On the Use of an Intermediate Class in Boolean Crowdsourced Relevance Annotations for Learning to Rank Comments

Alberto Barrón-Cedeño
Giovanni Da San Martino*
albarron@[hbku.edu.qa|gmail.com]
gmartino@hbku.edu.qa
Qatar Computing Research Institute, HBKU
Doha, Qatar

Simone Filice
filice.simone@gmail.com
Qatar Computing Research Institute, HBKU
Doha, Qatar

Alessandro Moschitti
amoschitti@hbku.edu.qa
Qatar Computing Research Institute, HBKU
Doha, Qatar

ABSTRACT

In many Information Retrieval tasks the boundary between classes is not well defined and assigning a document to a specific class may be complicated, even for humans. For instance, a document which is not directly related to the user's query may still contain relevant information. In this scenario, an option is to define an intermediate class collecting ambiguous instances. Yet some natural questions arise: is this annotation strategy convenient? How should the intermediate class be treated?

To answer these questions, we explored two community question answering datasets whose comments were originally annotated with three classes and re-annotated a subset of instances considering a binary *good* vs *bad* setting. Our main contribution is to show empirically that the inclusion of an intermediate class to assess Boolean relevance is not useful. Moreover, in case the data is already annotated with a 3-class strategy, the instances from the intermediate class can be safely removed at training time.

CCS CONCEPTS

•**Information systems** →**Relevance assessment;** Retrieval models and ranking; Learning to rank; *Question answering;*

KEYWORDS

relevance assessment, learning to rank, crowdsourcing, community question answering

ACM Reference format:
Alberto Barrón-Cedeño, Giovanni Da San Martino, Simone Filice, and Alessandro Moschitti. 2017. On the Use of an Intermediate Class in Boolean Crowdsourced Relevance Annotations for Learning to Rank Comments. In *Proceedings of SIGIR '17, Shinjuku, Tokyo, Japan, August 07-11, 2017,* 4 pages.
DOI: http://dx.doi.org/10.1145/3077136.3080763

*A. Barrón-Cedeño and G. Da San Martino contributed on pair to this work. They should be considered first authors.

SIGIR '17, Shinjuku, Tokyo, Japan

978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080763

1 INTRODUCTION

Explicit judgments are often necessary to build up a dataset to learn from. A standard mechanism to lift up the annotation quality is repeated labeling: having annotators judging the same instance more than once [1]. As this is costly, *crowdsourcing* is an alternative: it allows for the annotation of large amounts of data for training machine learning models at affordable rates in relatively short time [21]. The most straightforward strategy to annotate the relevance of a document for information retrieval (IR) is binary: the so called *turker* must decide whether a document is *relevant* or *irrelevant* given a query. A third label *unsure* can be added to allow for doubtful documents to be discarded from a training set or to define a different 3-ways model; although this is unrealistic in real-life scenarios.

The corpora of the three editions of the SemEval task on community question answering [16–18] include an intermediate label *potential*, which refers to answers which do not address directly the question, but are still considered useful. In this work we perform a thorough study on the usefulness of annotating such an IR dataset considering three levels of relevance instead of two by means of two complementary sets of experiments. On the left hand side, we try to exploit the additional information provided by the intermediate *potential* class in order to produce better rankings, where the relevant documents are positioned on the top positions of the ranking. On the right hand side, we re-annotate a subset of instances considering the standard binary relevant vs irrelevant setting and train ranking models to assess which annotation schema results in the best outcome.

Our results show that including an intermediate relevance level does not affect significantly the performance of the ranker. However, it effectively attracts instances which are not considered relevant by the model. Thus (*i*) if a corpus containing such intermediate class is at hand, discarding such instances from the beginning would allow for a faster training process, without harming performance and (*ii*) if a person is about to annotate a corpus for relevance, it is safe to do it considering only *relevant* versus *irrelevant*, which might result in a neater crowdsourcing task.

2 BACKGROUND

Different annotation schemes exist to manually assess the relevance of a pool of documents D given a query q. They have different levels of complexity and granularity and allow for different kinds of learning and evaluation.

Table 1: Class distribution in the CQA-QL-2015 and CQA-QL-2016 corpora.

	good	*potential*	*bad*
CQA-QL-2015			
train	8,069 (48.78%)	1,659 (10.03%)	6,813 (41.19%)
devel	875 (53.19%)	187 (11.37%)	583 (35.44%)
test	997 (50.46%)	167 (8.45%)	812 (41.09%)
CQA-QL-2016			
train	6,651 (37.16%)	3,110 (17.37%)	8,139 (45.47%)
devel	818 (33.52%)	413 (16.93%)	1,209 (49.55%)
test	1,329 (40.64%)	456 (13.95%)	1,485 (45.41%)

In the *ranking* schema, the annotator mimics the ranking model. Her task is ordering the documents in D according to their relevance against q. That is, no particular label is assigned to each document $d \in D$. This kind of annotation has been used in related tasks, such as the evaluation of machine translation [5]. Nevertheless, the cost of annotating under this schema is $O(n^2)$, with n being the size of D, as in the worst scenario each pair of documents in D has to be compared to come out with the final ranking.

In the *graded relevance* schema the annotator is shown q and one $d \in D$ at a time. In this case, the task consists of selecting one of k ordinal values from a Likert scale. For instance, d could be *highly relevant*, *fairly relevant*, *marginally relevant* or *irrelevant* with respect to q [11]. Other scales include more items whose extremes represent exactly the document the user is searching and spam or junk. This is the case of the crowdsourcing [20] and the federated search [6] tracks at TREC, which use scales of five and six elements, respectively. This strategy is in general simpler than the previous one and, alike it, allows for the evaluation of IR models by means of DCG-like metrics [12]. The cost of the annotation is $O(n)$.

In the *Boolean* schema the same pair $\{q, d\}$ is shown to the annotator. The task is simpler in this case: judging whether d is relevant with respect to q or not. Although the information obtained out of this schema is less expressive than in the former ones, it still allows for the modeling of the ranking problem with learning to rank and for the evaluation of models with standard metrics, such as MAP. This schema is applied in ad hoc retrieval and, due to its simplicity, it is used often when judging relevance by means of crowdsourcing [19]. As a consequence of the "hard case bias" [4], a good percentage of the noise in the annotation results from the harder-to-decide instances, which lie in between the two classes.

3 DATASETS ANNOTATION

We use the CQA-QL-2015 and the CQA-QL-2016 corpora [17, 18]. These datasets represent a benchmark for the evaluation of community question answering models. We focus in the comment ranking task: given a forum question q and its associated thread of comments D, rank the comments according to their relevance with respect to q. Unlike similar datasets in which the relevance of a comment is inferred from the forum users' judgments [13], in this case the relevance was judged by crowdsourcing. The labeling was made using an extension of the Boolean schema: *good* vs *bad*. An additional label *potential* was included as well. The crowdsourcing instructions of [18] include the following class definitions:

Table 2: Confusion matrix between the original 3-class annotation (left) and the new 2-class annotation (top) of the test partition of the CQA-QL-2016 corpus.

	good	*bad*
good	1,151 (92.23%)	97 (7.77%)
potential	222 (57.81%)	162 (42.19%)
bad	103 (7.37%)	1,295 (92.63%)

good at least one subquestion is directly answered by a portion of the comment;
bad no subquestion is answered and no useful information is provided (e.g., the answer is another question, a *thanks*, dialog with another user, a joke, irony, attack, or is not in English);
potential no subquestion is directly answered, but the comment gives potentially-useful information about one or more questions.

It is worth noting that *potential* does not play the same role as the commonly used *unsure* [7]. Indeed, the *potential* label is intended to attract the aforementioned hard-to-decide instances. Table 1 includes the class distribution in the two datasets.

[4] recommend to remove *potential* instances from the training and testing sets. Nevertheless, this is unrealistic, as in a real scenario we cannot skip an instance due to the difficulty to rank it. In order to study whether to discard *potential* instances —only from the training partition— is a good idea or if they should indeed be discarded, we performed a new Boolean annotation of a subset of the CQA-QL-2016 corpus.[1] We carried out the annotation using Crowdflower,[2] mimicking as best as possible the setting used by [18]. That is, the turkers observed one question and ten related comments, which they had to annotate as *good* or *bad*. In order to prevent bias, we removed the definition of *potential* rather than transferring it to either of the two labels. Different to the original annotation, which obtained five annotations per instance, we opted for requesting ten, as it has been observed that the more times an instance gets annotated, the higher the quality [10]. Our $400USD budget allowed us to annotate 93% of the instances in the test partition.

As expected, the instances with the highest agreement were those originally judged as *good* and *bad*. Concretely the average agreements were of 0.88, 0.73, and 0.90 for originally *good*, *potential*, and *bad* instances, respectively. Table 2 shows the confusion matrix between the original 3-class and the 2-class annotations. As observed in Section 5, for evaluation purposes [18] assumes that *potential* +*bad* instances represent the subset of *irrelevant* comments. Nevertheless, as our confusion matrix shows, the originally-judged as *potential* instances are spread across both *good* and *bad* subsets with a rough proportion of 60-40%.

4 RE-RANKING MODEL

In order to perform our experiments, we used the best-performing ranking model presented at SemEval 2016 Task 3-A [8]. It consists of a kernel-based SVM classifier which operates on question–comment pairs by adopting a combination of a linear kernel and a tree kernel. The classifier scores are used to rank the comments in the thread.

[1] This new annotation is available at http://alt.qcri.org/resources/cqa.
[2] https://www.crowdflower.com/

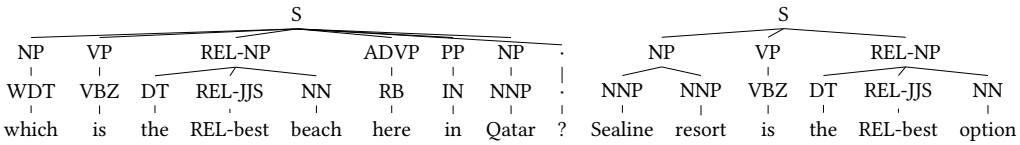

Figure 1: Structural representation of a question-answer pair.

The linear kernel is applied on vectors containing three groups of features:

*(*i*) Similarity features (sim).* Linguistic similarity scores between q and d. They include lexical similarities of lemmas, and syntactic similarities of PoS tags and parse trees. This subset includes semantic similarities between additive representations of word2vec embeddings [14] trained on the entire Qatar Living corpus from SemEval 2015[3]. Therefore, we do not use these features when dealing with the CQA-QL-2015 corpus.

*(*ii*) Heuristic features (heur).* They capture some comment characteristics such as its length, its forum category, whether it includes URLs, emails, particular words, etc.

*(*iii*) Thread-based features (thread).* As discussed in [3], comments in a common thread are strongly interconnected: users reply to each other and start a concrete discussions. These features aim at capturing some thread-level dependencies, such as whether a comment is part of a dialogue, or whether a comment is followed by an acknowledgment from the user who asked the question.

Tree kernels (TK) [15] are also applied to evaluate inter-pair similarities between $\{q, d\}$ pairs. As shown in Figure 1, a question-comment pair is represented by its corresponding shallow parse trees, where common or semantically similar lexical nodes are linked using a tagging strategy (which is propagated to their upper constituents), as proposed in [9]. This approach discriminates aligned sub-fragments from non-aligned ones, allowing the learning algorithm to capture relational patterns, e.g., *the* REL-*best beach* and *the* REL-*best option*. Given two q–d pairs $p_a = \langle q_1, d_1 \rangle$ and $p_b = \langle q_2, d_2 \rangle$, the following tree kernel combination is defined:

$$\text{PTK}^+(p_a, p_b) = \text{PTK}(q_1, q_2) + \text{PTK}(d_1, d_2) \quad (1)$$

where PTK is the Partial Tree Kernel [15].

5 EXPERIMENTS

We describe two sets of experiments: the first one compares 2-class against 3-class annotations; the second set of experiments refer to the 3-class-annotated dataset and aims at investigating the best use of the intermediate class during learning.

5.1 3- versus 2-Classes Annotations

Given that only the test has been re-annotated, in order to compare 2- versus 3-way annotations, we used 5-fold cross validation on the CQA-QL-2016 test set. While the gold labels of the test fold are always those binary annotated, the gold labels of each training fold in the three experiments are: (*i*) annotated with 2 classes; (*ii*) annotated with 3 classes with *potential* instances turned into *bad*; and (*iii*) annotated with 3 classes with *potential* instances removed. In all the experiments, we set kernel parameters to their default values; the C parameter of the SVM is set to 1.

[3]http://alt.qcri.org/semeval2015/task3

Table 3: 5-fold cross validation (mean±sd) on the 2016 test set. Training folds annotated with 2 or 3 classes. In the latter, *potential* examples are either discarded or changed to *bad*.

Train fold labels	MAP	Acc
3 classes, *potential* → *bad*	84.33 ± 1.9	72.08 ± 1.5
3 classes, *potential* discarded	85.01 ± 2.3	74.82 ± 1.2
2 classes	85.13 ± 2.6	75.48 ± 2.0

Table 3 shows the results. We performed a two-matched-samples t-test at the 0.05 level of significance between each pair of experiments. It showed that 3-class annotations do not significantly improve MAP nor accuracy. However, if the dataset is already annotated with three classes, an alternative is discarding *potential* instances. Besides speeding up the learning process, it gives better performance with respect to turning all *potential* into *bad*.

5.2 Exploiting *Potential* in 3-Class Annotations

Since re-annotating with binary classes the whole data can be an expensive procedure, we now study the effects of removing *potential* instances at training time on various scenarios in which less and less 2-class re-annotated data is available. In all the following, we will compare training without *potential* instances against turning them into *bad* ones.

In the first experiment on the CQA-QL-2016 corpus, we still consider the binary re-annotated test set, but we train the same model used in Section 5.1 on the training set: removing *potential* instances slightly decreases MAP (79.59 vs 79.64).

In the following experiments, we assume no re-annotation has been performed on the test set. Since the goal of the SemEval competition is to re-rank the comments in such a way that *good* ones are ranked higher than *potential* and *bad* ones, we focus on the binary learning task *good* vs {*potential* ∪*bad* }. In this experiment, removing *potential* instances brings only little improvement in MAP on the test set: 79.32 against 79.09. The results so far confirm that removing potential won't affect significantly the performances of the learning algorithm.

We perform a final experiment on the CQA-QL-2015 corpus. In this case, we use the features described in Section 4 but not the tree kernels. This is to give evidence that our hypothesis is not valid only for one type of kernels. Furthermore, it allows us to significantly reduce the training times. We perform a 5-cross fold validation on the union of the training, development and test sets. The task is again *good* vs {*potential* ∪ *bad* }, and we analyze the performance trend when the *potential* instances are gradually downweighted. Table 4 reports F_1 since it is the official performance measure used

Table 4: Mean±sd F_1 according to the weight given to the *potential* instances in the binary *good* vs. *rest* setting on the CQA-QL-2015 corpus.

w	F_1	w	F_1	w	F_1
0.0	73.99 ± 1.0	0.4	73.44 ± 1.2	0.8	73.31 ± 1.2
0.1	73.90 ± 1.0	0.5	73.38 ± 1.2	0.9	73.34 ± 1.2
0.2	73.75 ± 1.0	0.6	73.35 ± 1.2	1.0	73.38 ± 1.2
0.3	73.60 ± 1.2	0.7	73.33 ± 1.2		

in the SemEval 2015 competition. F_1 decreases from 73.99±0.9928 (weight = 0.0) to 73.38±1.2235 (weight = 1.0); which gives further evidence that discarding *potential* instances (weight = 0.0) on the training folds is beneficial.

6 CONCLUSIONS

An additional intermediate class can be considered for the Boolean annotation of instances to train learning-to-rank models; especially, when the annotation is crowdsourced. Such intermediate *potential* label is intended to attract edging instances.

In this paper we thoroughly studied the impact of adding this additional class. We took two community question answering datasets, whose answers were originally annotated by crowdsourcing with three classes: *good*, *potential*, and *bad*. Firstly, we re-annotated a fraction of the instances with a Boolean *good* vs *bad* setting. Secondly, we performed experiments in which *potential* instances are weighted in different ways —as important as the others, down-weighted, or fully discarded. Our results, also on the realistic *good* vs *bad* evaluation settings, show that the performance of the models exploiting or ignoring the information provided by the *potential* instances is not statistically different, although both the training and prediction time get shorter.

Our empirical evidence results in two lessons. On the one hand, if a person is about to annotate a dataset for learning-to-rank tasks under the Boolean schema, she should resist the temptation of adding an extra intermediate class, as most likely it will just make the problem more complex to the annotators and could even raise the price of crowdsourcing the task. On the other hand, if a person is about to train a model on an existing dataset, which includes the intermediate class, it is safe to discard the corresponding instances of such a class, as the performance of the realistic scenario of identifying relevant documents will not be affected.

As future work, we would like to explore the impact of intermediate classes in more complex annotation schemes, such as graded relevance.

ACKNOWLEDGMENTS

This research was performed by the Arabic Language Technologies (ALT) group at Qatar Computing Research Institute, HBKU. It was carried out within the Interactive sYstems for Answer Search project (Iyas).

We thank H. Mubarak and P. Nakov for their assessment on the original crowdsourced annotation of the CQA-QL-[2015,2016] corpora (http://alt.qcri.org/semeval2016/task3).

REFERENCES

[1] Omar Alonso and Matthew Lease. 2011. Tutorial: Crowdsourcing for Information Retrieval: Principles, Methods, and Applications. In *Proceedings of the SIGIR '11*. Beijing, China. https://www.slideshare.net/mattlease/crowdsourcing-for-information-retrieval-principles-methods-and-applications

[2] Association for Computational Linguistics 2016. *Proceedings of SemEval '16*. Association for Computational Linguistics, San Diego, CA.

[3] Alberto Barrón-Cedeño, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti. 2015. Thread-Level Information for Comment Classification in Community Question Answering. In *Proceedings of ACL-HLT '15*. Association for Computational Linguistics, Beijing, China, 687–693.

[4] Eyal Beigman and Beata Beigman Klebanov. 2009. Learning with Annotation Noise. *Proceedings ACL-IJCNLP '09* August, 280–287.

[5] Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of WMT '13*. Association for Computational Linguistics, Sofia, Bulgaria, 1–44.

[6] Thomas Demeester, Dolf Trieschnigg, Dong Nguyen, and Ke Hiemstra, Djoerd Zhou. 2014. Overview of the TREC 2014 Federated Web Search Track. In *Proceedings of the Twenty-Third Text REtrieval Conference*. Gaithersburg, MD.

[7] Yao-Xiang Ding and Zhi-Hua Zhou. 2016. Crowdsourcing with Unsure Option. In *Proceedings of the NIPS fi16 Workshop on Crowdsourcing and Machine Learning (CrowdML)*. Barcelona, Spain.

[8] Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. KeLP at SemEval-2016 Task 3: Learning Semantic Relations between Questions and Answers, See [2], 1116–1123.

[9] Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. 2015. Structural Representations for Learning Relations between Pairs of Texts. In *ACL-HLT '15*. Association for Computational Linguistics, Beijing, China, 1003–1013.

[10] Panos Ipeirotis. 2011. Crowdsourcing using Mechanical Turk: Quality Management and Scalability. In *Proceedings of CSDM '11*. Hong Kong, China.

[11] Kalervo Järvelin and Jaana Kekäläinen. 2000. IR Evaluation Methods for Retrieving Highly Relevant Documents. In *Proceedings of SIGIR '00*. ACM, New York, NY, 41–48.

[12] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 422–446.

[13] Yandong Liu, Jiang Bian, and Eugene Agichtein. 2008. Predicting Information Seeker Satisfaction in Community Question Answering. In *Proceedings of SIGIR '08*. ACM, New York, NY, 483–490.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

[15] Alessandro Moschitti. 2006. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In *Proceedings of ECML '06*. Springer-Verlag Berlin Heidelberg, Berlin, Germany, 318–329.

[16] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 Task 3: Community Question Answering. In *Proceedings of SemEval '17*. Association for Computational Linguistics, Vancouver, Canada.

[17] Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, James Glass, and Bilal Randeree. 2015. SemEval-2015 Task 3: Answer Selection in Community Question Answering. In *Proceedings of SemEval '15*. Association for Computational Linguistics, Denver, CO.

[18] Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 Task 3: Community Question Answering, See [2], 525–545.

[19] Mark Smucker and Chandra Prakash Jethani. 2011. The Crowd vs. the Lab: A Comparison of Crowd-Sourced and University Laboratory Participant Behavior. In *Proceedings of CIR '11*. Beijing, China, 9–14.

[20] Mark D. Smucker and Matthew Kazai, Gabriella Lease. 2013. Overview of the TREC 2013 Crowdsourcing Track. In *Proceedings of the Twenty-Second Text REtrieval Conference*. Gaithersburg, MD.

[21] Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing Translation: Professional Quality from Non-professionals. In *Proceedings of ACL-HLT '11*. Association for Computational Linguistics, Stroudsburg, PA, 1220–1229.