Taking the best from the Crowd: Learning Question Passage Classification from Noisy Data

Azad Abad
University of Trento
azad.abad@unitn.it

Alessandro Moschitti
Qatar Computing Research Institute
amoschitti@qf.org.qa

Abstract

In this paper, we propose methods to take into account the disagreement between crowd annotators as well as their skills for weighting instances in learning algorithms. The latter can thus better deal with noise in the annotation and produce higher accuracy. We created two passage reranking datasets: one with crowdsource platform, and the second with an expert who completely revised the crowd annotation. Our experiments show that our weighting approach reduces noise improving passage reranking up to 1.47% and 1.85% on MRR and P@1, respectively.

1 Introduction

One of the most important steps for building accurate QA systems is the selection/reranking of answer passage (AP) candidates typically provided by a search engine. This task requires the automatic learning of a ranking function, which pushes the correct answer passages (i.e., containing the answer to the question) higher in the list.

The accuracy of such function, among other, also depends on the quality of the supervision provided in the training data. Traditionally, the latter is annotated by experts through a rather costly procedure. Thus, sometimes, only noisy annotations obtained via automatic labeling mechanisms are available. For example, the Text REtrieval Conference (TREC[1]) provides open-domain QA datasets, e.g., for factoid QA. This data contains a set of questions, the answer keywords and a set of unannotated candidate APs. The labeling of the latter can be automatically carried out by checking if a given passage contains the correct answer keyword or not. However, this method is prone to generate passage labels, i.e., containing the answer keyword but not supporting it. For instance, given the following question, Q, from TREC 2002-03 QA, associated with the answer key *Denmark*:

Q: *Where was Hans Christian Anderson born?*

the candidate passage:

AP: *Fairy Tales written by Hans Christian Andersen was published in 1835-1873 in* ***Denmark***.

would be wrongly labeled as a correct passage since it contains *Denmark*. Such passages can be both misleading for training and unreliable for evaluating the reranking model, thus requiring manual annotation.

Since the expert work is costly, we can rely on crowdsourcing platforms such as CrowdFlower[2] for labeling data, faster and at lower cost (Snow et al., 2008). This method has shown promising results but it still produces noisy labels. Thus, a solution consists in (i) using redundant annotations from multiple annotators and (ii) resolving their disagreements with a majority voting approach (Sheng et al., 2008; Zhang et al., 2015). However, the consensus mechanism can still produce annotation noise, which (i) depends on crowd workers' skill and the difficulty of the given task; and (ii) can degrade the classifier accuracy.

In this paper, we study methods to take into account the disagreement among the crowd annotators as well as their skills in the learning algorithms. For this purpose, we design several instance weighting strategies, which help the learning algorithm to deal with the noise of the training examples, thus producing higher accuracy.

More in detail: firstly, we define some weight factors that characterize crowd annotators' skill, namely: *Prior Confidence*, which indicates the previous performance of the crowd worker re-

[1] http://trec.nist.gov

[2] http://www.crowdflower.com

*Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*SEM 2016)*, pages 136–141,
Berlin, Germany, August 11-12, 2016.

ported by the crowdsourcing platform; *Task Confidence*, which is determined by the total number of annotations performed by the crowd worker in the target task; and *Consistency Confidence*, which quantify the agreements between the annotator and the majority voting labels. We used these parameters for building our weighting functions, which aim at reducing the impact of the noisy annotations in learning algorithms.

Secondly, we build a passage reranking dataset based on TREC 2002/2003 QA. We used Crowdflowers for carrying our an intial noisy annotation and we had an expert to manually verify and corrected incorrect labels. This is an important QA resource that we will release to the research community. Additionally, the accuracy of our models, e.g., classifiers and search engines, tested on such gold standard data establish new baselines, useful for future research in the field.

Finally, we conducted comparative experiments on our QA dataset using our weighting strategies. The results show that (i) our rerankers improve on the IR baseline, i.e., BM25, by 17.47% and 19.22% in MRR and P@1, respectively; and (ii) our weighting strategy improves the best reranker (using no-weighting model) up to 1.47% and 1.85% on MRR and P@1, respectively.

2 Related Work

Crowdsourcing has been used in different domains to collect annotations. Kilgarriff (1998) proposed a model for generating golden standard datasets for word-sense disambiguation. The work in (Voorhees, 2000; Volkmer et al., 2007; Alonso and Mizzaro, 2012) considers relevance judgments for building IR systems. Works closer to this paper proposed by Donmez et al. (2009), Qing et al. (2014), Raykar et al. (2010), Whitehill et al. (2009) and Sheng et al. (2008), targeted the quality of crowdsourced annotation and how to deal with noisy labels via probabilistic models. Our approach is different as we do not improve the crowd annotation, but design new weighing methods that can help the learning algorithms to deal with noise. Plank et al. (2014) also propose methods for taking noise into account when training a classifier. However, they modify the loss function of a perceptron algorithms while we assign different weights to the training instances.

Regarding QA and in particular answer sentence/passage reranking there has been a large body of work in the recent years, e.g., see (Radlinski and Joachims, 2006; Jeon et al., 2005; Shen and Lapata, 2007; Moschitti et al., 2007; Surdeanu et al., 2008; Wang et al., 2007; Heilman and Smith, 2010; Wang and Manning, 2010; Yao et al., 2013), but none of them was devoted to exploit annotation properties in their model.

3 Crowdsourced Dataset

Initially, we ran a crowdsourcing task on CrowdFlower micro-tasking platform and asked the crowd workers to assign a relevant/not relevant annotation label to the given Q/AP pairs. The crowd workers had to decide whether the given AP supports the raised question or not. We consider the TREC corpora described in Section 5.1 and in particular the first 20 APs retrieved by BM25 search engine for every question. We collect 5 judgments for each AP. Additionally, we removed the maximum quota of annotations a crowd worker can perform. We demonstrated that this (i) does not affected the quality of the annotations in Section 5.1; and (ii) allows us to collect reliable statistics about the crowd annotators since they can participate extensively to our annotation project. The intuition behind the idea is: *a crowd worker is more reliable for a given task if (s)he annotates more passages*. Finally, we used control questions discarding the annotation of crowd annotators providing incorrect answers.

Overall, we crowdsourced 527 questions of the TREC 2002/2003 QA task and collected 52,700 judgments. The number of the participant workers was 108 and the minimum and maximum number of answer passages annotated by a single crowd annotator were 21 and 1,050, respectively.

To obtain an accurate gold standard, we asked an expert to revise the passages labeled by crowd annotators when at least one disagreement was present among the annotations. This *super* gold standard is always and only used for testing our models (not for training).

4 Weighting models for learning methods

We define weighing schema for each passage of the training questions. More in detail, each question q is associated with a sorted list of answer passages. In turn, each passage p is associated with a set of annotators $\{a_p^1, a_p^2, ..., a_p^k\}$, where a_p^h is the annotator h, $j_p^h \in \{+1, -1\}$ is her/his judgment, and k is the number of annotators per

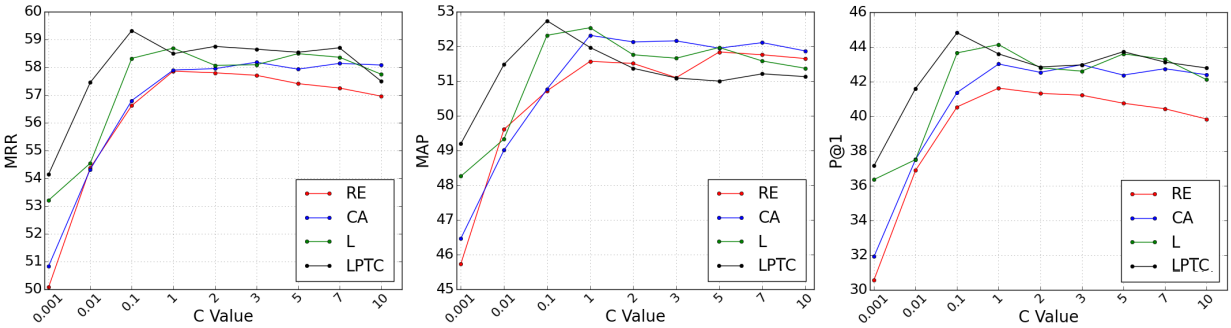

Figure 1: The impact of the C values on different models with (LPTC, L) and without (CA, RE) instance weighting.

passage. We defined a weighting function, $f(\cdot)$, for scoring the passage p as:

$$f(p) = |\sum_{h=1}^{k} j_p^h W(a^h)|. \quad (1)$$

The weighting function consists of a summation of two factors: (i) j_p^h, which indicates the judgment value the annotators, h, have provided for the passage p; and (ii) $W(u)$, which aims at capturing the reliability of the crowd worker u, using the product of three factors:

$$W(u) = P(u)T(u)C(u), \quad (2)$$

where *Prior Confidence*, $P(u)$, indicates the prior trust confidence score of the crowd worker, u, provided by the crowdsourcing platform based on the quality of the annotations (s)he has done in the previous tasks. *Task Confidence*, $T(u)$, indicates the total number of annotations performed by the crowd worker u in this task. The score is re-scaled and normalized between (0,1) by considering the maximum and minimum number of annotations the workers have done in this task. *Consistency Confidence*, $C(u)$, indicates the total number of annotation agreements between the annotator u and the majority voting in this task. The score is normalized and re-scaled between (0,1) as well.

We use Eq. 1 in the optimization function of SVMs:

$$\min \quad \frac{||\vec{w}||^2}{2} + c\sum_i \xi_i^2 f(p_i), \quad (3)$$

where $\vec{w}$ is the model, c is the trade-off parameters, ξ_i is the slack variable associated with each training example $\vec{x}_i$, p_i is the passage related to the example x_i (i.e., associated with a constraint), and $f(p_i)$ (Eq. 1) assigns a weight to such constraint.

5 Experiments

5.1 Experimental Setup

QA Corpora. In this paper, we used the questions from TREC 2002 and 2003 from the large newswire corpus, AQUAINT. We created the Q/AP pairs training BM25 on AQUAINT and retrieving candidate passages for each question.

Crowdsourcing Pilot Experiments. Before running the main crowdsourcing task, we evaluated the effect of the initial configurations of the platform on the quality of the collected annotation. We conducted two pilot crowdsourcing experiments, which show that without quota limitation, the collected sets of annotations have both high level of agreement (0.769) calculated with the Kappa statistic (Carletta, 1996).

Classifier Feature. We used the rich set of features described in the state-of-the-art QA system (Tymoshenko and Moschitti, 2015). Such features are based on the similarity between question and the passage text: N-gram overlap (e.g., word lemmas, bi-gram, part-of-speech tags and etc.), tree kernel similarity, relatedness between question category and the related named entity types extracted from the candidate answer, LDA similarity between the topic distributions of question and answer passage.

Reranking Model We used (i) a modified algorithm of SVM-rank [3] using the Eq. 3 to train our rerankers; (ii) the default cost-factor parameter; and (iii) some other specific values to verify if our results would be affected by different C values.

Baselines. We compared our results with three different baselines, namely: **BM25**: we used Terrier search engine[4], which provides BM25 scor-

[3] http://svmlight.joachims.org
[4] http://terrier.org

Model	MRR	MAP	P@1
Baselines			
BM25	41.75 ± 6.56	37.25 ± 4.52	25.57 ± 6.17
RE	57.41 ± 7.31	51.75 ± 6.27	41.38 ± 11.12
CA	57.75 ± 6.77	52.09 ± 5.68	42.94 ± 8.55
Our Weighting Results			
L	58.73 ± 6.88	52.48 ± 6.00	44.12 ± 9.75
P	58.51 ± 5.63	52.07 ± 4.63	43.15 ± 7.32
LP	58.76 ± 6.52	52.60 ± 6.03	44.22 ± 8.72
TC	58.31 ± 5.44	52.09 ± 4.96	42.83 ± 7.69
LTC	58.85 ± 5.85	52.58 ± 5.52	43.74 ± 8.50
LPTC	59.22 ± 6.30	52.63 ± 5.96	44.79 ± 8.82

Table 1: Results over 5 fold cross validation. Our Weighting Results are all better than the Baselines with a statistical significant test of 95%.

ing model to index the answer passages (Robertson and Walker, 1997). The APs are extracted from AQUAINT text corpus and treated as documents. BM25 is used to retrieve 20 candidate answers for each question and rank them by their relevance scores. **RE** (regular expression): we trained a classifier with the noisy annotations produced by labels automatically derived with RE applied to answer keys (no weighting strategy). **CA** (crowd annotations): we train a classier with the same configuration as RE but using majority voting as a source of supervision.

Evaluation Metrics We evaluated the performance of the classifier with the mostly used metrics for QA tasks: the Mean Reciprocal Rank (MRR), which computes the reciprocal of the rank at which the first relevant passage is retrieved, Precision at rank 1 (P@1), which reports the percentage of question with the correct answer at rank 1, and Mean Average Precision (MAP), which measures the average of precision of the correct passages appearing in the ranked AP list. All our results are computed with 5-folds cross validations, thus the above metrics are averaged over 5 folds.

5.2 Weighting Experiments

In these experiments, we used the labels provided by crowd annotators using majority voting for training and testing our models. Most interestingly, we also assign weights to the examples in SVMs with the weighting schemes below:

- Labels Only (L), i.e., we set $P(u) = T(u) = C(u) = 1$ in Eq. 2. This means that the instance weight (Eq. 1) is just the sum of the labels j_p^h.

- Prior Only (P): to study the impact of prior annotation skills, we set $C(u) = T(u) = 1$ in Eq. 2, and we only use $P(u)$ (crowdflower trust), i.e., we do not account for the sign of annotations, j_p^h.

- Labels & Prior (LP): the previous model but we also used the sign of the label, j_p^h.

- Task & Consistency (TC): we set $P(u) = 1$ such that Eq. 2 takes into account both annotator skill parameters for the specific task, i.e., task and consistency confidence, but only in the current task and no sign of j_p^h.

- L & TC (LTC): same as before but we also take into account the sign of the annotator decision.

- LPTC: all parameters are used.

Table 1 shows the evaluation of the different baselines and weighting schemes proposed in this paper (using the default c parameter of SVMs). We note that: firstly, the accuracy of BM25 is lower than the one expressed by rerankers trained on noisy labels (-15.66% in MRR, -14.5% in MAP, -15.81 in P@1%).

Secondly, although there is some improvement using crowd annotations for training[5] compared to the noisy training labels (RE), the improvement is not significant (+0.34% in MRR, +0.34% in MAP, +1.56% in P@1). This is due to three reasons: (i) the crowdsourcing annotation suffers from a certain level of noise as well (only 27,350 of the answer passages, i.e., 51.80%, are labeled with "crowd fully in agreement"), (ii) although the RE labels may generate several false positives, these are always a small percentage of the total instances as the dataset is highly unbalanced (9,535 negative vs. 1,005 positive examples); and (iii) RE do not generate many false negatives as they are precise.

Thirdly, the table clearly shows the intuitive fact that it is always better to take into account the sign of the label given by the annotator, i.e., LP vs. L and LTC vs. TC.

Next, when we apply our different weighting schema, we observe that the noise introduced by the crowd annotation can be significantly reduced as the classifier improves by +1.47% in MRR, +0.54% in MAP and +1.85% in P@1, e.g., when using LTC & LPTC compared to CA, which does not provide any weight to the reranker.

Finally, as the trade-off parameter, c, may alone mitigate the noise problem, we compared our models with the baselines according to several value of the parameter. Fig. 1 plots the rank measures averaged over 5-folds: our weighting methods, especially LPTC (black curve), is constantly

[5]The test labels are always obtained with majority voting and we removed questions that have no answer in the first 20 passages retrieved by BM25.

better than the baseline, CA, (blue curve) in MRR and P@1.

6 Conclusions

Our study shows that we can effectively exploit the implicit information of crowd workers and apply it to improve the QA task. We demonstrated that (i) the best ranking performance is obtained when the combination of different weighting parameters are used; and (ii) the noise of annotations, present in crowdsourcing data, can be reduced by considering weighting scores extracted from crowd worker performance. In the future, we will explore better weighting criteria to model the noise that is induced by annotations of crowd workers.

Acknowledgement

This work has been partially supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Researchand Innovation action).

References

Omar Alonso and Stefano Mizzaro. 2012. Using crowdsourcing for trec relevance assessment. 48(6).

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.*

Pinar Donmez, Jaime G. Carbonell, and Jeff Schneider. 2009. Efficiently learning the accuracy of labeling sources for selective sampling. KDD '09, New York, NY, USA.

David A. Ferrucci. 2011. Ibm's watson/deepqa. ISCA '11, New York, NY, USA. ACM.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. HLT '10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. CIKM '05, New York, NY, USA. ACM.

Adam Kilgarriff. 1998. Gold standard datasets for evaluating word sense disambiguation programs.

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *In Proceedings of the 45th Conference of the Association for Computational Linguistics*.

Barbara Plank, Dirk Hovy, and Anders Sogaard, 2014. *Learning part-of-speech taggers with inter-annotator agreement loss*, pages 742–751. Association for Computational Linguistics.

Ciyang Qing, Ulle Endriss, Raquel Fernández, and Justin Kruger. 2014. Empirical analysis of aggregation methods for collective annotation. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING-2014)*.

Filip Radlinski and Thorsten Joachims. 2006. Query chains: Learning to rank from implicit feedback. *CoRR*, abs/cs/0605035.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *J. Mach. Learn. Res.*

S. E. Robertson and S. Walker. 1997. On relevance weights with little relevance information. SIGIR '97, pages 16–24, New York, NY, USA. ACM.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. pages 12–21, Prague, Czech Republic, June. Association for Computational Linguistics.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. KDD '08, New York, NY, USA. ACM.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. EMNLP '08, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online qa collections. In *In Proceedings of the 46th Annual Meeting for the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT*, pages 719–727.

Kateryna Tymoshenko and Alessandro Moschitti. 2015. Assessing the impact of syntactic and semantic structures for answer passages reranking. CIKM '15, pages 1451–1460, New York, NY, USA.

Timo Volkmer, James A Thom, and Seyed MM Tahaghoghi. 2007. Modeling human judgment of digital imagery for multimedia retrieval. *Multimedia, IEEE Transactions on*, 9(5).

Ellen M. Voorhees. 2000. Variations in relevance judgments and the measurement of retrieval effectiveness. *Inf. Process. Manage.*, pages 697–716.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. COLING '10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. *EMNLP-CoNLL*, 7.

Jacob Whitehill, Ting fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. Curran Associates, Inc.

Xuchen Yao, Benjamin Van Durme, Chris Callison-burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *In North American Chapter of the Association for Computational Linguistics (NAACL.*

Jing Zhang, Victor S. Sheng, Jian Wu, Xiaoqin Fu, and Xindong Wu. 2015. Improving label quality in crowdsourcing using noise correction. CIKM '15, New York, NY, USA. ACM.