Cost-efficient Crowdsourcing for Span-based Sequence Labeling: Worker Selection and Data Augmentation

Anonymous ACL submission

Abstract

Crowdsourcing is a scalable data collecting method used in many NLP tasks. Due to the disparity of expertise among crowd workers, prior studies utilize worker selection to improve the quality of the crowdsourced dataset. However, most of them are designed for and tested on simple classification tasks. In this paper, we focus on span-based sequence labeling tasks in NLP, which are more challenging as nearby labels have complex inter-dependencies. We propose a new worker selection algorithm based on combinatorial multi-armed bandit (CMAB). Our algorithm maximizes the quality of the annotations while reducing the overall cost by using both majority-voted and expert annotations for evaluations. A key challenge is that practical datasets are highly imbalanced and of small scale, which makes offline simulation of worker selection difficult. To address this issue, we present a novel data augmentation method called *shifting*, *expanding*, and *shrinking* (SES), which is customized for sequence labeling. We augment two datasets, CoNLL 2003 NER and Chinese OEI, on which we extensively test our worker selection algorithm. The results show that our algorithm achieves up to 100.04% F_1 score compared with an expert-evaluation-only (i.e., all annotations evaluated by experts) baseline, saving up to 65.97% of costs to ask experts. We also include a dataset-independent test in which the annotation evaluation is simulated through a Bernoulli distribution. Similarly, our algorithm achieves 97.56% F_1 and saves 59.88% expert costs.

1 Introduction

Crowdsourcing is obtaining labeled data from crowd workers (Howe, 2006). Several online crowdsourcing platforms have emerged and prospered in recent years, such as Amazon Mechanical Turk[1] and Taskrabbit[2]. Prior studies have applied

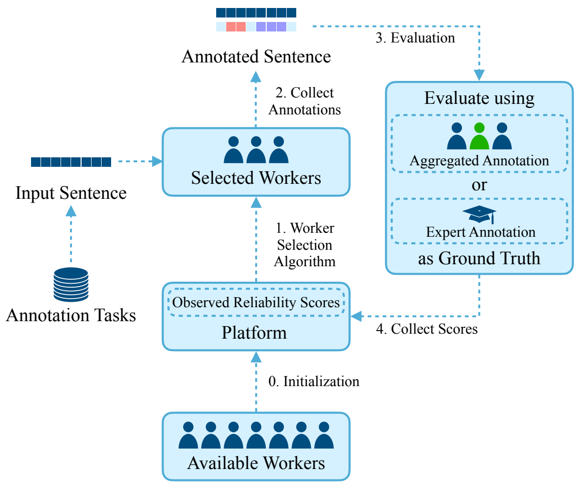


Figure 1: Our online worker selection framework for crowdsourcing.

crowdsourcing to collect data for a wide range of tasks including image labeling, text classification, and sequence labeling (Venanzi et al., 2014). Generally, one can reduce the cost and improve the efficiency of label collection by hiring crowd workers instead of expensive experts (Nowak and Rüger, 2010). With these advantages, crowdsourcing has gained great interest and played an important role in data collection for deep learning models.

In this paper, we focus on crowdsourcing for span-based sequence labeling tasks. Sequence labeling involves determining a categorical label to each word in a sentence (Erdogan, 2010). Many tasks come in the form of span-based sequence labeling, including named entity recognition (NER) and opinion expression identification (OEI) (Collobert et al., 2011). In simple sentence classification tasks, labels are assigned independently. While in sequence labeling tasks, nearby labels have inter-dependencies and are attached to the context (Rodrigues et al., 2014). This makes sequence labeling tasks more difficult, and annotations from crowd workers less accurate. Therefore, improving annotation quality becomes an important and challeng-

[1] https://www.mturk.com/
[2] https://www.taskrabbit.com

ing problem.

On span-based sequence labeling tasks, prior studies Rodrigues et al. (2014); Nguyen et al. (2017); Simpson and Gurevych (2019) mainly focus on annotation aggregation. These methods are used *after* data collection. Due to the disparity of skill levels among crowd workers, it could help to improve data quality if we can identify and utilize workers with the highest accuracy *during* data collection. This approach is known as online worker selection to which we resort in this paper. In online worker selection, the platform allocates a limited budget between a set of workers iteratively to maximize the quality of annotations (Chen et al., 2013). The skill level of the workers is unknown a priori and observed through annotations, which leads to a tradeoff between exploring new workers and exploiting the best workers at the moment. A feedback signal is required to update the learning process. This is easy in simple classification tasks where there is a binary feedback signal (i.e., correct or incorrect). But for sequence labeling tasks, it is much more difficult to define such a binary signal due to correlations of nearby labels. To address this, we use the span-level F_1 score (Derczynski, 2016) to measure the quality of annotations and it serves as the feedback signal in the worker selection process. The F_1 score is usually calculated based on expert-provided ground truth. Then we can formulate worker selection as an optimization problem that maximizes the overall F_1 score of the produced annotations.

While expert annotations tend to be of high quality as ground truth, they usually come at a large cost. Moreover, they are only available on a small portion of the task sequences. In this paper, we employ crowd workers for sequence annotations, of which the aggregation may serve as ground truth. We aim to replace as many expert ground truth labels as possible with aggregated crowd ground truth labels, while the overall F_1 score of the produced dataset remains high. An expert ground truth is replaced only when the inter-annotator agreement (i.e., Fleiss' Kappa (Fleiss, 1971)) among crowd workers is high enough. The intuition is that the sequence is easy to correctly annotate for a majority of crowd workers, which hence, does not require expert evaluation. Our worker selection algorithm is illustrated in Figure 1. We iteratively assign tasks to a subset of available workers, evaluate their annotations, and use the scores as a criterion of worker selection in future rounds. Detailed descriptions are deferred to Section 3.1.

We evaluate our worker selection algorithm on two datasets (Rodrigues et al., 2014; Zhang et al., 2022). However, real datasets are imbalanced and of small scale which may fail our worker selection algorithm. Firstly, a reasonable number (e.g., 3) of annotations on each sentence is required since we aggregate crowd annotations by majority voting (MV). Secondly, online algorithms (e.g, CMAB) require a relatively large number (e.g., thousands) of iterations to converge on a near-optimal set of workers (Chen et al., 2013). To address these issues, we design a data augmentation method for span-based sequence labeling datasets. The main purpose of our augmentation method is to reflect the possible errors when human workers give annotations in practice. Thus, using generated annotations in MV will not lead to meaningless aggregation results. We propose three kinds of modifications, *shifting*, *expanding*, and *shrinking* to the expert annotation on each sentence. For each sentence, we generate all possible annotations human workers might give. For each worker, we select one annotation on each sentence to make sure the average F_1 score is very close to the worker's F_1 score calculated on the real dataset. Our augmentation method solves the imbalance and insufficiency problem in real datasets, enabling offline evaluation of worker selection algorithms.

The main contributions of our paper are summarized as follows:

- To our best knowledge, we present the initial work of worker selection on span-based sequence labeling tasks. This is critical as such tasks are more challenging and crowd workers produce less reliable annotations compared with simple classification tasks.

- Due to label inter-dependencies, simple binary feedback is not applicable on span-based sequence labeling tasks. We utilize the span-level F_1 score evaluated by experts and crowd workers combined as the feedback signal, which is shown to precisely reflect the worker accuracy and hence effectively guide the worker selection process.

- We propose a data augmentation method to address the imbalance and insufficiency of real datasets, enabling offline simulation of worker selection.

- We conduct extensive experiments on the augmented datasets. We use expert-evaluation-only as the baseline comparison, which is expected to generate the highest F_1 score. On the Chinese OEI dataset, our method achieves up to 99.47% F_1 score with 47.19% reduction in the expert cost. On the CoNLL 2003 NER dataset, our method achieves up to 100.04% F_1 score with 65.97% reduction in the expert cost.

We have all of our source codes and datasets released for research purposes[3].

2 Related Work

Many studies (Rodrigues et al., 2014; Rodrigues and Pereira, 2018; Nangia et al., 2021) have used crowdsourcing for its efficiency and scalability. However, crowdsourcing suffers from the diversity of crowd workers' expertise and effort levels that are hardly measurable to task requesters. Different approaches to improving the quality of collected data have been proposed and studied. For span-based sequence labeling tasks, prior studies mainly focus on annotation aggregation. Rodrigues et al. (2014) proposed CRF-MA, a CRF-based model with an assumption that only one worker is correct for any label. HMM-crowd from Nguyen et al. (2017) outperforms CRF-MA, but the effect of sequential dependencies is not taken into account. Simpson and Gurevych (2019) uses a fully Bayesian approach BSC which is proved to be more effective in handling noise in crowdsourced data. Aggregation methods are used *after* the data collection process completes. But we aim to assure data quality and reduce cost *during* collecting. To this end, we focus on worker selection in our paper.

In online worker selection, we need to balance between exploring new workers and exploiting observed good workers. This exploration-exploitation tradeoff is extensively studied in the bandit literature (Lai and Robbins, 1985). In practice, we usually employ multiple crowd workers at the same time to finish the tasks more effectively. The combinatorial multi-armed bandit (CMAB) (Chen et al., 2013) models this circumstance. Biswas et al. (2015); Rangi and Franceschetti (2018) reformulate the problem as a bounded knapsack problem (BKP) and address it with the B-KUBE (Tran-Thanh et al., 2014) algorithm. Song and Jin (2021) introduce empirical entropy as the metric in CMAB and minimize the cumulative entropy with upper confidence bound (UCB) based algorithm. Li et al. (2022) consider the scalability of worker selection on large-scale crowdsourcing systems. These studies propose different methods under the CMAB settings, but on more complex span-based sequence labeling tasks there exists no discussion. We present the study of worker selection with CMAB on span-based sequence labeling tasks and show that our work performs well on the quality and efficiency of data collection.

3 Methodology

3.1 System Overview

Consider an online crowdsourcing system that can reach out to a group of crowd workers $W = \{w_1, w_2, \ldots, w_N\}$. The workers are required to provide sequential annotations to a set of sentences $S = \{s_1, s_2, \ldots, s_M\}$. More specifically, a worker annotates a sentence by assigning a tag from a finite possible tag set C (e.g., a set of BIO tags (Ramshaw and Marcus, 1995)) to each word. An annotation on sentence s_i by worker w_j is a tag sequence $\boldsymbol{a}_{ij} = a_1 a_2 \ldots a_k \ldots a_l$ where $a_k \in C$ and l denotes the length of the sentence. We assume that every sentence is annotated by K different workers independently. We define a task as the process of annotating one entire sentence, and hence there are in total KM tasks. We seek to acquire an annotated dataset in which the average F_1 score of $\boldsymbol{a}_{ij}$ is maximized. If we know which workers give the best annotations a priori, we can simply ask these workers to finish all the tasks. However, such information is unavailable in practice, and we aim to design an algorithm that learns the best workers throughout the crowdsourcing process.

In the beginning, we let each crowd worker annotate one sentence. We also ask the experts(e.g., well-trained linguists assumed to give the most precise annotations) to give one annotation for each of these sentences. Then we calculate the F_1 score of the annotation with the expert annotations as ground truth. We use these scores as the initial F_1 scores of workers. At each time step t after initialization (as illustrated in Figure 1), we select a subset of workers $W_t \subset W$ to do annotation, based on criteria discussed in Section 3.3. The size of the subset W_t should be neither too big nor too small (e.g., $0.3N$). We randomly choose a subset of sentences $S_t \subset S$, assign each $s_i \in S_t$ to K dif-

[3] https://anonymous.4open.science/r/Costw-225D

ferent workers in W_t, and collect their annotations $\boldsymbol{A}_i = \{\boldsymbol{a}_{i1}, \boldsymbol{a}_{i2}, \ldots, \boldsymbol{a}_{iK}\}, \forall i \in \{1, 2, \ldots, |S_t|\}$. To evaluate workers' F_1 scores on $\boldsymbol{A}_i$, one can use the expert annotations as the ground truth, which, however, can be very expensive (İren and Bilgen, 2014). To cut down this cost, we reduce the usage of expert evaluations whenever crowd annotations are similar enough. We use the Fleiss' Kappa score κ to measure this similarity. The κ score ($\kappa \leq 1$) is a statistical measure of inter-annotator agreement. A larger value of κ indicates stronger agreement between the workers. κ score exceeding an empirical threshold indicates that the crowd workers reach a consensus on s_i. In that case, we aggregate $\boldsymbol{A}_i$ with MV and use the aggregated annotation as the ground truth of sentence s_i. If the workers do not reach a consensus, we resort to expert annotations as ground truth. Next, we can calculate the F_1 scores of each $\boldsymbol{a}_{ij} \in \boldsymbol{A}_i$ and update the F_1 scores of the selected workers.

3.2 Problem Formulation

At time step t, we obtain K crowd annotations $\boldsymbol{A}_i$ on each sentence $s_i \in S_t$. We denote all annotations collected on S_t by $\boldsymbol{\mathcal{A}}_t = \{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_{|S_t|}\}$. To simplify our expression, we use $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{ij})$ to represent the F_1 score of $\boldsymbol{a}_{ij}$ using expert annotation as ground truth, and $\mathrm{F}_1^{\mathrm{MV}}(\boldsymbol{a}_{ij})$ to represent the F_1 score of $\boldsymbol{a}_{ij}$ using the MV aggregation of $\boldsymbol{A}_i \in \boldsymbol{\mathcal{A}}_t$ as ground truth. On collected annotation sets, $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{A}_i)$ denotes the average F_1 score of all $\boldsymbol{a}_{ij} \in \boldsymbol{A}_i$. Similarly, $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{\mathcal{A}}_t)$ denotes the average F_1 score of all $\boldsymbol{A}_i \in \boldsymbol{\mathcal{A}}_t$. As $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{\mathcal{A}}_t)$ reflects the true accuracy of crowd annotations, our objective is to maximize the average expectation, or equivalently the cumulative expectation of $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{\mathcal{A}}_t)$ over time T. We formulate this problem as a CMAB problem below:

$$\max \quad \sum_{t=1}^{T} \mathbb{E}[\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{\mathcal{A}}_t)] \tag{1}$$

$$\text{s.t.} \quad W_t \subset W, t \in \{1, 2, \ldots, T\} \tag{2}$$

Since we have no information about workers' average F_1 scores, we need to balance exploring potentially better workers and exploiting the current best workers during worker selection. This tradeoff is extensively discussed in bandit literature where arms with unknown distributions form super-arms. The arms are associated with a set of random variables $X_{j,t}$ with bounded support on [0, 1]. Variable $X_{j,t}$ indicates the random outcome of arm j in time step t. The set of random variables $\{X_{j,t} | t \geq 1\}$ associated with arm j are independent and identically distributed according to certain unknown distribution D_j with unknown expectation $\bar{\mu}_j$. The platform plays a super-arm at each time step, and the reward of arms in it is revealed. These rewards are used as a metric for selecting the super-arm in future time steps. After enough time steps, the platform will be able to identify the best super-arm and keep playing it to maximize the overall reward. Similar to bandit terminologies, we call each worker $w_j \in W$ an arm and the worker subset $W_t \subset W$ a super-arm selected at t.

3.3 Worker Selection Algorithm

Specifically, there are three methods to calculate the reward of worker w_j at time step t as follows.

Expert Only This is a benchmark approach where the F_1 score is calculated using only expert annotations as ground truth. This method provides intuitively the most accurate F_1 scores. The reward of worker w_j is defined as:

$$\mu_j^{\mathrm{Exp}}(t) = \mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{ij}(t)) \tag{3}$$

The expert-only method requires an expert annotation on every sentence, which is costly and usually not practical.

Majority Voting (MV) To reduce expert annotations, we aggregate $\boldsymbol{A}_i$ for each sentence s_i, and use the aggregated annotation via MV as ground truth, i.e.,

$$\mu_j^{\mathrm{MV}}(t) = \mathrm{F}_1^{\mathrm{MV}}(\boldsymbol{a}_{ij}(t)) \tag{4}$$

Expert+MV When a task is difficult, workers may give very different annotations on the same sentence, and one can be uncertain about the voted (and possibly noisy) ground truth. In this case, we want to resort to both crowd workers and experts. The choice is based on the well-known Fleiss' Kappa score κ that can quantitatively evaluate the agreement of crowd workers. For each sentence s_i, if $\kappa(\boldsymbol{A}_i)$ is greater than a preset empirical threshold value τ, the reward of annotating workers is $\mathrm{F}_1^{\mathrm{MV}}(\boldsymbol{a}_{ij}(t))$. Otherwise, the reward is $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{ij}(t))$. In this way, MV is only used when the crowd workers can reach an agreement. Thus the reward is always calculated based on reliable

Algorithm 1 The worker selection algorithm with the Expert+MV metric.

Let each worker $w_j \in W$ annotate a random sentence and initialize variable $\bar{\mu}_j$ with F_1 by expert evaluation
For each worker $w_j \in W$, initialize $T_j \leftarrow 1$
$t \leftarrow |W|$
while unannotated sentences exist **do**
  $t \leftarrow t + 1$
  Select $W_t \subset W$ based on certain criterion (e.g., (6), (7))
  Split W_t into several disjoint subsets $\{W_{t1}, \ldots, W_{ti}, \ldots, W_{tn}\}$, each containing K workers
  for all W_{ti} **do**
    Let each $w_j \in W_{ti}$ annotate an sentence s_i and collect the annotations $\boldsymbol{A}_i$
    if $\kappa(\boldsymbol{A}_i) > \tau$ **then**
      Update T_j and $\bar{\mu}_j$ with $\mathrm{F}_1^{\mathrm{MV}}(\boldsymbol{a}_{ij}(t))$
    else
      Update T_i and $\bar{\mu}_j$ with $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{ij}(t))$
    end if
  end for
end while

ground truth. We summarize the reward of worker w_j as:

$$\mu_j^{\text{Exp+MV}}(t) = \begin{cases} \mathrm{F}_1^{\mathrm{MV}}(\boldsymbol{a}_{ij}(t)), & \kappa(\boldsymbol{A}_i) > \tau \\ \mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{ij}(t)), & \kappa(\boldsymbol{A}_i) \leq \tau \end{cases} \quad (5)$$

The ϵ-Greedy and Combinatorial Upper Confidence Bound (CUCB) are two effective algorithms to solve the CMAB problem. For each worker $w_j \in W$, both algorithms maintain a variable $\bar{\mu}_j(t)$ as the average reward (i.e., the average F_1 score) of worker w_j at time step t. CUCB additionally maintains a variable $T_j(t)$ as the total number of sentences worker w_j has annotated till time step t. Details of the worker selection algorithm with our **Exp.+MV** metric are shown in Algorithm 1. As for the selection criterion mentioned in the algorithm, ϵ-Greedy utilize a hyper-parameter ϵ which refers to the probability of exploring random workers. Thus $1 - \epsilon$ refers to the probability of exploiting the best workers till the current time step. Formally, W_t is selected with a random variable $p \in [0, 1]$ as below:

$$W_t = \begin{cases} \text{random } W_t \subset W, & p < \epsilon \\ \underset{W_t \subset W}{\operatorname{argmax}} \sum_{w_j \in W_t} \bar{\mu}_j, & p \geq \epsilon \end{cases} \quad (6)$$

CUCB handles the tradeoff by adding an item considering T_j and t to $\bar{\mu}_j$ like:

$$W_t = \underset{W_t \subset W}{\operatorname{argmax}} \sum_{w_j \in W_t} \left(\bar{\mu}_j + \sqrt{\frac{3 \ln t}{2 T_j}} \right) \quad (7)$$

This makes workers with less annotations more likely to be selected as the algorithm proceeds. We provide a brief analysis in Appendix B.

3.4 Data Augmentation Method

CMAB-based algorithms require a relatively large number (e.g., thousands) of iterations to converge on selecting a near-optimal set of workers. Hence real datasets can be insufficient on scale. In the best case, the algorithm always selects the same best super-arm at every time step t. Therefore, we need to ensure that these workers have annotations on every sentence in the dataset. Generating the missing annotations for each worker w_j is a great challenge when we expect the generated annotations to reflect the factual reliability of w_j. In other words, we expect the average F_1 score of each $w_j \in W$ to remain constant before and after augmenting the dataset with generated annotations. This is critical and difficult since real datasets are imbalanced and of small scale that cannot well support worker selection algorithms.

As there is no work on generating missing annotations, we start with several naive algorithms such as randomly generating label sequences as annotations, and mixing expert annotations with completely incorrect (e.g., empty) annotations. But these algorithms either cannot produce annotations with expected F_1 scores, or generate confusing annotations which make later aggregation meaningless. This motivates us to design a data augmentation method specialized for span-based sequence labeling datasets. For each sentence $s_i \in S$, we modify the annotation span based on the expert annotation. We use three types of modifications to generate new annotation spans with different F_1 scores as illustrated in Figure 2. The goal of these modifications is to simulate varying annotation errors made by human annotators.

Shifting We move both the left and the right border of the annotation span simultaneously in the same direction by one word per step.

Expanding We set one of the span borders fixed, and move the other border by one word per step to *increase* the length of the annotation span.

Dataset	#Sent.	#Antr.	#Antr. /Sent.	#Sent. /Antr.	Span Length
Chinese OEI	8047	70	3.2	368	5.05
CoNLL 2003	4580	47	3.6	350	1.51

Table 1: Statistics of datasets. **Sent.** stands for sentence. **Antr.** stands for annotator. Numbers of annotators per sentence, numbers of annotated sentences per annotator, and span lengths are means.

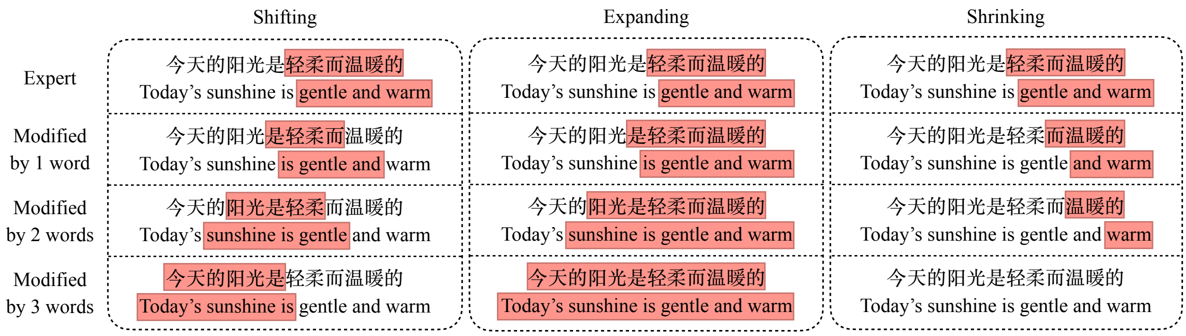


Figure 2: An example of the three methods to generate annotations. Chinese characters and corresponding English words with red backgrounds indicate annotation spans.

Shrinking We set one of the span borders fixed, and move the other border by one word per step to *decrease* the length of the annotation span.

We perform these modifications on a span multiple times, generating new annotation spans, until (1)the modified span does not overlap with the original one, (2)one of the span borders reaches an end of sentence or another span in the same sentence, or (3) the span length becomes 0.

For each sentence $s_i \in S$, s_i may contain multiple annotation spans. We perform modifications on each span in s_i, and find all combinations of spans to form possible sentence annotations. With these methods, we can imitate crowd annotations with different kinds of errors in practice. Next, for each worker $w_j \in W_{ti}$, if w_j has no annotation on s_i in the original dataset, we select one from all the expert and generated annotations on s_i. We first calculate $\bar{\varphi}_j$ as the average F_1 score of all annotations by w_j on the original dataset, and then follow the detailed steps described in Algorithm 2 to do the selection. We aim to keep the overall F_1 score of w_j unchanged.

4 Experiments

4.1 Original Datasets

We compare our CMAB-based algorithms to several widely adopted baselines on two span-based sequence labeling datasets.

CoNLL 2003 The CoNLL 2003 English named-entity recognition dataset (Tjong Kim Sang and De Meulder, 2003) is a collection of news article from Reuters Corpus(Lewis et al., 2004). The dataset contains only expert annotations for four named entity categories (PER, LOC, ORG, MISC). Rodrigues et al. (2014) collected crowd annotations on 400 articles from the original dataset.

Chinese OEI The Chinese OEI dataset (Zhang et al., 2022) consists of sentences on the topic of COVID-19 collected from Sina Weibo[4], in which the task is to mark the spans of opinion expressions. The Chinese OEI dataset contains expert and crowd labels for two opinion expression categories (POS, NEG). Detailed statistics are shown in Table 1.

4.2 Data Augmentation Results

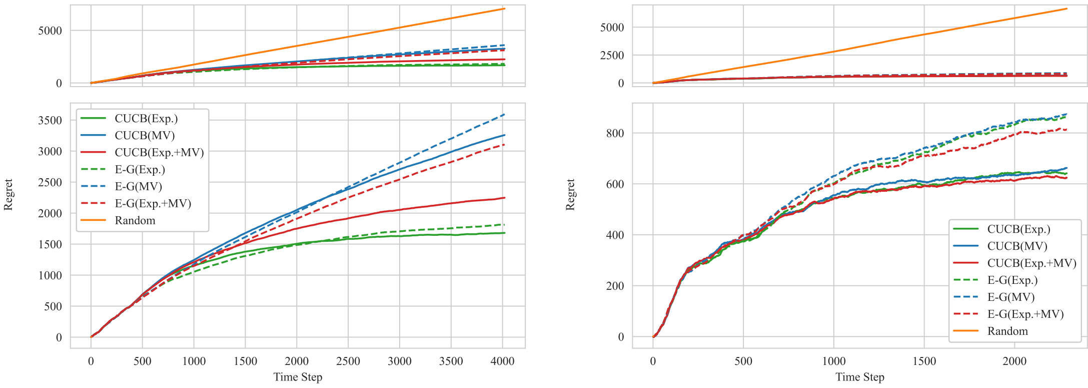
We augment both datasets with the method proposed in Section 3.4. Through our method, the average F_1 score of each $w \in W$ remains nearly unchanged before and after augmenting the original dataset with generated annotations[5]. Due to space limitation, we present the comparisons of different augmentation methods in Table 5 in the appendix, which shows that our method clearly outperforms the others.

4.3 Main Results

We test the **Exp.+MV** method with 4 baselines: **Oracle**, **Random**, **Exp.**, and **MV**. **Oracle** always

[4]https://english.sina.com/weibo/

[5]The augmentation procedure takes about 2 hours on a computer with a 2.9 GHz Quad-Core Intel Core i7 CPU.



(a) Chinese OEI

(b) CoNLL 2003

Figure 3: Cumulative regrets w.r.t time steps of all different worker selection methods.

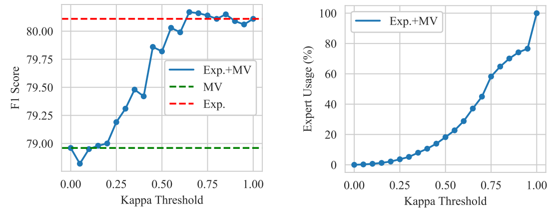
selects the empirical best super-arm W^{opt} at every time step t. **Random** selects a different set of workers randomly at every time step t. **Exp.**, **MV**, and **Exp.+MV** are CMAB-based algorithms introduced in Section 3.3. The CMAB-based algorithms are tested with CUCB and ϵ-Greedy as the worker selection criterion respectively.

We first examine the performance of our worker selection algorithms by the cumulative regret defined as:

$$R(T) = \sum_{t=1}^{T} \left(\sum_{w_j \in W^{opt}} \bar{\mu}_j - \sum_{w_k \in W_t} \mu_k(t) \right) \quad (8)$$

The regret reveals to what extent the super-arm selected by a certain algorithm is worse than the one selected by the oracle. In the experiment, we request 10 annotations on each sentence to ensure that the CMAB-based algorithms can converge. We set the size of the super-arm to 20, i.e., 20 workers are selected in each time step t. On the Chinese OEI dataset, we set the kappa threshold τ in **Exp.+MV** to 0.4, which results in 57.02% reduction of expert annotation cost. On the CoNLL 2003 dataset, we set the kappa threshold to 0.65, resulting in 43.83% reduction of expert annotation cost. The kappa thresholds are adjusted to different values so that **Exp.+MV** would perform the best respectively on these two datasets.

Figure 3 shows that **Random** is constantly worse than all other methods on both datasets. On the Chinese OEI dataset, **Exp.+MV** outperforms **MV** steadily. **Exp.+MV** produces greater regret compared with **Exp.**, but it is acceptable since we cut down up to 57.02% expert cost. On the CoNLL 2003 dataset, **Exp.+MV** even works better than



(a) F_1 score w.r.t τ

(b) Expert usage w.r.t τ

Figure 4: F_1 scores of the produced annotations and usage of expert for annotation evaluations w.r.t the kappa threshold τ of the **Exp.+MV** method on the CoNLL 2003 dataset.

Exp.. This indicates on simpler tasks like NER, crowd workers may provide extra intelligence compared with experts. Besides, we find that algorithms work better with the CUCB criterion rather than ϵ-Greedy. In short, **CUCB(Exp.+MV)** outperforms other baselines with cumulative regret and expert cost both considered.

Next, we discuss how different kappa threshold values τ affect the average F_1 score of the produced annotation dataset. We test $\tau \in [0, 1]$ with a step of 0.05. In real datasets like CoNLL 2003 and Chinese OEI, the number of annotations per sentence is often quite small. To better fit the practical situations, we ask for 4 annotations on each sentence in the following experiments. Other settings remain unchanged. Since CUCB performs better than ϵ-Greedy on both datasets, we display only the results from CUCB in later experiments.

On the Chinese OEI dataset, as illustrated in Figure 5, F_1 increases sharply with $\tau \in [0, 0.4]$. When $\tau = 0.4$, **Exp.+MV** achieves 99.47% F_1 score of **Exp.**, and saves 47.19% of the expert cost.

Method	Token-level			Span-level Exact			Span-level Prop.		
	P	R	F_1	P	R	F_1	P	R	F_1
Oracle	65.69	83.99	70.00	78.15	72.23	74.96	87.97	80.03	83.82
Random	55.95	66.42	57.50	64.42	55.64	59.40	75.70	62.61	68.54
ϵ-G(Exp.)	64.94	80.48	**68.56**	75.24	68.16	**71.34**	85.85	76.79	**81.06**
ϵ-G(MV)	64.44	80.22	67.98	74.69	67.59	70.77	85.67	76.09	80.59
ϵ-G(Exp.+MV)	64.68	80.94	**68.41**	75.08	68.37	**71.40**	85.93	76.62	**81.01**
CUCB(Exp.)	65.65	80.34	**69.24**	75.94	69.12	**72.20**	86.17	77.22	**81.45**
CUCB(MV)	65.39	80.00	68.91	75.95	68.90	72.08	86.13	76.67	81.12
CUCB(Exp.+MV)	65.33	81.12	**69.11**	75.70	69.30	**72.21**	86.17	77.28	**81.48**

Table 2: Detailed P, R, and F_1 scores of all methods on the CoNLL 2003 dataset.

The F_1 score goes up slowly until τ reaches 0.8. When $\tau = 0.8$, the F_1 score of **Exp.+MV** becomes exactly the same as the one of **Exp.**, and **Exp.+MV** still saves 6.6% of the expert cost.

The results on the CoNLL 2003 dataset are shown in Figure 4. Similarly, the F_1 score of the produced annotation dataset grows fast as $\tau \in [0, 0.45]$. When $\tau = 0.45$, the **Exp.+MV** method already produce an annotation dataset with its F_1 reaching 99.86% of **Exp.**. At this point, **Exp.+MV** saves 88.57% of the expert cost. When $\tau = 0.65$, **Exp.+MV** outperforms **Exp.** with a 100.04% F_1 score and a 65.97% reduction in expert usage.

Previous results show that with our **CUCB(Exp.+MV)** worker selection algorithm, we do not need to ask the experts to evaluate crowd annotations on every sentence. Instead, we propose to utilize crowd intelligence for annotation evaluations through our kappa-thresholded MV. And the dataset produced by our method is of nearly the same or even higher quality compared with using only expert evaluations.

All of the F_1 scores in the previous experiments are span-level proportional scores calculated by the proportion of the overlap referring to the expert annotation(Zhang et al., 2022). To provide additional comparisons between different methods, we also invoke token-level and span-level exact P, R, F_1 scores as supporting metrics. We run the whole process from data augmentation to worker selection with all 3 metrics separately. The kappa threshold τ in **Exp.+MV** is set to 0.4 on the Chinese OEI dataset and 0.65 on the CoNLL 2003 dataset. Detailed scores are listed in Table 2 and 4. The results show that **Exp.+MV** achieves scores as good as **Exp.** and much better than **MV**, which validates previous experiments and shows our worker selection methods are robust to different metrics.

We also test our worker selection methods with a feedback simulator. The simulator generates numerical feedback from $Bernoulli$ distribution in annotation evaluations. This is to eliminate the varying level of difficulty in different tasks and evaluate our worker selection algorithms under more stable settings. Our algorithm achieves good results on the simulator as well. Due to space limitations, we put the definitions and results in Appendix A.

5 Conclusion

This paper focuses on the worker selection problem for span-based sequence labeling tasks. We present the initial work of applying CMAB-based methods to address the problem. Due to label interdependencies, the binary feedback signal in conventional CMAB is not applicable. We propose to use span-level F_1 with **Exp.+MV** as feedback. The real datasets are unbalanced and insufficient for offline simulation of worker selection. To address this, we develop a data augmentation method for span-based sequence labeling datasets that reflects the possible errors in annotating practice. The F_1 scores of generated annotations are nearly the same as workers' actual ones. With the augmented datasets, we conduct extensive experiments. On the Chinese OEI dataset, our method achieves up to 99.47% F_1 score with 47.19% reduction in the expert cost. On the CoNLL 2003 dataset, our method achieves up to 100.04% F_1 score with 65.97% reduction in the expert cost. Both are compared with expert-evaluation-only baselines. Our method achieves up to 94.86% F_1 score and saves 65.97% expert cost on the data-free simulator as well.

Limitations

In this paper, we provide theoretical analysis and offline simulation results of our worker selection algorithm. These results show that our algorithm performs well. But due to the budget limitation, we are unable to apply our algorithm on real online crowdsourcing systems and test it with real-time annotation tasks.

References

Arpita Biswas, Shweta Jain, Debmalya Mandal, and Y. Narahari. 2015. A truthful budget feasible multi-armed bandit mechanism for crowdsourcing time critical tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, page 1101–1109, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 151–159, Atlanta, Georgia, USA. PMLR.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(76):2493–2537.

Leon Derczynski. 2016. Complementarity, F-score, and NLP evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 261–266, Portorož, Slovenia. European Language Resources Association (ELRA).

Hakan Erdogan. 2010. Sequence labeling: Generative and discriminative approaches. In *Proceedings of the 9th International Conference on Machine Learning and Applications*, pages 1–132.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Evrard Garcelon, Vashist Avadhanula, Alessandro Lazaric, and Matteo Pirotta. 2022. Top k ranking for multi-armed bandit with noisy evaluations. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 6242–6269. PMLR.

Jeff Howe. 2006. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4.

Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6(1):4–22.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397.

Feng Li, Jichao Zhao, Dongxiao Yu, Xiuzhen Cheng, and Weifeng Lv. 2022. Harnessing context for budget-limited crowdsensing with massive uncertain workers. *IEEE/ACM Transactions on Networking*, 30(5):2231–2245.

Rajeev Motwani and Prabhakar Raghavan. 1995. *Randomized Algorithms*. Cambridge University Press.

Nikita Nangia, Saku Sugawara, Harsh Trivedi, Alex Warstadt, Clara Vania, and Samuel R. Bowman. 2021. What ingredients make for an effective crowdsourcing protocol for difficult NLU data collection tasks? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1221–1235, Online. Association for Computational Linguistics.

An Thanh Nguyen, Byron Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. 2017. Aggregating and predicting sequence labels from crowd annotations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 299–309, Vancouver, Canada. Association for Computational Linguistics.

Stefanie Nowak and Stefan Rüger. 2010. How reliable are annotations via crowdsourcing: A study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the International Conference on Multimedia Information Retrieval*, MIR '10, page 557–566, New York, NY, USA. Association for Computing Machinery.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Anshuka Rangi and Massimo Franceschetti. 2018. Multi-armed bandit algorithms for crowdsourcing systems with online estimation of workers' ability. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 1345–1352. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM.

Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Sequence labeling with multiple annotators. *Machine learning*, 95(2):165–181.

Filipe Rodrigues and Francisco C. Pereira. 2018. Deep learning from crowds. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Edwin Simpson and Iryna Gurevych. 2019. A Bayesian approach for sequence tagging with crowds. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1093–1104, Hong Kong, China. Association for Computational Linguistics.

Yiwen Song and Haiming Jin. 2021. Minimizing entropy for crowdsourcing with combinatorial multi-armed bandit. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings. 2014. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, 214:89–111.

Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, page 155–164, New York, NY, USA. Association for Computing Machinery.

Xin Zhang, Guangwei Xu, Yueheng Sun, Meishan Zhang, Xiaobin Wang, and Min Zhang. 2022. Identifying Chinese opinion expressions with extremely-noisy crowdsourcing annotations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2801–2813, Dublin, Ireland. Association for Computational Linguistics.

Deniz İren and Semih Bilgen. 2014. Cost of quality in crowdsourcing. *Human Computation*, 1.

A Feedback Simulator

The performance of crowd workers may fluctuate on different kinds of annotation tasks. To validate the **Exp.+MV** worker selection method under more stable settings, we exclude the datasets in the worker selection process and directly generate the numerical feedback when workers give annotations. Specifically, for each worker w, we precalculate the average F_1 score of all annotations by w on the original dataset using expert and MV evaluation respectively, denoted as $\bar{\varphi}_w^{Exp.}$ and $\bar{\varphi}_w^{MV}$. At time step t, for each $s_i \in S_t$, we assign K tasks to K different workers in W_t, then use a random value on $[0, 1]$ as the workers' agreement κ. If $\kappa > \tau$, we generate feedback for the K workers from $Bernoulli(\bar{\varphi}_w^{MV})$ independently. Otherwise, the feedback is generated from $Bernoulli(\bar{\varphi}_w^{Exp.})$. We set the kappa threshold value τ to 0.4 in **Exp.+MV**. The results of this experiment are shown in Table 3. **Exp.+MV** saves 59.88% of expert usage under these settings.

Method	F_1
Oracle	74.12
Random	65.12
Exp.	**69.78**
MV	66.80
Exp.+MV	**68.29**

Table 3: The overall span-level proportional F_1 scores of all methods with the feedback simulator.

B Regret Analysis

We provide a brief regret analysis of the worker selection framework assuming that we use the ϵ-greedy algorithm and that each worker's reward follows a Bernoulli distribution.

The main proof follows the proof of Theorem 1 in (Garcelon et al., 2022). The key contribution here is that we need to specify that the evaluation signal (generated by majority voting) is a generalized linear model of workers' true reward signal (generated by expert/oracle). To this end, we utilize the following form of the Chernoff bound which applies for any random variables with bounded support.

Lemma 1 *(Chernoff Bound (Motwani and Raghavan, 1995)) Let $X_1, X_2, \cdots, X_N$ be independent random variables such that $x_l \le X_i \le x_h$ for all $i \in \{1, 2, \cdots, N\}$. Let $X = \sum_{i=1}^N X_i$ and $\mu = \mathbb{E}(X)$. Given any $\delta > 0$, we have the following result:*

$$P(X \le (1-\delta)\mu) \le e^{-\frac{\delta^2\mu^2}{N(x_h - x_l)^2}}. \qquad (9)$$

For the purpose of our discussion, let $X_i \in \{0, 1\}$ be a binary random variable, where $X_i = 0$ denotes that worker i provides an incorrect solution, and $X_i = 1$ denotes that worker i generates a correct solution. Define $X = \sum_{i \in \mathcal{N}} X_i$.

We aim to approximate P_{MV}, which is the probability that the majority of the N workers provide the correct estimate. We apply the Chernoff Bound

Method	Token-level			Span-level Exact			Span-level Prop.		
	P	R	F_1	P	R	F_1	P	R	F_1
Oracle	62.88	68.62	64.80	54.48	51.97	53.07	72.79	64.07	68.15
Random	58.49	57.30	57.42	43.99	35.50	39.18	69.01	52.36	59.55
ϵ-G (Exp.)	61.91	64.58	**62.61**	51.72	46.37	**48.76**	72.28	60.25	**65.72**
ϵ-G (MV)	60.87	63.52	61.55	48.72	44.66	46.37	70.15	58.94	64.05
ϵ-G (Exp.+MV)	61.76	64.46	**62.47**	49.14	45.35	**46.96**	71.21	59.92	**65.08**
CUCB (Exp.)	63.02	63.75	**62.93**	52.24	45.51	**48.56**	73.05	59.53	**65.60**
CUCB (MV)	61.94	62.09	61.55	49.57	44.39	46.66	71.22	57.59	63.68
CUCB (Exp.+MV)	62.83	63.62	**62.75**	51.31	45.60	**48.16**	72.48	59.33	**65.25**

Table 4: Detailed P, R, and F_1 scores of all methods on the Chinese OEI dataset.

in Lemma 1 to P_{MV}. We can compute

$$\mathbb{E}(X) = \bar{p} = \frac{\sum_{i=1}^{N} p_i}{N}. \tag{10}$$

Based on (9), we let $\mu = \mathbb{E}(X)$, $\delta = \frac{N(\bar{p}-\frac{1}{2})}{\frac{N}{2}+N(\bar{p}-\frac{1}{2})}$, $x_l = 0$, $x_h = 1$, and get the following result:

$$\begin{aligned} P_{\mathrm{MV}} &= P\left(X \geq \frac{N}{2}\right) = 1 - P\left(X \leq \frac{N}{2}\right) \\ &\geq 1 - e^{-\frac{\delta^2 \mu^2}{N}} && (11) \\ &= 1 - e^{-\frac{\frac{N^2(\bar{p}-\frac{1}{2})^2}{[\frac{N}{2}+N(\bar{p}-\frac{1}{2})]^2}[\frac{N}{2}+N(\bar{p}-\frac{1}{2})]^2}{N}} && (12) \\ &= 1 - e^{-\frac{N^2(\bar{p}-\frac{1}{2})^2}{N}} && (13) \\ &= 1 - e^{-N\left(\frac{\sum_{i=1}^{N} p_i}{N} - \frac{1}{2}\right)^2}. && (14) \end{aligned}$$

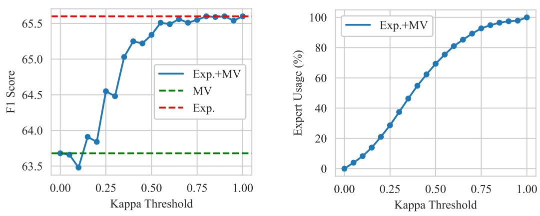
Through approximating P_{MV} by its lower bound in (14), we can see that the evaluation signal (represented by P_{MV}) is an increasing function in each worker's capability p_i and twice-differentiable. That is, P_{MV} is a generalized linear function, which satisfies Assumption 3 in (Garcelon et al., 2022). Therefore, one can follow the proof of Theorem 1 in (Garcelon et al., 2022) that the ϵ-greedy algorithm yields a sub-linear regret with order $\tilde{O}(T^{2/3})$.

Algorithm 2 The annotation selection algorithm.

For each worker $w_j \in W$, maintain (1)a variable $\hat{\varphi}_j$ as the average F_1 score of the selected annotations by w_j so far, (2)a set $\boldsymbol{A}^j$ of selected annotations by w_j
Generate all possible annotations $\boldsymbol{A}_1^p$ on $s_1 \in S$, calculate $F_1^{\mathrm{Exp}}(\boldsymbol{a}_{1k})$ for each $\boldsymbol{a}_{1k} \in \boldsymbol{A}_1^p$
For each $w \in W$, initialize $\hat{\varphi}_j$ with the $F_1^{\mathrm{Exp}}(\boldsymbol{a}_{1k})$ closest to $\bar{\varphi}_j$, and append the $\boldsymbol{a}_{1k}$ to $\boldsymbol{A}^j$
for all $s_i \in S \backslash s_1$ **do**
  Generate all possible annotations $\boldsymbol{A}_i^p$ on $s_i \in S$, calculate $F_1^{\mathrm{Exp}}(\boldsymbol{a}_{ik})$ for each $\boldsymbol{a}_{ik} \in \boldsymbol{A}_i^p$
  for all $w_j \in W$ **do**
    if $\hat{\varphi}_j > \bar{\varphi}_j$ **then**
      Update $\hat{\varphi}_j$ with the maximal $F_1^{\mathrm{Exp}}(\boldsymbol{a}_{ik})$ less than $\bar{\varphi}_j$, and append $\boldsymbol{a}_{ik}$ to $\boldsymbol{A}^j$
    else
      Update $\hat{\varphi}_j$ with the minimal $F_1^{\mathrm{Exp}}(\boldsymbol{a}_{ik})$ greater than $\bar{\varphi}_j$, and append $\boldsymbol{a}_{ik}$ to $\boldsymbol{A}^j$
    end if
  end for
end for

Worker ID	Ori. F_1	Rnd. Gen. F_1	SES Only F_1	SES +Alg.2 F_1	Worker ID	Ori. F_1	Rnd. Gen. F_1	SES Only F_1	SES +Alg.2 F_1
25	62.90	60.07	69.59	62.89	37	37.15	96.10	26.79	37.16
32	60.87	41.37	68.79	60.87	13	36.19	31.62	25.14	36.20
42	53.88	4.37	66.57	53.88	20	36.11	71.44	25.02	36.12
5	52.07	50.74	60.76	52.06	64	35.97	65.66	25.39	35.97
55	50.70	30.24	61.13	50.70	63	35.22	75.40	24.73	35.22
2	50.53	91.99	60.92	50.53	6	35.15	65.74	25.00	35.16
52	50.08	41.93	60.91	50.08	10	34.63	51.28	25.08	34.64
17	49.82	43.73	35.82	49.82	66	33.75	60.98	24.99	33.75
57	49.25	13.17	35.59	49.25	53	32.90	27.51	24.78	32.89
11	49.04	53.71	35.19	49.03	4	32.72	8.40	24.77	32.72
26	48.89	5.17	35.59	48.82	21	32.19	73.47	24.78	32.19
36	48.71	15.53	35.27	48.70	62	32.16	48.71	24.89	32.16
46	48.67	44.84	35.19	48.67	1	32.10	34.42	24.96	32.10
29	48.60	95.39	35.21	48.60	41	31.94	77.55	24.88	31.93
35	47.07	23.64	35.34	47.07	51	31.78	68.07	24.85	31.78
49	46.80	60.30	35.27	46.80	31	31.61	29.44	24.59	31.61
54	45.63	18.74	34.45	45.64	8	31.05	28.55	24.76	31.05
14	45.13	60.99	34.54	45.13	67	30.91	95.51	24.22	30.91
43	44.93	34.91	33.72	44.93	58	30.70	21.64	23.96	30.70
7	44.37	23.89	33.50	44.37	65	30.61	4.51	24.17	30.60
59	44.36	72.37	33.61	44.37	38	30.47	4.82	24.11	30.47
23	43.38	4.85	33.58	43.38	28	29.86	2.63	24.00	29.86
56	43.37	41.96	33.31	43.37	45	29.38	36.13	24.15	29.38
0	41.60	66.81	28.19	41.61	30	28.70	61.16	21.88	28.71
18	41.40	31.53	28.56	41.40	15	25.73	38.92	21.40	25.73
16	41.31	57.13	28.03	41.31	19	24.69	4.39	21.31	24.70
22	41.05	85.83	28.21	41.06	44	23.42	7.15	21.08	23.42
47	40.78	82.33	27.91	40.78	9	22.88	96.22	21.22	22.89
61	40.22	12.20	28.44	40.22	33	22.36	29.89	19.50	22.36
40	40.01	84.98	28.38	40.02	39	20.69	57.73	19.26	20.69
50	39.35	56.04	28.64	39.35	69	20.39	63.02	19.26	20.40
27	38.77	34.07	27.87	38.77	3	17.12	28.70	18.66	17.13
48	38.35	23.77	27.57	38.35	24	16.96	42.73	18.68	16.98
34	38.29	5.69	28.08	38.30	68	14.53	13.63	7.69	14.53
12	37.96	85.14	27.44	37.96	60	13.66	22.69	8.15	13.66

Table 5: Comparisons between different data augmentation methods on the span-level exact F_1 score of every crowd worker. **Ori.** stands for the original score in real datasets before any augmentation. **Rnd. Gen.** is a naive augmentation method with random generated annotations. **SES Only** indicates the *shifting*, *shrinking*, and *expanding* method we proposed. **SES + Alg.2** means SES with Algorithm 2 which is our final method.



(a) F_1 score w.r.t τ (b) Expert usage w.r.t τ

Figure 5: F_1 scores of the produced annotations and usage of expert for annotation evaluations w.r.t the kappa threshold τ of the **Exp.+MV** method on the Chinese OEI dataset.