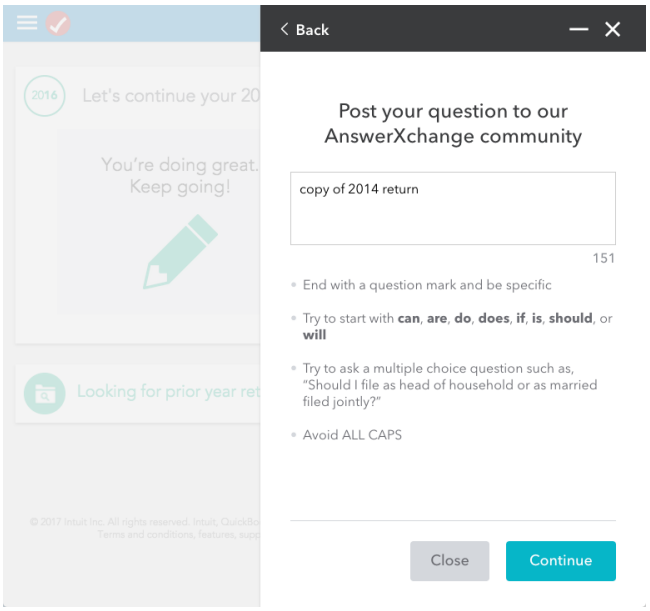

Figure 1. AnswerXchange question-posting user experience. Question title (a short summary of question limited to 255 characters) is mandatory. Question details (not shown) are optional and unlimited in size.

The majority of users can find answers by searching the existing content. The overall quality of a customer self-help system is therefore determined by how well the self-help system assists in finding the relevant content. The number of search sessions resulting in assisted support contacts (being as large as hundreds of thousands of customers per year) and fraction of user up or down votes on self-support content provide a convenient proxy metrics of content quality and search relevance in TurboTax self-help [5].

ESIDA'18, March 11, Tokyo, Japan.

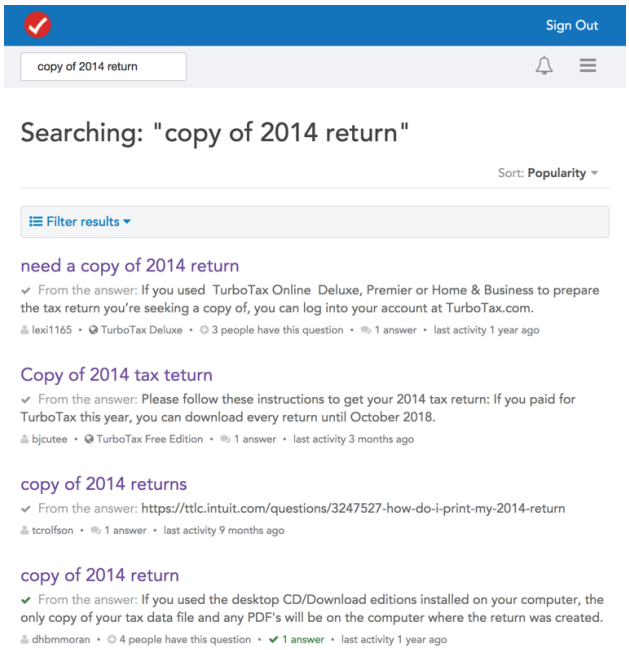

Figure 2. An example of duplicate AnswerXchange search results. Question titles and answer snippets are shown in purple and in black, respectively.

One problem with the existing question-posting experience (Figure 1) is that searches may result in multiple and often duplicate answers that are relatively close to the intent of the original question, but still do not match the original search intent (Figure 2). This interferes with the user's ability to select from a diverse set of possible answers [5] and, often results either in the submission of a duplicate question or switching to a less-desired support channel. A related problem is that users may submit poor quality questions by not providing all of the relevant information needed for a good quality answer [5]. One solution is a manual review of the user generated content to archive some of the duplicate questions and related answers, if any, and keeping the best performing content in "live" status (i.e. making it available for search). This approach is labor intensive and does not address the problem with the question-posting user experience. Duplicate questions may quickly build up, adding unnecessary burden on community question answering along the way.

The goal of this study is to address the problems of duplicate content prevention in AnswerXchange by combining machine learning and intelligent user interfaces. In what follows, we describe duplicate detection algorithms developed earlier and present a custom model trained on AnswerXchange questions. Next, we introduce the concept of "duplicate clusters" that provide a framework for semi-automated duplicate content prevention. Finally, we present several custom designed data-driven intelligent user interfaces for addressing duplicate content problem.

RELATED WORK

The task of estimating semantic similarity of text documents has multiple practical applications and is of growing interest from the research community. The areas of research include web page similarity, document similarity, sentence similarity, search query similarity and utterance similarity in conversational user interfaces. These tasks are also related to a more general problem of detecting duplicates in database records [2].

Questions in social Q&A systems media are often confined to one or two relatively short sentences and may warrant domain specific approaches to addressing question similarity. For example, two questions in a social Q&A system can be considered semantically identical if a single answer satisfies the needs of both original askers [3]. The answer may not yet exist in the production database but could be generated if needed. The task of duplicate-question detection is also related to the task of re-formulating a newly formed question [6] and automatically finding an answer to a new question [8].

The most recent results in the area of duplicate content scoring came from the 2017 Kaggle "Quora Pair" competition with model submissions from more than 3,000 teams (https://www.kaggle.com/c/quora-question-pairs). In this competition, the participants were tasked to classify if Quora question pairs are duplicates or not based on 200,000 training instances. Finally, SemEval2017 Task on Community Question Answering ("Question–Comment Similarity", "Question–Question Similarity", etc.) resulted in submissions from 23 teams [4].

The problem of duplicate detection and curation is closely related to the task of predicting content quality in social Q&A systems. Content quality metrics may be helpful in selecting the best performing question and answer for the duplicate-question pair. Answer and question quality in the social Q&A systems has been the focus of increasing attention from the scientific community [1, 9].

DUPLICATE-SCORING MODEL

AnswerXchange Search

AnswerXchange search is built with Apache Lucene open-source software (http://lucene.apache.org). By default, Lucene uses "tf-idf" (https://en.wikipedia.org/wiki/tf-idf) and "cosine-similarity" as standard methods of ranking search results. Shorter documents with the same set of matching keywords typically rank higher than longer documents with similar semantic meaning. An average AnswerXchange search query is 2-3 terms long (i.e. shorter than a typical AnswerXchange question) and it is often comparable in length with the title of a potentially duplicate question. The question details play a lesser role compared to titles contributing to extra boosting of duplicate content by Lucene. The AnswerXchange Lucene ranking algorithm tends to boost new content and also accounts for various metadata such as helpfulness votes.

Training Data

The problem of near-duplicate detection can be formulated as an unsupervised or supervised machine learning task [7]. In the unsupervised case, duplicate pairs and clusters can be found based on distance metrics such as cosine-similarity of the weighted tf-idf vectors, Jaccard similarity coefficient, distance in word2vec space, etc. In the supervised case, the problem of finding topical near-duplicate relations can be formulated as follows: given a pair of questions, the machine learnt model has to predict a "duplicate score" and determine if questions are duplicates based on a pre-defined threshold. In this paper, we employ a "hybrid" approach starting with cosine-similarity metrics for data pre-processing and then adding a more accurate custom-built scoring model to the processing pipeline.

As the fraction of duplicate pairs in AnswerXchange is relatively low, the question pairs ranked by cosine-similarity provide a convenient data set for labeling based on the importance sampling approach. Towards this goal, we computed bag-of-words cosine-similarity (Appendix A) for 790,000 questions available for search in AnswerXchange at the end of 2017 U.S. Tax Day (April 18). Next, four AnswerXchange moderators added class labels (0 or 1) to a random sample of 4,000 near-duplicate pairs. Instances open to doubt have been flagged by moderators and then re-labeled by a consensus. 1,000 randomly sampled non-duplicate pairs have been added for the final version of the training data set to make it equally divided between duplicate and non-duplicate pairs.

Duplicate-Scoring Model Features

The model features can be learnt from training data and/or by knowledge acquisition from AnswerXchange moderators. We have used the following model features:

- Cosine-similarity with tf-idf weighting (see Appendix A).
- Probabilistic topic ID of the question computed with Latent Dirichlet Allocation (see Appendix A).
- U.S. tax year in the question.
- Distinct words in the question pair.
- Common words in the question pair.
- Type of the question (e.g. "closed-ended" questions "Can I deduct …?" typically account for tax related, while "how" questions often account for product related question).
- First word of the question.

Duplicate-Scoring Model Performance

Based on the set of 5,000 labeled question pairs, we trained and tested a linear (logistic regression) and non-linear (random forest) binary classifiers using Python machine learning library "scikit-learn". The model predicts class label (0 for a non-duplicate and 1 for duplicate pair) and also the duplicate score (i.e. probability of the question pair to belong to either class ranging from 0.0 to 1.0) that can be used to select user experience based on predefined threshold(s). We also trained a separate version of the logistic regression classifier using cosine-similarity as a single model feature. Shown in Table 1 are common metrics used for predictive model evaluation: area under curve (AUC) for receiver operating characteristic, F1 score and logarithmic loss (log loss) function for classification.

Model	AUC	F1 Score	Log Loss
Logistic Regression	0.95	0.88	0.27
Random Forest	0.94	0.87	0.31
Cosine-similarity	0.83	0.73	0.48

Table 1. Model performance metrics for duplicate-scoring models (details are explained in the text).

As seen from Table 1, both logistic regression and random forest models achieve performance that is consistent with the goals of this exploratory study. At the same time, cosine-similarity version underperforms the first two by a wide margin. This can be explained by the inability to find an optimal threshold separating duplicate and non-duplicate pairs using the cosine-similarity alone. The following two examples illustrate the relationship between keyword-based cosine-similarity and duplicate-question score computed with logistic regression.

The first example is an AnswerXchange question pair with a relatively low cosine-similarity of 0.61: (1) "I need a copy of my federal tax return for 2014" and (2) "I need 2015 Tax Return". Both questions can be answered with a single instruction about getting a copy of prior year tax return filed with TurboTax and hence are duplicates. The second example is a question pair with high cosine-similarity of 1.0: (1) "do i have to file state taxes?" and (2) "how to file state taxes". These questions are not duplicates because they belong to tax and product categories [5], respectively, and would require two different answers.

DUPLICATE CLUSTERS

Preferential Attachment and Topology

After identifying 5,597,799 duplicate question pairs in AnswerXchange (Appendix A), we built an undirected graph of 281,031 duplicate questions. Each duplicate pair and duplicate question identified with the model constituted graph edge and graph vertex, respectively. The resulting graph consists of 14,616 connected components hereafter referred to as "duplicate clusters." To explore duplicate-cluster scaling behavior, we ranked clusters by the number of questions and plotted the number of questions per cluster vs. cluster rank in log-log scale (Figure 3). The largest cluster has 23,236 questions and the smallest ones only have two. The plot also includes graph (or edge) density:

$$D = 2E/V(V-1),$$

where E is number of edges (i.e. duplicate pairs) and V is the number of vertices (i.e. questions). Graph density is

equal to 1.0 for the fully connected graphs. In the latter case, each question in the cluster is connected to all remaining questions in the same duplicate cluster. Based on both question counts and graph density, the duplicate clusters in Figure 3 can be divided into three distinct groups marked as mega-clusters, transitional clusters and micro-clusters. These groups account for 84%, 2% and 14% of duplicate questions, respectively.

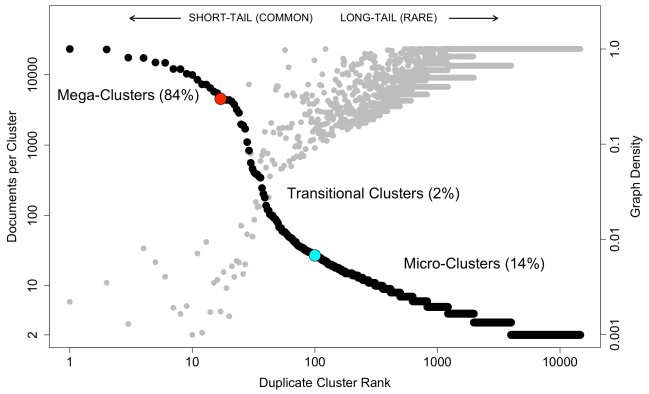

Figure 3. Scaling behavior of duplicate clusters (black dots) in AnswerXchange questions. The clusters are ranked by the number of questions in the descending order. Graph density for the clusters is shown in gray. Cyan and red dots refer to the clusters shown in Figures 4 and 5, respectively.

An example of micro-cluster with 23 vertices is shown in Figure 4. Graph density is 0.54 and most of vertices are interconnected with an exception of three vertices connected by bridges to a denser graph core. The corresponding articulation points are marked by blue dots. Note that even if questions 1 and 2 are duplicates and questions 2 and 3 are duplicates, this does not mean that questions 1 and 3 are duplicates as well. This explains why a duplicate-cluster density is typically less than 1.0 unless the graph size is limited to two questions. As seen from Figure 3, micro-cluster scaling behavior follows Zipf distribution (https://en.wikipedia.org/wiki/zipf's_law):

$$n(r) = Nr^{-\alpha},$$

where r ranges from about 100 to the total number of clusters R. Accordingly, the growth of N (ΔN) and R (ΔR) would be constrained by the following equation:

$$\Delta N/N = \alpha\, \Delta R/R.$$

It is worth mentioning that Zipf distribution is an asymptotic case of a more general Yule-Simon distribution (https://en.wikipedia.org/wiki/Yule-Simon_distribution) typical for the preferential attachment process, meaning that a newly posted duplicate is more likely to become attached to the existing cluster than to form a new duplicate pair. The scaling parameter for the micro-clusters:

$$\alpha = \left.\log\big(n(r_1)\big) - \log\big(n(r_2)\big)\right/ \log(r_1) - \log(r_2)$$

can be estimated as 0.6. By extrapolating Zipf distribution to r=1 (that would correspond to a non-existing largest micro-cluster), one can estimate N value as 400. This value, however, is almost two orders of magnitude less than the number of questions in the top mega-cluster.

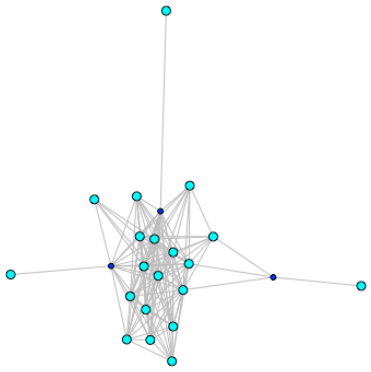

Figure 4. A micro-cluster marked by cyan dot in Figure 3. Articulation points are shown by smaller blue dots.

To explain the scale break in the distribution shown in Figure 3, let us examine larger duplicate clusters in more detail. Shown in Figure 5 is a mega-cluster with 4,549 questions. The cluster has density equal to 0.0017 and 1048 articulation points. This means that the mega-clusters may consist of multiple sub-clusters that are semantically related to each other but with the elements that are not duplicates unless they belong to the same sub-cluster.

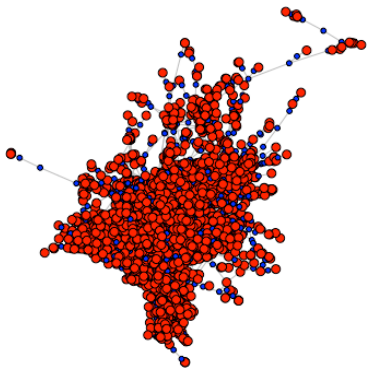

Figure 5. Same as in Figure 4, but now for a mega-cluster.

As the number of duplicates reaches certain level, the clusters start coalescing by establishing bridges with other clusters, duplicate pairs and stand-alone questions, quickly evolving from dense connected graphs to sparse graphs with a complex network topology. The area of transition is marked as transitional clusters in Figure 3.

Semi-Automated Duplicate Content Curation

While the task of duplicate content archiving is straightforward once duplicate pairs are found (Appendix A), the duplicate content can build up again unless question-posting and/or search experiences are modified.

Our next goal is therefore to explore how the concept of duplicate clusters discussed in the previous section can be applied to these tasks. The curation of micro-clusters can be done automatically or semi-automatically (i.e. with minimum human involvement) by retaining one or few best performing long-tail documents (i.e. documents that include both questions and answers) and assigning them a cluster ID for subsequent re-use.

The curation of mega-clusters represents a more challenging problem. First, a single best performing document in a mega-cluster may simply not exist since the cluster may contain multiple sub-clusters connected by bridges. Second, duplicate curation by a human is a cumbersome task due to the mega-cluster complex topology. While the exact solution may simply not exist, approximate solutions may be sufficient to reduce the number of duplicates posted in the AnswerXchange to an acceptable level. One approach would be to break the mega-clusters into smaller parts by deleting bridges in the graph or by employing a conventional hierarchical clustering. For example, the duplicate cluster shown in Figure 5 can be split to 1363 connected components by removing all articulation points (blue dots in Figure 5). Most of the resulting connected components, however, are disconnected documents.

A more practical approach is to archive non-performing short-tail content from the mega-cluster and curate the resulting connected components. Shown in Figure 6 is a subset of mega-cluster from Figure 5 that now only includes documents with at least 100 views. This results in breaking the original mega-cluster into 68 connected components which are easier to curate.

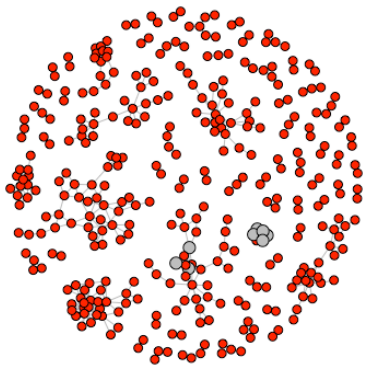

Figure 6. A subset of the mega-cluster shown in Figure 5. Grey dots mark documents used in Figure 7.

The next task is to present duplicate content in a form suitable for semi-automated content curation. Figure 7 shows an example of duplicate content metrics for eight documents with at least 1000 views. The left column is a sub-cluster ID followed by a post ID identifying an AnswerXchange document consisting of the original question and all accumulated answers (not shown). The text of the question and type of the question (i.e. user-generated content marked as UGC or knowledge base content labeled as FAQ) are included in the third and fourth columns, respectively. The last two columns are views accumulated over a given period and percentage of up-votes. The documents can be ranked by views and/or votes providing a mechanism of identifying and removing non-performing content either manually or automatically based on a set of predefined content quality thresholds.

ID	POST_ID	DOCUMENT	TYPE	VIEWS	UPVOTE
1	1,899,475	Can I deduct job-search expenses?	FAQ	17,019	74.8
1	2,666,148	HI. Where do I enter my job search	UGC	1,759	77.9
1	3,048,015	Where do I include job search	UGC	1,060	78.1
1	3,356,358	Where do I enter my job search	FAQ	6,727	70.3
1	3,705,028	Where do I deduct job search	UGC	2,999	67
2	2,895,188	Where do I enter my medical	FAQ	25,243	79.9
2	2,899,090	Why doesnt my refund change after I enter my medical expenses?	FAQ	13,765	79.1
2	2,956,890	where do i enter OUT OF POCKET medical expenses	UGC	1,509	86.6

Figure 7. Duplicate document metrics for the documents marked by grey dots in Figure 6.

Duplicate metrics can be operationalized by adding an algorithm to match the best question to the best answer in the sub-cluster. Such a system would include answer deleting and merging manually or automatically by attaching automatically generated "best" answer to the "best" duplicate question. The solution can be implemented as a back-end tool for trusted users assigned to the task of duplicate archiving and hidden from the less experienced regular users. The solution goes beyond simple duplicate archiving by providing an option to merge available answers to the existing duplicate questions. The non-human part of the solution includes quality ranking of the existing answers, e.g. up and down vote statistics as shown in Figure 7. In this way, the newly formed question-answer pairs provide better quality content available for search by combining the visually appealing questions and the best ranked answers. This is done by combining artificial and human intelligence since the answer to a related question (that the system recommended) can be confirmed by the contributor if needed. The cluster notes can be edited by trusted users and applied to all articles within the cluster.

Real Time Duplicate Detection

Finding duplicates to a given question requires (N-1) pairwise comparisons to the questions in the database and may be not feasible in real time. The computational time can be reduced by selecting potential duplicate matches with AnswerXchange search. The top performing documents in the clusters can be assigned an ID and indexed separately by the search engine. Once the search engine returns the documents ranked by relevancy to the newly formulated question, the duplicate-scoring model is applied to the top matches to see if the new question is a duplicate and, if so, which duplicate cluster it belongs to.

DATA-DRIVEN USER EXPERIENCES

Accumulation of duplicate content can be prevented by integrating a custom-built duplicate-scoring model and question-posting experience. Another option is to expose an intelligent interface to the trusted users by providing extra features for answering duplicate questions. Finally, the duplicate question curation can be part of the content moderation process carried out by the AnswerXchange trusted users or trained bots.

Question Deduplication While Posting

The first feature (Figure 8) extends the AnswerXchange "Question Optimizer" system [6]. The system prompts the asker with personalized instructions created dynamically based on real time analysis of the question's semantics and writing style. The "Question Optimizer" has been re-designed to make duplicate question more difficult to submit without addressing the recommended re-phrasing. The annotations to concept are presented next.

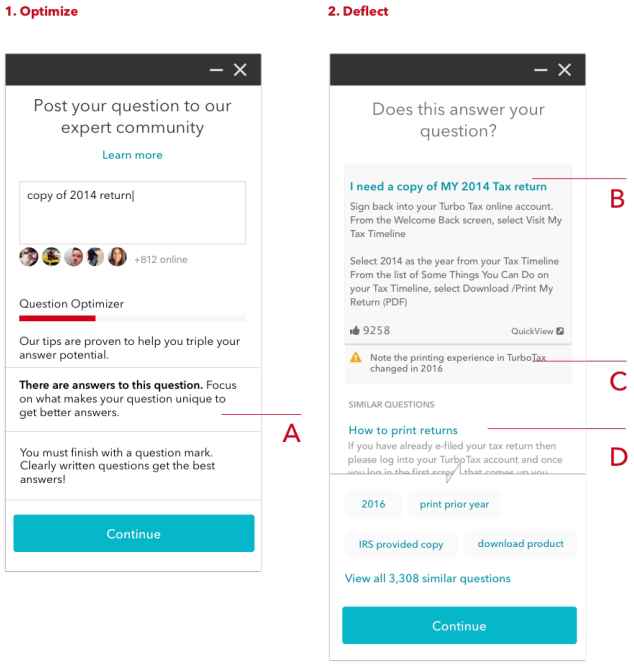

Figure 8. Question-posting experience reveals the duplicates and helps users re-phrase as a unique question.

A) The "Question-Optimizer" technology is envisioned to include duplicate content detection in addition to providing timely advice on how to re-phrase or deflect.

B) If question falls in a known duplicate cluster, the best matching and most referenced answer matches are shown.

C) Trusted users may attach "cluster notes" to curated duplicate clusters and appear automatically with any question within the cluster. In the example shown in Figure 8, the duplicate cluster is about printing and the message notes that the printing experience recently changed in the product - information which may be useful to anyone with printing-related questions.

D) The suggested answers are deduplicated using duplicate score equalization so the answers are more useful. A "cluster browser" is also added below to the results to help refine amongst the most popular variations.

Question Deduplication While Answering

The second feature addresses the situation where a potential duplicate has been submitted and needs to be intercepted as part of question answering experience. This concept is illustrated in Figures 9-10.

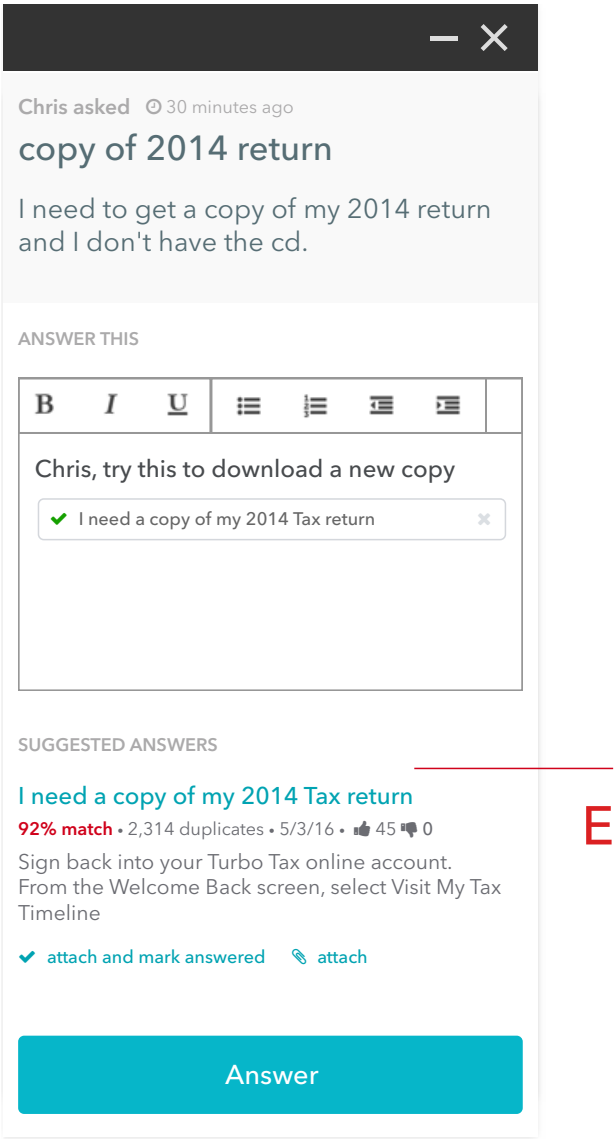

Figure 9. Contributor experience tagging and attaching curated answer to the question.

Specifically, Figure 9 illustrates the contributor (typically a trusted user) answering experience and includes the following annotation:

E) The suggested answered question duplicate is presented to the original asker and also displays the duplicate probability. The contributor can easily attach it to their answer, which also tells the system the question was a duplicate and should be archived in favor of the attached.

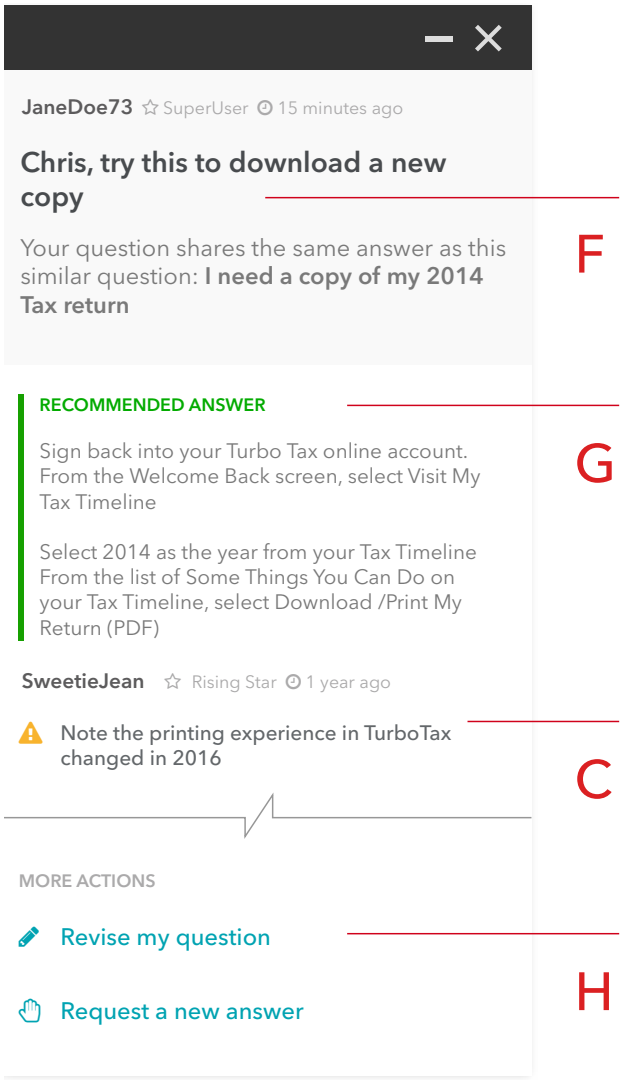

Figure 10. Original asker view of deduplicated question with personalized answer.

Once the duplicate question is answered it becomes available to the original asker (Figure 10).

C) Re-purposing trusted users notes similar to those used in question-posting experience (Figure 8).

F) A personalized note introduces the "recommended answer" while explaining it's a duplicate.

G) The duplicate answer is presented with a sense of authority.

H) If the original asker is unsatisfied with the answer, they may revise their question and it will re-enter the answer queue. They also have the option to request a new answer without submitting the question.

Finally, flagging the unanswered question automatically as a duplicate may be validated or invalidated by the trusted users and to update training dataset for model re-training.

Question Deduplication with Automated Answers

The "Answer Bot" (Figure 11) is a feature driven by artificial intelligence alone. The "Answer Bot" increases self-support efficiency by responding to a customer's questions by e-mail with answers from the matching duplicate cluster if the posted question is flagged by the duplicate-scoring model as a duplicate.

I) "Answer Bots" may automatically answer questions determined to be duplicates. Like the contributor-assisted experience, the bot will recommend the answer from the best answer within the duplicate cluster. The user is made aware that a bot answered the question, and if unsatisfied may request a new answer, or revise their question.

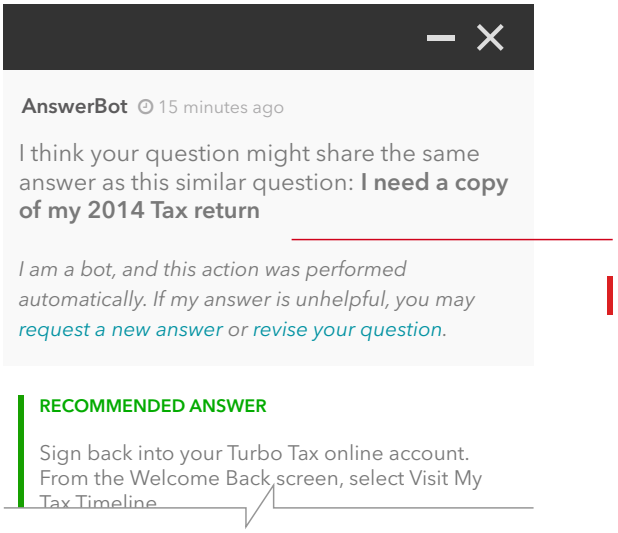

Figure 11. Automated deduplication user experience as part of customized e-mail to the original asker.

Further, the "Answer Bot" attaches the question to the existing duplicate cluster automatically while providing a generic or personalized answer. The bot replies trigger automated archiving of the duplicate content. The question remains visible to the original asker but is not made available to AnswerXchange users and is suppressed from search results. A related option is to create two separate queues of duplicate questions for answering. The questions in the first queue would be assigned to designated moderators who can customize duplicate content for the original asker and archive it afterwards. The less complicated questions in the second queue can be assigned to the "Answer Bot".

DISCUSSION AND CONCLUSION

Social Q&A systems often presume that the users comply with recommendations not to replicate the existing content. This is not the case for AnswerXchange where users often avoid consuming existing content by posting a new duplicate question. These users may not realize that AnswerXchange is a social Q&A site or lack the ability to find and apply existing answers to their question. We need to intervene with intelligent user interfaces to alter the duplicate posting behavior. Towards this goal, we present two algorithms for duplicate content curation and providing real time inputs to the AnswerXchange user interfaces. The first algorithm determines if two questions are near-duplicates and can be combined with a search to detect duplicates in real time. The second algorithm uncovers all duplicate pairs in AnswerXchange and is capable of handling deduplication task with a corpus of millions of questions. We conclude the paper by presenting three question deduplication user interfaces. Our hypothesis to validate include: (1) Will askers accept a duplicate when presented with an acceptable answer? (2) Will they accept a duplicate with or without a personalized contributor note? (3) If dissatisfied will they revise or request a new answer? (4) Will they accept recommended answers from Answer Bots? We are planning to validate these hypothesis with a set of rapid experiments prior to production.

APPENDIX A: DUPLICATE PAIR DETECTION

Detecting duplicates for N=790,000 questions based on a custom-built model would require (N(N-1)/2 pairwise computations. The task of finding duplicate pairs becomes computationally expensive once the corpus reaches several hundred thousand documents. At the same time, computing cosine-similarity for a question pair is faster than scoring the same pair with custom-built model and can be used to reduce the number of potential duplicate pairs from billions to millions of pairs. Further, dividing content by M probabilistic topics can reduce the number of pairwise comparisons by M, while not necessarily affecting the number of expected near-duplicate pairs.

M	Duplicates	Execution time (min)
50	63,355	13
30	72,920	18.5
10	73,068	36
1	83,773	265

Table A1. Duplicate statistics and computation time vs. number of probabilistic topics (M). Cosine-similarity threshold is 0.7. M=1 means processing N(N-1)/2 pairs.

Shown in Table A1 are results of the numerical experiments conducted on MacBook Pro laptop with 2.8 GHz processor speed. The processing pipeline included (1) dividing questions into M topics, (2) computing cosine-similarity for all pairs in a topic, and (3) applying duplicate-scoring model to the pairs with cosine-similarity above a pre-defined threshold. The total number of duplicate pairs was found to be 5,597,799 and contained 281,031 unique questions (or 35% of the AnswerXchange "live" questions). In 2017, they contributed 56% to the AnswerXchange document views. The documents in the identified duplicate pairs can be ranked by a suitable question (and answer) proxy content quality metrics as discussed earlier, for example by the number of views, votes, age of the post, or by a weighed combination thereof. The document with the lower score can be removed consecutively from each pair resulting in a removal of 217,767 documents (27% of the AnswerXchange "live" questions).

ACKNOWLEDGMENTS

We thank anonymous reviewers for valuable comments.

REFERENCES

1. Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, Gilad Mishne. 2008. Finding High-Quality Content in Social Media. In: *Proc. of the International Conference on Web Search and Data Mining*, 183-193.
2. Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, Vassilios S. Verykios. 2007. Duplicate Record Detection: A Survey. *IEEE Trans. Knowl. Data Eng*., 19, 1-16.
3. Klemens Muthmann, Alina Petrova. 2014. An automatic approach for identifying topical near-duplicate relations between questions from social media Q/A sites. In: *Classifying Big Data from the Web,* 1-6.
4. Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, Karin Verspoor. 2017. SemEval-2017 Task 3: Community Question Answering. In: *Proc. of the 11th Int. Workshop on Semantic Evaluation*, 27-48.
5. Igor A. Podgorny, Matthew Cannon, Todd Goodyear. 2015a. Pro-active detection of content quality in TurboTax AnswerXchange. In: *Proc. of ACM Conference Companion on CSCW*, 143-146.
6. Igor A. Podgorny, Chris Gielow, Matthew Cannon, Todd Goodyear. 2015b. Real time detection and intervention of poorly phrased questions. In *CHI'15 Extended Abstracts*, 2205-2210.
7. R. S. Ramya, K. R. Venugopal, S. S. Iyengar, L. Patnaik. 2016. Feature Extraction and Duplicate Detection for Text Mining: A Survey. *Global Journal of Computer Science and Technology* 56, 5.
8. Anna Shtok, Gideon Dror, Yoelle Maarek, Idan Szpektor. 2012. Learning from the Past: Answering New Questions with Past Answers, *WWW*, 759-768.
9. Ivan Srba, Mária Bieliková. 2016. A Comprehensive Survey and Classification of Approaches for Community Question Answering. In: *TWEB*, 10(3), 18:1-18:63.