Information Sciences 608 (2022) 1029–1044

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An error consistency based approach to answer aggregation in open-ended crowdsourcing

Lei Chai [a,c], Hailong Sun [b,c,*], Zizhe Wang [a,c]

[a] SKLSDE, School of Computer Science and Engineering, Beihang University, XueYuan Road No.37, HaiDian District, Beijing 100191, China
[b] SKLSDE, School of Software, Beihang University, XueYuan Road No. 37, HaiDian District, Beijing 100191, China
[c] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, XueYuan Road No. 37, HaiDian District, Beijing 100191, China

ARTICLE INFO

Article history:
Received 6 December 2021
Received in revised form 28 June 2022
Accepted 1 July 2022
Available online 5 July 2022

Keywords:
Crowdsourcing
Open-ended text annotation
Answer aggregation
Annotation error consistency

ABSTRACT

Crowdsourcing plays a vital role in today's AI industry. However, existing crowdsourcing research mainly focuses on those simple tasks that are often formulated as label classification, while complex open-ended tasks such as question answering and translation have not received much attention. Such tasks usually have open solution spaces and non-unique true answers, which pose great challenges for designing effective crowdsourcing algorithms. In this work, we are concerned specifically with complex text annotation crowdsourcing tasks, where each answer of a task is in the form of free text. We propose an error consistency-based approach to inferring a satisfying result from a set of open-ended answers. First, each answer is represented with two vectors that capture the local word collocation and the global sentence semantics respectively. Second, the true answer is approximated by the sum of the answer vectors weighted by the reciprocals of their respective errors. Third, an algorithm called *AEC* (**A**ggregation based on **E**rror **C**onsistency) is designed to infer the aggregated result by maximizing the consistency of the errors of an answer in two vector spaces. Experimental results on two datasets demonstrate the effectiveness of our approach.

1. Introduction

Crowdsourcing is increasingly used for collecting labeled data (hereafter termed as annotation) to train machine learning models or solve the tasks that are difficult or costly for machines, e.g., image classification [13], optical character recognition (OCR) [43], and sentiment analysis [31]. Due to the openness and redundancy-based task assignment mechanism of crowdsourcing, multiple candidate answers of varying quality are usually obtained for each task. The question of how to effectively infer the true answer becomes a fundamental problem, which is often referred to as answer aggregation or truth discovery.

Although much research has been devoted to this topic [49], most existing work makes the following assumptions [38]: 1) the answers are independent of each other, 2) answers come from a finite solution space, and 3) there is a unique true answer to a task. As a result, existing approaches mostly address simple tasks, such as classification [1] and rating [27], thus are not suitable for complex open-ended crowdsourcing tasks with open solution space and non-unique true answers [36].

In this work, we address answer aggregation for complex text annotation crowdsourcing tasks, such as question answering and translation, where each answer of a task is in the form of free text. More specifically, how to analyze and aggregate

* Corresponding author at: SKLSDE, School of Software, Beihang University, XueYuan Road No. 37, HaiDian District, Beijing 100191, China.
E-mail addresses: chailei@act.buaa.edu.cn (L. Chai), sunhl@buaa.edu.cn (H. Sun), wangzz@act.buaa.edu.cn (Z. Wang).

https://doi.org/10.1016/j.ins.2022.07.001
0020-0255/

multiple open-ended text annotations with diverse reliability and phraseology becomes a challenging problem, which has attracted the interest of the NLP (Natural Language Processing) community [38]. To solve this problem, the following issues need to be addressed:

- Annotation Embedding. Before aggregating crowdsourced annotations, we need a suitable representation of these annotations, which not only captures various aspects of information for the textual annotations like word allocation and contextual information, but also reflects the relationship between annotations.
- Similarity Modeling & Parameter Estimation. Many existing works have demonstrate that parameters such as worker ability, task difficulty and annotation errors are crucial for effective aggregation of answers, and these parameters are usually estimated by annotation similarity. In classification problems, the similarity relation between annotations degenerates into an equivalence relation. For example, whether the classification label of annotation a^w produced by worker w is equal to other annotations and the truth label for the same task determines the reliability of worker w. Due to the complexity of textual annotations, the similarity relationship between annotations cannot be directly computed to estimate these parameters. It is challenging to model the relationship between the complex annotations and these parameters.

To overcome the above challenges, we propose an answer aggregation framework for open-ended crowdsourcing. For annotation representations, previous work [25] has shown that the similarity relationship between two text sequences with different representation methods are not exactly consistent and may even be quite different. In particular, the *GLEU* metric [46] mainly focuses on the word collocations in the sentences while the distributed sentence embedding concentrates more on the global information like context, syntax, semantics between sentences. Hence we represent and model the annotations similarity in the feature spaces of "GLEU" and "universal sentence embedding" respectively for fully utilizing the rich information in the textual annotations. For parameter estimation and truth selection, we propose an optimization algorithm based on the basic idea of "annotation error consistency", which we believe that for each task, the inherent error of each annotation regarding the truth should not be affected by different metrics, the error of a crowdsourced answer should be consistent across different feature spaces. Since the truth is not known, we take the average degree of deviation between each annotation and others in different feature spaces as an approximation to the actual error and approximate it gradually by our proposed *AEC* algorithm. Specifically, we compute the error of each annotation by minimizing the relative distance between each annotation and the truth in the feature spaces of GLEU [46] and Universal Sentence Encoder [6], then we select the best annotation with the minimum error. In summary, the main contributions of this work are as follows:

- We propose to represent the open-ended text annotations with multiple embeddings that can fully capture the rich information of word collocation and global sentence semantics in the text.
- We propose an error consistency based answer aggregation approach by maximizing the error consistency between different representations with a gradient descent based algorithm.
- We conducted experiments on a crowdsourced translation dataset and a simulated question answering dataset, the results demonstrate the effectiveness of our method.

The paper is organized as follows. In Section 2, we briefly review related work. We conduct preliminary studies to figure out the research questions in Section 3. We present the details of our proposed method in Section 4. The experimental results are presented and analyzed in Section 5. After that, we illustrate the limitation of this work, and briefly describe the direction of the follow-up works in Section 6. Finally, we conclude our paper in Section 7.

2. Related works

Answer aggregation is critical for obtaining high-quality labeled data in crowdsourcing and has been widely studied [32,11,12,7,41,20,40,47]. Without loss of generality, existing work can be categorized according to the following two factors:

Task modeling, which involves modeling a task in terms of difficulty, latent topics and so on.

Worker modeling, which focuses on characterizing a worker's attributes such as capability, confidence and bias.

2.1. Task modeling

In many works, the difficulty of a task is modeled with a real number [34,45], which is thought to be highly relevant to the accuracy of the annotations. For example, according to [45], the relationship between accuracy and task difficulty is depicted with the following equation:

$$Pr(v_i^w = v_i^*|d_i, c^w) = \frac{1}{1 + e^{-\frac{c^w}{d_i}}}, \tag{1}$$

where $c^w \in (0,+\infty)$ represents the worker capability, the $d_i \in (0, +\infty)$ denotes the difficulty level of a task t_i. Specifically, the higher d_i represents the higher task difficulty. It can be easily derived that for a fixed worker's capability c^w, an easier task (with a lower d_i) will lead to a higher probability that a worker correctly answers the task.

Beyond that, several researchers [44] explore to model the task information with K-dimensional vectors. For example, [18,34] propose to analyze the information of task description with the topic model [3,48] to generate the k-dimension task vectors. This task-related information is helpful to accurately derive the task difficulty parameter.

2.2. Worker modeling

The worker information is usually studied to model the quality of workers in crowdsourcing tasks. The worker probability c_t^w is commonly used in previous works [1,12,33,16] to represent the capability that worker w could accurately process the task t. A higher c_t^w indicates a higher capability of worker w to answer task t. Moreover, in category-based annotation tasks, the confusion matrix [11,23,41,42] is often used to model the probability of each worker producing a given categorical annotation given the golden annotation. For example, for a K-option task t, the confusion matrix con^w for worker w is a $K \times K$ matrix, where the j-th row $(1 \leqslant \mathrm{j} \leqslant \mathrm{k})$ $con_j^w = [con_{j,1}^w, con_{j,2}^w, \ldots, con_{j,k}^w]$ represents the probability distribution of worker w's possible answers for a task if the truth of the task is the j-th choice. Every $con_{j,m}^w$ $(1 \leqslant \mathrm{j} \leqslant \mathrm{k}, 1 \leqslant \mathrm{m} \leqslant \mathrm{k})$ means that the probability that worker w chooses the m-th choice when the true answer is the j-th, which can be formulated as follows:

$$c_{j,m}^w = Pr(a_w = m | a_{true} = j), \tag{2}$$

It is worth noting that the confusion matrix-based methods cannot be directly applied to non-category tasks like multiple-choice tasks, sequence annotation tasks and open-ended annotation tasks. Moreover, annotation error can be evaluated by modeling tasks and workers, which is a direct aid to truth discovery.

2.3. Complex crowdsourcing tasks

Most existing crowdsourcing work focuses on modeling the information of tasks and workers in simple tasks, such as *Decision-Making* [18,34,42,21,17], *Single-Choice*[28,8], *Multiple-Choice* [37,15], and *Numeric Tasks* [27].

Recently, there has been some work on answer aggregation for complex tasks. Nguyen et al. [35] proposed an HMM-based aggregation method for sequence labeling tasks such as named entity resolution. Braylan et al. [4] proposed a multi-dimensional scaling-based method for modeling and aggregation for complex annotation (e.g., Syntactic Parse Tree, Sequence Annotation, Ranking of Elements ordered, Translation). Li et al. [26] propose a reliability-aware sequence aggregation method for solving sequence tasks like translation.

Recent research efforts on the complex annotation tasks indicate a new possibility for generating high-quality complex annotations and better solve the complex tasks with the wisdom of crowds. However, several challenges still need to be solved to improve its reliability and expand the scope. Unlike the sequence tasks, the answer space of open-ended textual annotation is usually very large or infinite, it cannot be represented and aggregated as a sequence classification problem. It is worth discussing how to accurately represent the textual annotation and compute their reliability.

3. Preliminary studies

We conduct a series of preliminary studies to illustrate the core issues of proposing the answer aggregation algorithm for open-ended crowdsourcing and to present the design process of our approach. The challenges mentioned in Section 1 lead to two research questions in open-ended crowdsourcing. One question ($RQ1$) involves identifying an appropriate representation for each annotation prior to the step of answer aggregation, that the important information of the textual annotations can be fully captured. The second question ($RQ2$) is about estimating the approximate representation of the true answer for each task, so that the estimating parameters (like annotation error) can be accurately derived.

3.1. RQ1: How to Represent the Open-Ended annotations into the Feature Spaces? Do Different Encoding Methods Affect the Similarity Relationship Between Annotations?

Annotations for simple tasks like $Decision-Making, Single-Choice$, or *Numeric Tasks* can be directly represented in a simple and unambiguous way. For example, a sentiment analysis task generally asks for a label selection in ('*positive*', '*negative*', '*neutral*') for a given sentence, it is a typical $Single-Choice$ task. The simple representation allows direct evaluation of the relationship based on the exact match between the annotations and the true answer. In open-ended crowdsourcing, the annotations are in the form of free text. The relationship between annotations derived by different annotation embedding methods and evaluate metrics may lead to quite different results, which pose challenges for parameters estimation and answer aggregation. Thus it is vital to design a new representation and aggregation method that goes beyond simple exact matching.

Many models have been proposed for sentence representation, such as N-gram[19], universal sentence encoder[6], GloVe [39], and Bert[14]. These sentence representation methods can be divided into two categories. One is word-centered, with which sentence representation is obtained based on the word relationship between annotations, like N-gram-based methods. The other type of method pays more attention to obtaining the global information about the sentence like syntax, semantics or context information by the distributed representation. Considering the different emphasis of these two types

of representation methods, we analyze the differences between these representation methods by measuring the annotation similarity on the open-ended crowdsourcing dataset.

Table 1 (a) shows the answers provided by four crowd workers to a popular science Q&A task 'How substances change in a chemical reaction? Table 1 (b) shows the annotation similarity measured by the metric of *gleu* and *embedding similarity*, and their difference. Given that the values of *gleu-sim* and *emb-sim* are in the range of [0, 1], which represent the similarity between two annotations, we take the relative value of their difference to represent the 'Difference', which could be derived by $(emb - sim\text{-}gleu - sim)/gleu - sim^*100\%$. As shown in the table, the significant differences exist in the annotation similarity measured by the metric of *gleu* and *embedding*. In particular, the difference caused by these two metrics on the similarity between *annotation*1 and *annotation*4 is 47.2%, indicating that different metrics (e.g., *gleu similarity* and *embedding similarity*) for relationship modeling between annotations can lead to very different results.

To verify this phenomenon, we conducted further experiments on the *WSA* and *ARC* datasets. The results are shown in Fig. 1.

Fig. 1(a) and (b) lists the average distance (The detailed calculation process will be introduced in Section 4.2) for each annotation. The x-axis represents the annotation index and the y-axis represents the corresponding average distance. Fig. 1 (c) and (d) exhibit the average annotation distance with a normal distribution. The x-axis denotes the value of average distance and the y-axis represents the corresponding probability density.

The wide difference in value and distribution between the red lines (average distance derived from the $gleu - sim$) and the blue lines(average distance derived from the $emb - sim$) indicates that the representation and evaluation methods have a significant impact on the analysis of the relationship between open-ended annotations. Therefore, in this work, we represent each annotation in the feature spaces of *GLEU* and *UniversalSentenceEncoder* respectively. Beyond that, we compare the impact of the different representation methods and their combination on the experimental results to explain 'why *GLEU* and *Universal Sentence encoder*' in Section 5.3.

3.2. RQ2: how to accurately estimate the true answer with crowdsourced annotations?

The above related works indicate that most of the existing work estimates the annotation error (or reliability) by the key parameters like 'worker capability' and 'task difficulty'. Then the annotation with the smallest error coefficient is selected as the best answer. Considering that open-ended annotations are usually represented by high-dimensional vectors in the feature space, the error coefficient for each annotation can be directly derived by the similarity of vectors between annotations and the truth, when we can make an accurate estimate of the truth annotation in a certain feature space. Since the true answer is usually unknown for truth inference tasks in crowdsourcing, how to accurately estimate the true answer becomes the key to accurately representing the annotation error.

Braylan et al.[4] intuitively model the relation between the truth, noisy annotations and error from the perspective of geometry. As shown in Fig. 2, the emoji faces represent the crowdsourced annotations with different categories of 'positive' and 'negative', bold lines represent the similarities between annotations, the circle represents the true answer T_t for task t, and dotted lines show the error for each annotation. Geometrically, the true answer is located in the error-weighted center of annotations. Moreover, the Weighted Majority Algorithm (*WMA*)[30] analyzes the relationship between weights, noisy sources and errors in a more rigorous way. The result of *WMA* illustrates that after the process of adjusting the weights, the upper bound on the number of mistakes from a pool of noisy sources is:

$$\mathcal{O}\log|A| + m, \tag{3}$$

if one noisy source in A makes at most m mistakes. This conclusion suggests that there is an upper bound on the error when the appropriate error coefficients are used as weights for the weighted sum of noisy sources. Therefore we believe that it is

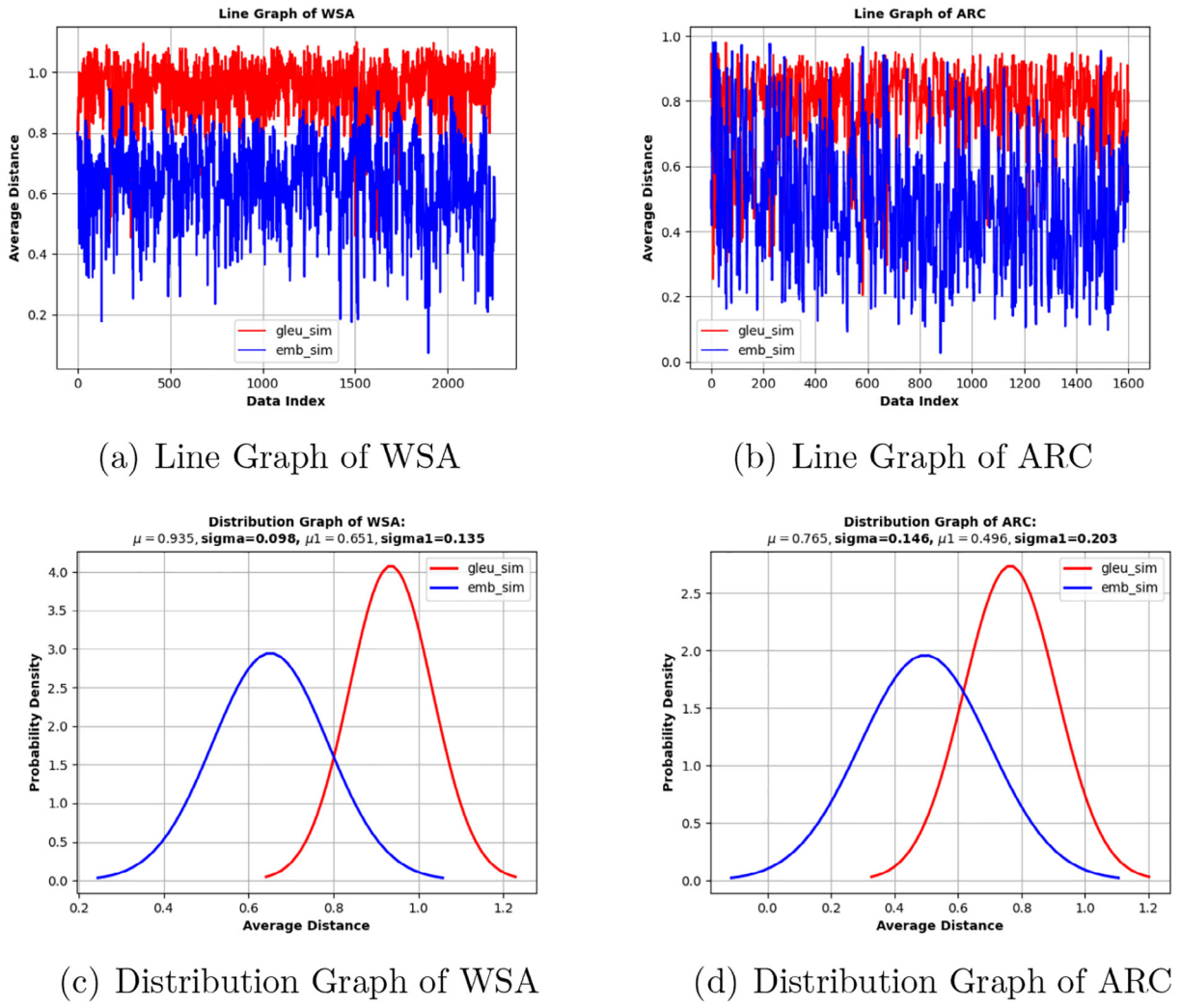
Table 1
Example of ARC dataset (a). Annotation similarity (b) measured by the metric of gleu and embedding, and their difference.

Example of ARC dataset

Worker	Annotation
1	During a chemical reaction, the total amount of matter stays the same.
2	During a chemical reaction, matter is destroyed.
3	During a chemical reaction, one or more new substances are formed.
4	During a chemical reaction, the total number of atoms increases.

Annotation Similarity

AnnoIndex1	AnnoIndex2	gleu-sim	emb-sim	Difference
1	2	0.659	0.651	1.21%
1	3	0.679	0.492	27.5%
1	4	0.519	0.764	47.2%
2	3	0.653	0.635	2.76%
2	4	0.675	0.584	13.5%
3	4	0.611	0.521	14.7%



(a) Line Graph of WSA (b) Line Graph of ARC

(c) Distribution Graph of WSA (d) Distribution Graph of ARC

Fig. 1. The Difference of Data distribution in the feature spaces of *gleu* (red lines) and *universal sentence encoder* (blue lines).

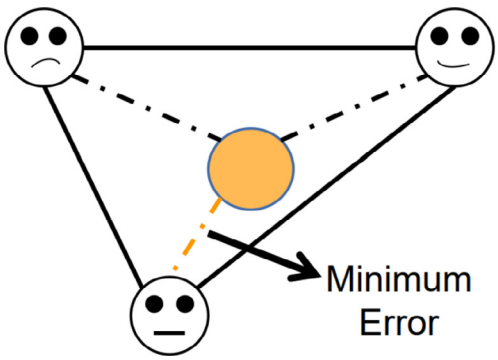


Fig. 2. The Relation Between the Truth, Annotations and Error.

appropriate to estimate the true answer with an error-weighted annotation embedding after considering the intuitive interpretation of Braylan et al.[4] and the quantitative analysis of *WMA* for the relationship between the truth, noisy annotations and errors. It can be formulated as follows:

$$Etr_t = \sum_{i=1}^{m} \frac{emb_{t,i}}{\mathbf{Nor}(e_{t,i})}, \tag{4}$$

where Etr_t denotes the estimated truth of the task t, $emb_{t,i}$ represents the embedding of annotation $a_{t,i}$, $\mathbf{Nor}(e_{t,i})$ denotes the error with normalization for annotation $a_{t,i}$, so that the sum of the weights is 1. It is worth noting that the errors $e_{t,i}$ are the true values derived by the similarity between annotation $emb_{t,i}$ and truth Tr_t, so that the relation between the truth, noisy annotations and errors could be better modeled.

We tested the accuracy of this estimation method on the experimental dataset used in this paper to verify its feasibility, the experimental results are shown in Table 2. For each task t, we computed the similarity between each annotation vector (including our approximate representation of the true value Etr_t) and the truth vector Tr_t based on the Euclidean distance. The average of the similarity ranking is used to illustrate the reliability of this truth estimation method. As shown in Fig. 2, the second column shows the average number of annotations per task for each dataset, and the third column shows the average similarity ranking between the estimated truth and the true answer among all candidate annotations. The average ranking $\mathbf{2.61}(\mathbf{10.96}), \mathbf{1.92}(\mathbf{11}), \mathbf{2.23}(\mathbf{11}), \mathbf{1.31}(\mathbf{5})$ indicate that the estimated truth *Etr* is a more accurate approximation of the ground truth *Tr* compared to other annotations. Hence, it is appropriate to estimate the true answer with an error-weighted annotation embedding.

4. Methods

4.1. Problem formulation

In this section, we first present the problem formulation, and then we show our answer aggregating approach with the steps of 'Annotation representation', 'Modeling and optimization of error consistency', and 'Annotation selection'. Table 3.

Table 3 contains the notations frequently used in this work. Without loss of generality, we focus on the answer aggregation in open-ended crowdsourcing, where each answer a of a task t is in the form of free text. To perform the aggregation task, we define a task set T=$t_1, t_2, \ldots, t_n$ with n tasks. These tasks are assigned to m workers. We define W=$w_1, w_2, \ldots, W_m$ to represent the set of m workers. Generally, a worker answers only a small fraction of the tasks rather than all the tasks. We represent the annotation provided by worker j for task i as $a_{i,j}$. We leverage A=$\{a_t\}$ to denote all the collected annotations in the crowdsourcing task. The answer aggregation problem in open-ended crowdsourcing is then defined as follows,

Definition 1. (Answer Aggregation). Given the set of annotations A produced by crowd workers W for a set of tasks T, the objective of label aggregation is to select the best annotation S_t with the minimum error coefficient e_t for each task $t \in T$.

As shown in Fig. 3, the overall framework consists of the following three steps.

1. *Annotation representation*. We generate two vectors using *GLEU* and *Universal Sentence Encoder* respectively to represent an annotation, that can fully capture the rich information of word allocation and global sentence semantics contained in the textual annotations.
2. *Modeling and optimization of error consistency*. In this step, we first represent the approximated true answer based on the annotation vectors and the respective errors of each answer. Then we try to maximize the error consistency between the two vector spaces by the **AEC** algorithm.
3. *Annotation selection*. For each task we select the annotation with the minimum error as the final result.

4.2. Annotation representation

As described in Section 3.1, we represent each annotation in the feature spaces of *GLEU* and *Universal Sentence Encoder* to fully capture the rich information in the textual annotations.

4.2.1. Global representation with embedding similarity

To capture the global information like syntax, semantics, or contextual information in the textual annotations, we implement a *universal sentence encoder*[6] to encode the annotations into 512-dimensional embeddings emb_g. It is worth noting

Table 2
The reliability of the truth estimation, measured by the average rank of the similarity between the estimated truth and the true answer.

Dataset	Annos/Task	EtrAveRank
WSA_{J1}	10.96	2.61
WSA_{T1}	11	1.92
WSA_{T2}	11	2.23
ARC	5	1.31

Table 3
Notations.

Notations	Description
T	The set of tasks
W	The set of crowd workers
A	The set of annotations
Dis_l	The set of local distance, a higher local distance indicates a higher local error of annotations
Dis_g	The set of global distance, a higher global distance indicates a higher global error of annotations
emb_l	The set of local embedding, which reflect the local information of annotations
emb_g	The set of global embedding, which reflect the global information of annotations
Etr_t	The estimated truth of task t, it is the approximate representation of the ground truth
S_t	The selected answer for task t, it's the best annotation we select from the crowdsourced annotations
D_t	The difficulty of task t, a higher task difficulty indicates a lower probability for receiving correct annotations
C_w	The capability of worker w, a higher worker capability indicates a higher probability for providing correct annotations
$a_{i,j}$	The annotation for task i, provided by worker j
$e_{i,j}$	The error coefficient of $a_{i,j}$, a higher error $e_{i,j}$ indicates a lower probability to be selected as the best answer for annotation $a_{i,j}$
$e(t)$	The set of error coefficient for task t, it is a m-dim vector, where m equals the number of annotations for the task t.

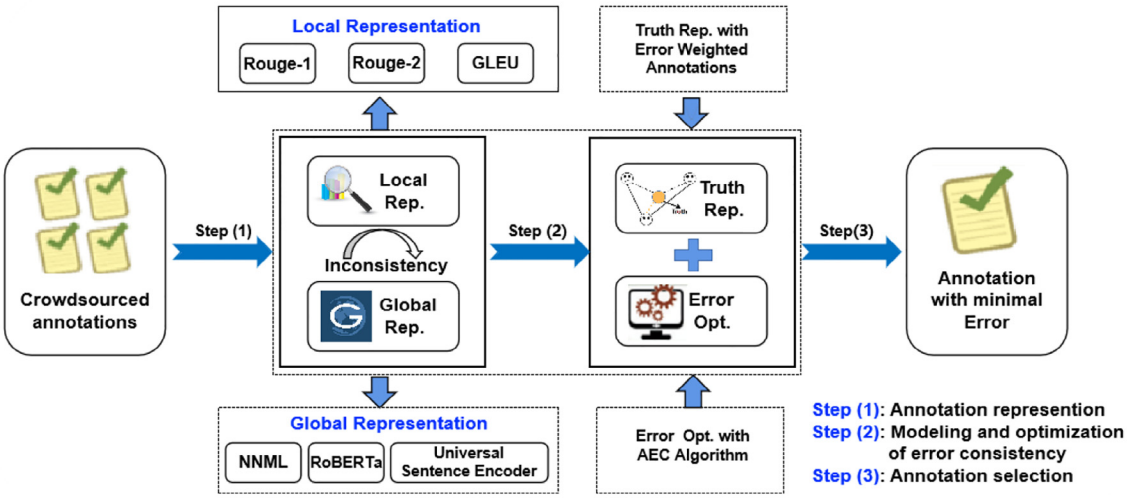


Fig. 3. Overview of the Answer Aggregation Framework for Open-Ended Crowdsourcing.

that the *universal sentence encoder* is a model that encodes sentences into embedding vectors, which can be simply used to calculate sentence level semantic similarity scores and achieve excellent performance on the basis of semantic text similarity.

Then we compute the cosine similarity between every two annotations for the same task. Accordingly, the global distance of annotation $a_{i,j}$ is computed as follows:

$$Dis_g(a_{i,j}) = \frac{1}{(m-1)\sum_k sim(emb_g(i,j), emb_g(i,k))}, \tag{5}$$

where $Dis_g(a_{i,j})$ is the global distance of the annotation $a_{i,j}, k \in [1], k \neq j$. Essentially, Eq. 5 defines the global error of the annotation. m is the number of workers providing annotations for the *task* i, sim is the cosine similarity.

4.2.2. Local representation with GLEU similarity

The *GLEU* score [46], mainly focuses on measuring the difference of word collocation between individual sentences. The *GLEU*[1] distance between annotations emphasizes the information of word collocation. Hence the local information of annotations can be represented by the *GLEU* similarities. The *GLEU* similarity between annotation $a_{i,j}$ and $a_{i,k}$ are as follows:

$$Gleu_{j,k} = 1 - \frac{GLEU(a_{i,j}, a_{i,k}) + GLEU(a_{i,k}, a_{i,j})}{2}, \tag{6}$$

Beyond that, the local average distance of annotation $a_{i,j}$ can be computed between the annotations as follow:

$$Dis_l(a_{i,j}) = \frac{\sum_k GLEU(a_{i,j}, a_{i,k}) + GLEU(a_{i,k}, a_{i,j})}{2(m-1)}, \tag{7}$$

The local embedding of annotation $a_{i,j}$ can be represented as:

$$emb_l(a_{i,j}) = Gleu_{j,1}, Gleu_{j,2}, \ldots, Gleu_{j,m-1}, \tag{8}$$

where m is the number of annotations of the *task i*. For example, the results in Table 1(b represent the distance relations between the four annotations. The numbers in the third column denote the relation between these annotations measured by *GLEU* similarity. Hence we can represent the *GLEU* embedding of annotation-1 with a three-dimensional vector:

$$emb_l(annotation-1) = (0.659, 0.679, 0.519), \tag{9}$$

Table 1 shows an example of input annotations and output similarities measured by *GLEU* and *universal sentence encoder*.

As described in Section 3, the distribution of annotation distances measured by *GLEU* and the *Universal Sentence Encoder* are not exactly consistent. Fig. 1 illustrates the large difference in the data distribution with the representation of *GLEU* similarity and embedding similarity. Therefore, it is important to consider the differences in different spaces in the process of representation and aggregation for open-end text annotations.

4.3. Modeling and optimization of error consistency

In this section, we will introduce how we model the relationship between the estimated truth, crowdsourced annotations and the respective error, formulate the problem of error optimization. It is worth noting that the *global error* and *local error* derived in Section 4.2 indicate the average degree of deviation between each annotation and others, which can be considered as approximations of the actual error.

4.3.1. Representation of the estimated truth

The error coefficient represents the difference between the crowdsourced annotations and the ground truth, where the ground truth is unknown. As discussed in the *RQ*2, the true answer in this work is approximated by the sum of the answer vectors weighted by the reciprocals of their respective errors.

The estimated truth of task t can be computed by the crowd annotations and error as follows:

$$Etr_t = \sum_{i=1}^{m} \frac{emb_{t,i}}{\mathbf{Nor}(e_{t,i})}, \tag{10}$$

where $emb_{t,i}$ represents the embedding of annotation $a_{t,i}$ and $\mathbf{Nor}(e_{t,i})$ denotes the error with normalization for annotation $a_{t,i}$, so that the sum of the weights is 1. $Etr_t(g)$ and $Etr_t(l)$ are computed by emb_g and emb_l separately.

4.3.2. Parameter optimization with annotation error consistency

Although significant differences in range and distribution (shown in Fig. 1) are observed between the global and local distance, the inherent error of each annotation regarding the truth should not be affected by different metrics. For example, the virtual difference between the annotation "It is regrettable for me to take 50 h" and the truth "I feel wasting the time if it takes 50 h" is fixed but the errors calculated by GLEU distance and embedding similarity are **0.6724** and **0.8145**, respectively. Considering that the truth annotation of each task is represented by crowd annotations and their errors, so we can accurately infer the inherent error of the annotations by maximizing the consistency between the different error representations.

First of all, we represent the inconsistency with a loss function as follows:

$$Loss_j(t) = \left| \frac{Sim(emb_l(a_{t,j}), Etr_t(l)) - Min_{gleu}(t)}{Max_{gleu}(t) - Min_{gleu}(t)} - \frac{Sim(emb_g(a_{t,j}), Etr_t(g)) - Min_{sen}(t)}{Max_{sen}(t) - Min_{sen}(t)} \right|, \tag{11}$$

where *Sim* is cosine similarity here, the coefficients of *Max* and *Min* are used to normalize the two types of errors into the value domain. The $Loss_j$ indicates the relative difference between the global error and the local error for annotation $a_{t,j}$, which

[1] Implemented by 'NLTK': A widely used Natural Language Toolkit created by the University of Pennsylvania

derived by the similarity between the annotation $a_{t,j}$ and the estimated truth Etr_t. In particular, as shown in Eq. 10, the estimated truth Etr_t is derived by the embedding and error coefficient of the annotations. Hence, for each task t, the error e_t is the only variable for the $Loss(t)$. After that, we can optimize the error coefficient for each annotation accurately by minimizing the loss function $Loss$, which can be represented as follows:

$$e(t) = \arg\min Loss_e(t), \tag{12}$$

we iteratively calculate the $e(t)$ with gradient descent algorithm. Firstly we compute the gradient as follows:

$$Grad = \frac{\partial Loss_t}{\partial e(t)}, \tag{13}$$

then we update the error coefficients with $Grad$ and the Step size Sp until the max epoch:

$$e(t) += S_p * Grad. \tag{14}$$

The optimized $e(t)$ comprehensively reflects the annotation error of each crowdsourced annotation by considering the information of local word collocation and global sentence embedding. It can be used to accurately select the best annotation for each task. We conduct a brief introduction to the annotation error consistency algorithm in *Algorithm* 1.

4.4. Annotation selection

After we derive the error coefficients e_t for each task t, we can select the best annotation with the minimum error:

$$S_t = \arg\min_{a_{t,j}} e_{t,j}, \tag{15}$$

where e_t is the error set of task t, S_t denotes the selected best annotation for task t with the minimum error coefficient.

Algorithm 1 AEC: Aggregation based on Error Consistency

Input: Task set T, local $GLEU$ embeddings emb_l, Universal Sentence embeddings emb_g, Global Annotation Distances Dis_g, Step size S_p, $epochs$;
Output: Final annotate error e for each annotation a;
initialize the annotate error e with Global Distances Dis_g;
That is: e_{init}=Dis_g;
for Task t in T **do**
  Select the max, min similarities $Max_{gleu}(t)$, $Max_{sen}(t)$, $Min_{gleu}(t)$, $Min_{sen}(t)$ for normalization,
  Represent the Estimated Truth $Etr_t(g)$, $Etr_t(l)$;
  for *epoch* in *epochs* **do**
    for Annotations $a_{t,j}$ in Task t **do**
      Compute the $Loss_j$;
    Compute the total $Loss_t = \sum_{j \in t} Loss_j$;
    $Grad$=$\frac{\partial Loss_t}{\partial e(t)}$
    $e(t)$ += S_p*$Grad$
return e

5. Experiments

5.1. Datasets

Complex open-ended tasks like content creation and translation are widely exist in the real world.

However, most of the existing datasets for open-ended tasks only contain multiple crowdsourced annotations, lack the golden annotations for performance evaluation. To better illustrate the effectiveness of our method in annotation aggregation for open-ended tasks, we adopt a real-world crowdsourced dataset[2] based on the translation task and a simulated dataset based on the question–answer task to evaluate the validity of our method. Detailed information about the datasets is provided below.

5.1.1. CrowdWSA: translation

The CrowdWSA[26] dataset is a collection of Japanese-to-English translations made by crowd workers on the real-world crowdsourcing platform.[3] The raw sentence pairs are collected from different Japanese-English parallel corpora, i.e., JEC Basic

[2] https://github.com/garfieldpigljy//CrowdWSA2019
[3] https://www.lancers.jp/.

Sentence Data[4] (one collection extracted, named as J1) and Tanaka[5] Corpus (two collections extracted, named as T1 and T2). Each task (source languages in Japanese) is asked for ten translations (named as annotations) by the crowd workers in the target language of English and one true answer collected from the initial *Japanese – English* parallel corpora. Particularly, the crowd workers on the crowdsourcing platform are mainly Japanese native speakers and non-native speakers of English, so the reliability of their translations is diverse.

5.1.2. ARC: question – answering

The ARC dataset is a collection of English question–answer collected from the *AI2 Reasoning Challenge* [9], raised by the Allen Institute for Artificial Intelligence.[6] It contains thousands of genuine grade-school level, text-only science questions. Four annotations (answers for the question) are provided for each task (questions in the collection) based on the advanced methods proposed in the research area of machine reading comprehension and QA systems[10,22], the true answer is included in the four annotations. We selected 400 task-annotation pairs to evaluate of our method. As shown in Table 4, the number of distinct words in the ARC dataset is significantly greater than that in the WSA dataset, which indicates much difficulty for annotations analysis. Considering the characteristics of this dataset like knowledge-driven, multi-grammatical, and vast feature space, experiments conducted on the ARC dataset can be a significant simulation for the research area like question–answering, machine reading comprehension, text summarization,etc.

5.2. Experiments settings

Unlike most existing aggregation methods for categorical or numerical annotations, we focus on the methods of analyzing the open-ended text annotations and trying to extract the optimal one from the crowdsourced annotations. Hence we have collated the latest methods that are equipped to analyze and aggregate the open-ended text annotations as baselines, and present them in detail below:

SMV. Majority voting is one of the most typical answer aggregation approaches. For the sequential data, Li and Fukumoto [26] adapt it into a **Sequence Majority Voting (SMV)** approach. For each task t, the true answer Tr_t is estimated by:

$$Tr_t = \frac{\sum_{i=1}^{m} emb_{t,i}}{m} \tag{16}$$

where $emb_{t,i}$ represent the embedding of the annotation $a_{t,i}$ for the task t, m denotes the number of workers for task t. Then, the estimated truth E_t are selected based on the embedding similarity:

$$E_t = \arg\max_{a_{t,j}} sim(e(a_{t,j}), Tr_t), \tag{17}$$

where $e(\cdot)$ represents the universal sentence encoder.

SMS. SMS (Sequence Maximum Similarity)[24] is proposed as a post-ensemble method for multiple summarization generation models. For each task t, the truth Tr_t are selected with the largest sum of similarity with other annotations of this task from the crowdsourced annotations a_t. The Sequence Maximum Similarity method can be formulated as:

$$E_t = \arg\max_{a_t} \sum_{j \neq k} sim(e(a_{t,j}), e(a_{t,k})), \tag{18}$$

where $a_{t,j}, a_{t,k}$ are different annotations for task t.

RASA. Differently from SMV and SMS, the method in RASA (Reliability Aware Sequence Aggregation)[26] iteratively estimates the reliability β_k for worker k and the truth Tr_t by:

$$\beta_k = \frac{\chi^2_{(\frac{\alpha}{2},|V_k|)}}{\sum (e(a_{t,k}) - e(Tr_t))^2}, \tag{19}$$

$$Tr_t = \frac{\sum \beta_k e(a_{t,k})}{\sum \beta_k}, \tag{20}$$

where $a_{t,k}$ is the annotation for the *task t* answered by the *worker k*. Specifically, χ^2 is the chi-squared distribution, and the significance level α is set as 0.05. The estimated truth Tr is initialized by the Sequence Majority Voting method. The annotation with max similarities to the embeddings of the estimated true answer is extracted as the estimated truth E.

MAS. The MAS (Multidimensional Annotation Scaling) Braylan and Lease [4] is a hierarchical Bayesian probabilistic model with a multidimensional scaling likelihood function. For annotation $a_{t,w}$, the inherited error $\varepsilon_{t,w}$ is inferred by the MAS model

[4] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JEC%20Basic%20Sentence%20Data.
[5] https://github.com/odashi/small_parallel_enja.
[6] https://allenai.org/data/arc.

Table 4
Datasets summary statistics.

Dataset	Type	Tasks	Workers	Annos	Annos/Task	UniqueWords
WSA_{J1}	Real	250	70	2490	9.96	1893
WSA_{T1}	Real	100	42	1000	10	746
WSA_{T2}	Real	100	43	1000	10	754
ARC	Simulated	400	–	1600	4	6329

in the embedding space with consideration of the capability of worker w and difficulty of task t. Then the estimated annotations are selected with the smallest inferred error ε_t out of all annotations. The method can be formulated as:

$$\varepsilon_{t,w}^{MAS} = \left\| emb_{t,w} \right\|, \tag{21}$$

$$w_t' = \arg\min_{w \in W_t} \varepsilon_{t,w}, \tag{22}$$

where $emb_{t,w}$ represent the embedding of annotation $a_{t,w}$, W_t denotes the group of workers answered the task t. Eq. 21 illustrates that the inherent error of annotations are derived with the annotation embedding and the truth was selected with minimal error, as shown in Eq. 22.

5.3. Experiment results

First, the experimental results presented in Table 5 show how well our model selects the best annotation from the crowdsourced annotations. It is evaluated by the similarity (*GLEU*[46] or $Rouge-1, Rouge-2$[29]) between the selected annotations and the truth. In summary, our method $AEC-C$ performs best for all experimental groups. For the WSA dataset, there are **at least 2.82%** performance improvements compared to the baseline (**further improved in the extension experiments**). It is worth noting that *RASA* performed better compared to *SMS* and *SMV*. All experimentsare based on embedding similarity but *RASA* considered the information of worker skill. This may indicate that the estimation of worker ability is still a key step for annotation aggregation.

Similar results presented on the ARC dataset. Our method achieved further improvement of **2.20% (Rouge-1)** and **1.93% (Rouge-2)** compared to the state-of-the-art result, while the performance of *MAS* is unexpectedly poor. After analyzing the MAS algorithm, we believe that the multidimensional scaling algorithm does not fully utilize the annotation information and has no enough capability to accurately learn the parameters of annotation representation, worker ability, and task difficulty at the same time. Hence it's vital to explore the parameter learning algorithm with fully utilize the rich information like word collocation and global sentence semantics in text annotations and that's why our method outperformed other models.

Ablation. As discussed in Section 4.3, the process of parameter estimation is divided into two parts. First, we initialize the error coefficient E with the global distances Dis_g. Then we iteratively optimize the annotation error coefficients with the *AEC* algorithm. Hence we conduct an ablation experiment to illustrate the effectiveness of our *AEC* algorithm.

As shown in Table 5, the $AEC-I(AEC-Initial)$ denotes our experiment result with initial error coefficients, $AEC-C(AEC-Consistency)$ represents the optimized experiment results with our *AEC* algorithm. To sum up, the performance achieved **3.65%**, **2.94%**, and **3.74%** improvement on WSA, ARC-1, and ARC-2 respectively after error optimization with the *AEC* algorithm. Particularly, both of *RASA* and $AEC-I$ analysis and aggregate annotations are based on the embedding similarity but *RASA* considered the information of workers' reliability, so we can observe that all the results of $AEC-I$ are lower than *RASA*. Beyond that, the drastic improvement between $AEC-C$ and *RASA* powerfully proves the importance of our *AEC* algorithm and the information integration of local word collocation and global sentence embedding.

Extension. The experiment results compared between *RASA* and $AEC-I$ (shown in Table 5) indicate the importance of the important factors like *worker capability* and *task difficulty*. Hence we conduct extension experiments to test the performance changes of our method combined with the *task difficulty*, *worker capability*, and both of them. The comparison results are shown in Table 6. We traverse the annotation distances for all tasks to estimate worker error across items:

$$C_w = \mathbf{Nor}\left(\sum_{t \in T} \sum_{m \in M} Sim(emb_g(a_{t,w}), emb_g(a_{t,m}))\right), (w \neq m) \tag{23}$$

Table 5
Experiment results compared with baselines. $AEC-I(AEC-Initial)$ denotes the experiment result of *AEC* with initial error coefficients, $AEC-C(AEC-Consistency)$ represents the optimised experiment results with our *AEC* algorithm.

Experiment		Baselines				Ablation		Oracle
Datasets	Evaluation	SMV	SMS	RASA[26]	MAS[4]	AEC-I	AEC-C	Oracle
WSA	GLEU	0.1762	0.2329	0.2418	0.2278	0.2384	**0.2534**	0.4108
ARC-1	Rouge-1	0.7913	0.7994	0.8038	0.7781	0.7964	**0.8258**	1.000
ARC-2	Rouge-2	0.5576	0.5844	0.5791	0.5612	0.5663	**0.6037**	1.000

Table 6
The results of extension experiments of *AEC*. $AEC-I$ represents the initial experiment results with our method, $AEC-T$ represents the experiment results with additional task difficulty information, $AEC-W$ denotes the experiment results with extra worker ability information, $AEC-B$ represents the experiment results with both of the information of task difficult and worker ability. Additionally, the evaluation metric is *GLEU* here.

Experiment			**AEC**		**Oracle**
Datasets	AEC-I	AEC-T	AEC-W	AEC-B	Oracle
WSA_{J1}	0.2685	0.2753	0.2721	**0.2772**	0.4990
WSA_{T1}	0.2499	0.2471	**0.2542**	0.2519	0.3698
WSA_{T2}	0.2418	0.2436	0.2537	**0.2623**	0.3637
Average	0.2534	0.2556	0.2600	**0.2638**	0.4108

where *Sim* is cosine similarity, M is the set of annotations for task t, *Nor* here is max–min normalization. We pass over the annotation distances for each task to estimate task difficulties:

$$D_t = \mathbf{Nor}\sum_{i,j\in W} Sim(a_{t,i}, a_{t,j})), (i \neq j) \tag{24}$$

Then we combine the information of worker error C_w and task difficulty D_t with the initial annotate error e_{init}. For the annotation $a_{t,w}$, the initial error coefficient in $AEC-B$ can be formulated as:

$$e_{init}(t,w) = \frac{Dis_g(t,w) * D_t}{C_w}, \tag{25}$$

the experiment results of *AEC* shown in Table 6 are obtained by different initialization and same *AEC* optimization. Considering that the ARC dataset does not have accurate information on workers and tasks, we only conduct the extension experiments on the WSA dataset. We can observe that almost all performance improvements can be achieved on the three sub-datasets after integrating the information of worker ability and task difficulty for parameter initialization, especially the $AEC-B$ for WSA_{T2}, **5.64%** performance improvement achieved. For the average situation, after combining the information of tasks difficulty and worker ability, our method achieved a **2.53%** performance improvement than before. What's more, **5.36%** performance improvement achieved than the state of the art method *RASA*.

To summarize, on the one hand, the extension experiment shows good scalability of our method, which could well integrate other important factors into the model to improve the aggregation results. In addition, much improvement of experiment performance further illustrates the effectiveness of our method for analyzing and aggregating open-end text annotations.

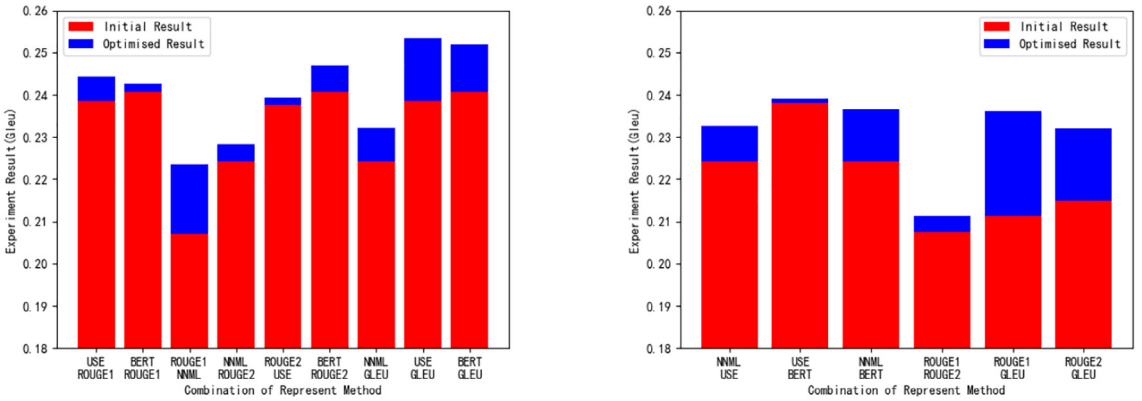
Why GLEU & ***Universal sentence encoder.*** In this paper, we optimize the error for each annotation by reducing the difference between the representation methods of *universal sentence encoder* and *gleu similarity*. We conduct further experiments to show the universality of this phenomenon. As discussed in Section 3, the representation methods can be classified into the following two categories:

(1) Evaluation metric based similarity: annotation relations are measured by the evaluation metrics, including $gleu, rouge-1$ and $rouge-2$.
(2) Distributed representation based similarity: annotation relations are computed by the similarity of their embedding vectors, the embedding methods are *nnml*[2], *universal sentence encoder* and *RoBERTa* (roberta-base-nli-mean-tokens) [14].

Two groups of comparative experiments are conducted. In the first group, nine comparative experiments are conducted with the combination of the two types of represent methods, each evaluate metric is combined with all the embedding methods. In the second group, six comparative experiments are conducted with method combination within the category, similar represent methods are combined to explore the performance changes. The experiment results shown in Fig. 4 mainly indicate that:

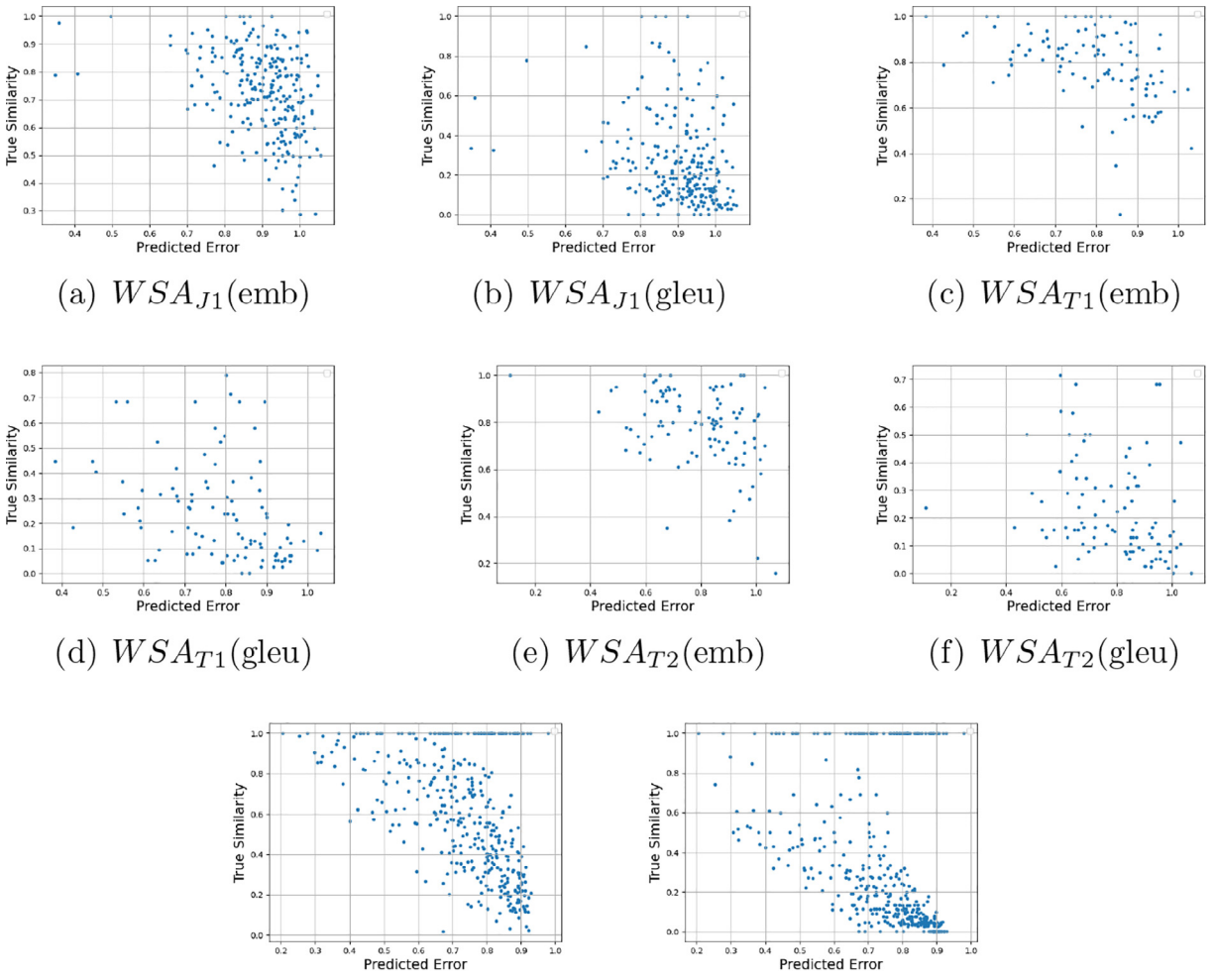
(1) The integration of *gleu* & *universal sentence encoder* achieved the best performance compared to other combinations (USE + GLEU), so we conduct this combination to illustrate the effectiveness of our method.
(2) The performance improvement (the blue part in Fig. 4) after error optimization indicate that information difference generally exists between different representation methods. Specifically, the method combination between the evaluation metrics and distributed representation methods usually gets more performance improvement than the combination within categories. For example, both "USE" and "BERT' have the advanced effect for distributed representation, but the experiment result of their combination is lower than "BERT" only. This phenomenon may indicate that information compensation mainly exists between the two categories of represent methods.

Similar results are shown on the *ARC* dataset. We show the results on the *WSA* dataset only due to the space limitation.



(a) External combination (b) Internal combination

Fig. 4. The experiment results with combination of different represent methods on WSA dataset.



(a) WSA_{J1}(emb) (b) WSA_{J1}(gleu) (c) WSA_{T1}(emb)

(d) WSA_{T1}(gleu) (e) WSA_{T2}(emb) (f) WSA_{T2}(gleu)

(g) ARC (emb) (h) ARC (gleu)

Fig. 5. The relation between predicted error and the true similarity evaluated by GLEU and Sentence Embedding.

Evaluation of the Prediction Quality. The above comparative experiments have proved the effectiveness of our method, mainly illustrates the ability to aggregate and select the best annotation. Beyond that, evaluating the relationship between the predicted error and the real error for each annotation is of great significance to intuitively illustrate how reliably the aggregation results are.

Given that the error coefficients of each annotation are derived by the distances from the true annotation, hence we consider the optimized global and local error as the predicted distances. The evaluation quality can be measured by how closely the predicted error for annotations correlates with actual truth similarity, where the actual results are computed by the embedding similarity and *gleu* similarity concerning the truth annotation.

The predict quality on the datasets used in the experiments are shown in Fig. 5, where the true similarities are represented by the *gleu* and *embedding* similarity respectively. In general, the obvious negative correlation between the predicted error and true similarity can be observed from these eight pictures. This is consistent with the experimental conclusion that the annotation with larger error from the truth should achieve lower similarity with the true annotation in actuality. Specifically, many of data point with the max true similarity occurred in subFig. 5 (g) and (h) is because that the truth exists in the optional annotations in the ARC dataset. The different results derived with the same algorithm may demonstrate the relative difficulty of aggregating. The simulation for the ARC dataset is easier than WSA, because the negative correlation is more obvious in subFig. 5 (g) and (h). In addition, this experimental result also shows that there is still a lot of room to explore better open-ended text aggregation methods to obtain more accurate correlations between the predicted error and true similarity.

6. Limitations & future work

In this paper, we mainly focus on the approach to aggregate answers in open-ended crowdsourcing. We proposed an *AEC* algorithm that extracts the potentially optimal annotation for each task from a set of open-ended answers. However, even the unselected annotations contain valuable knowledge, that also contributes to solve the task. So we believe that a new annotation generated by analyzing and understanding all of the crowdsourced annotations is possible to reach better results than any of the extracted ones. Therefore, it is interesting to explore the analyzing and aggregating method of open-end text annotations from the perspective of knowledge fusion and natural language generation in the future. In crowdsourcing, open-ended tasks are widely existed in the world, like content creation, translation, question–answering. However, only two datasets are used in this paper, we did not empirically test our algorithm on each representative task. Since the academic community is still in the early stages of exploring the problem of open-domain crowdsourcing, high-quality datasets for open-ended crowdsourcing are still a rare and scarce resource. In the next stage of our work, we will consider producing real datasets for representative open-domain crowdsourcing tasks, to facilitate research on open-ended crowdsourcing, as Li et al.[26] did in 2019.

The encoder and evaluation metric for textual annotations are not generally considered in this work. Some models are not considered (like $GPT-3$[5]) for several reasons, e.g., high overhead, low pervasiveness. Due to the lack of complete considerations and formal proofs, the conclusions presented in Table 1 and Fig. 4 can only be considered as a phenomenon. It is interesting to explore the effect of diverse embedding methods and evaluate metrics on text modeling in the future works.

7. Conclusion

Exploring the answer aggregation methods in open-ended crowdsourcing is of great significance to the investigation of complex AI tasks. Many knowledge-driven tasks like question answering, translation, reading comprehension can be better solved with the wisdom of crowds, much high-quality complex annotations will be generated to grow AI capabilities to accomplish more complex prediction tasks.

In this paper, we discuss and analyze the open-ended crowdsourcing problem starting from the observation that existing answer aggregation algorithms cannot handle the open-ended crowdsourcing problems well. Then we specify the core issues of open-ended crowdsourcing, and conduct a preliminary study on the research questions involved. After that, we propose an analyzing and aggregating framework for open-ended text annotations that goes beyond simple numerical or categorical annotations. To fully utilize the information of text annotations, we represent annotations in the feature spaces of local word collocation and global sentence embedding with annotation distances. Then we design the *AEC* algorithm to eliminate the inconsistency between different annotation representations and further derive the error coefficients for each annotation. Finally, the best annotation is selected with the minimum error for each task. The experimental results demonstrate the effectiveness of our method in deriving a satisfactory result from a set of open-ended answers. Beyond that, our method has good extensibility, which is of great significance to integrate future advanced embedding methods and effective factors like worker abilities, task difficulties to further improve the performance of answer aggregation.

CRediT authorship contribution statement

Lei Chai: Conceptualization, Methodology, Writing - original draft. **Hailong Sun:** Supervision, Writing - review & editing. **Zizhe Wang:** Conceptualization, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by National Natural Science Foundation under Grant Nos. (61932007, 62141209, 61972013).

References

[1] Aydin, B.I., Yilmaz, Y.S., Li, Y., Li, Q., Gao, J., Demirbas, M., 2014. Crowdsourcing for multiple-choice question answering. In: AAAI. Citeseer, pp. 2946–2953.
[2] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.
[3] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[4] A. Braylan, M. Lease, Modeling and aggregation of complex annotations via annotation distances, Proc. Web Conf. 2020 (2020) 1807–1818.
[5] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners.
[6] Cer, D., Yang, Y., Kong, S.-Y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., et al., 2018. Universal sentence encoder. arXiv preprint arXiv:1803.11175.
[7] S. Chatterjee, A. Mukhopadhyay, M. Bhattacharyya, Dependent judgment analysis: A markov chain based approach for aggregating crowdsourced opinions, Inf. Sci. 396 (2017) 83–96.
[8] P. Chen, H. Sun, Y. Fang, X. Liu, Conan: A framework for detecting and handling collusion in crowdsourcing, Inf. Sci. 515 (2020) 44–63.
[9] Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., Tafjord, O., 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457.
[10] Clark, P., Etzioni, O., Khot, T., Sabharwal, A., Tafjord, O., Turney, P., Khashabi, D., 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 30.
[11] A.P. Dawid, A.M. Skene, Maximum likelihood estimation of observer error-rates using the em algorithm, J. Roy. Stat. Soc.: Ser. C (Appl. Stat.) 28 (1) (1979) 20–28.
[12] G. Demartini, D.E. Difallah, P. Cudré-Mauroux, Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking, in: Proceedings of the 21st international conference on World Wide Web, 2012, pp. 469–478.
[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
[14] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[15] Y. Dong, L. Jiang, C. Li, Improving data and model quality in crowdsourcing using co-training-based noise correction, Inf. Sci. 583 (2022) 174–188.
[16] Du, G., Zhang, J., Jiang, M., Long, J., Lin, Y., Li, S., Tan, K.C., 2021. Graph-based class-imbalance learning with label enhancement. IEEE Trans. Neural Networks Learn. Syst., early access
[17] G. Du, J. Zhang, Z. Luo, F. Ma, L. Ma, S. Li, Joint imbalanced classification and feature selection for hospital readmissions, Knowl. Based Syst. 200 (106020) (2020) 1–12.
[18] J. Fan, G. Li, B.C. Ooi, K.-L. Tan, J. Feng, icrowd: An adaptive crowdsourcing framework, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015, pp. 1015–1030.
[19] Franz, A., Brants, T.,???? All our n-gram are belong to you (august 2006).
[20] T. Han, H. Sun, Y. Song, Z. Wang, X. Liu, Budgeted task scheduling for crowdsourced knowledge acquisition, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1059–1068.
[21] M.R. Jacobson, C.E. Whyte, T. Azzam, Using crowdsourcing to code open-ended responses: A mixed methods approach, Am. J. Evaluation 39 (3) (2018) 413–429.
[22] Khashabi, D., Khot, T., Sabharwal, A., Roth, D., 2018. Question answering as global reasoning over semantic abstractions. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32.
[23] Kim, H.-C., Ghahramani, Z., 2012. Bayesian classifier combination. In: Artificial Intelligence and Statistics. PMLR, pp. 619–627.
[24] H. Kobayashi, Frustratingly easy model ensemble for abstractive summarization, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4165–4176.
[25] J. Li, Crowdsourced text sequence aggregation based on hybrid reliability and representation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1761–1764.
[26] J. Li, F. Fukumoto, A dataset of crowdsourced word sequences: Collections and answer aggregation for ground truth creation, in: Proceedings of the First Workshop on Aggregating and Analysing Crowdsourced Annotations for NLP, 2019, pp. 24–28.
[27] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, J. Han, A confidence-aware approach for truth discovery on long-tail data, Proc. VLDB Endowment 8 (4) (2014) 425–436.
[28] S.-Y. Li, S.-J. Huang, S. Chen, Crowdsourcing aggregation with deep bayesian learning, Sci. China Inform. Sci. 64 (3) (2021) 1–11.
[29] Lin, C.-Y., 2004. Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81.
[30] N. Littlestone, M.K. Warmuth, The weighted majority algorithm, Inform. Comput. 108 (2) (1994) 212–261.
[31] B. Liu, Sentiment analysis and opinion mining, Synthesis lectures on human language technologies 5 (1) (2012) 1–167.
[32] J. Liu, F. Tang, L. Chen, Y. Zhu, Exploiting predicted answer in label aggregation to make better use of the crowd wisdom, Inf. Sci. 574 (2021) 66–83.
[33] Liu, Q., ICS, U., Peng, J., Ihler, A., 2012. Variational inference for crowdsourcing. sign 10, j2Mi.
[34] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, J. Han, Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation, in: Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining, 2015, pp. 745–754.
[35] Nguyen, A.T., Wallace, B.C., Li, J.J., Nenkova, A., Lease, M., 2017. Aggregating and predicting sequence labels from crowd annotations. In: Proceedings of the conference. Association for Computational Linguistics. Meeting. Vol. 2017. NIH Public Access, p. 299.
[36] A. Parameswaran, A.D. Sarma, V. Venkataraman, Optimizing open-ended crowdsourcing: the next frontier in crowdsourced data management, Bull. Tech. Committee Data Eng. 39 (4) (2016) 26.
[37] A.G. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, J. Widom, Crowdscreen: Algorithms for filtering data with humans, in: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 2012, pp. 361–372.
[38] S. Paun, D. Hovy, Proceedings of the first workshop on aggregating and analysing crowdsourced annotations for nlp, in: Proceedings of the First Workshop on Aggregating and Analysing Crowdsourced Annotations for NLP, 2019.

[39] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
[40] L.S. Penrose, The elementary statistics of majority voting, J. Roy. Stat. Soc. 109 (1) (1946) 53–57.
[41] V.C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, J. Mach. Learn. Res. 11 (4) (2010).
[42] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, M. Shokouhi, Community-based bayesian aggregation models for crowdsourcing, in: Proceedings of the 23rd international conference on World wide web, 2014, pp. 155–164.
[43] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum, recaptcha: Human-based character recognition via web security measures, Science 321 (5895) (2008) 1465–1468.
[44] Welinder, P., Branson, S., Perona, P., Belongie, S., 2011. The multidimensional wisdom of crowds. Neural Information Processing Systems
[45] J. Whitehill, T.-F. Wu, J. Bergsma, J. Movellan, P. Ruvolo, Whose vote should count more: Optimal integration of labels from labelers of unknown expertise, Adv. Neural Inform. Process. Syst. 22 (2009) 2035–2043.
[46] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al., 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.
[47] X. Zhang, X. Chen, H. Yan, Y. Xiang, Privacy-preserving and verifiable online crowdsourcing with worker updates, Inf. Sci. 548 (2021) 212–232.
[48] W.X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, X. Li, Comparing twitter and traditional media using topic models, in: European conference on information retrieval. Springer, 2011, pp. 338–349.
[49] Y. Zheng, G. Li, Y. Li, C. Shan, R. Cheng, Truth inference in crowdsourcing: Is the problem solved?, Proc VLDB Endowment 10 (5) (2017) 541–552.