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

Anonymous ACL submission

Abstract

This paper introduces a novel crowdsourcing worker selection algorithm, enhancing annotation quality and reducing costs. Unlike previous studies targeting simpler tasks, this study contends with the complexities of label interdependencies in sequence labeling. The proposed algorithm utilizes a Combinatorial Multi-Armed Bandit (CMAB) approach for worker selection, and a cost-effective human feedback mechanism. The challenge of dealing with imbalanced and small-scale datasets, which hinders offline simulation of worker selection, is tackled using an innovative data augmentation method termed *shifting*, *expanding*, and *shrinking* (SES). Rigorous testing on CoNLL 2003 NER and Chinese OEI datasets showcased the algorithm's efficiency, with an increase in F_1 score up to 100.04% of the expert-only baseline, alongside cost savings up to 65.97%. The paper also encompasses a dataset-independent test emulating annotation evaluation through a Bernoulli distribution, which still led to an impressive 97.56% F_1 score of the expert baseline and 59.88% cost savings. Furthermore, our approach can be seamlessly integrated into Reinforcement Learning from Human Feedback (RLHF) systems, offering a cost-effective solution for obtaining human feedback. This research not only addresses the challenges in worker selection for intricate NLP tasks but also paves the way for more economical and efficient human-in-the-loop systems.

1 Introduction

Crowdsourcing, the practice of obtaining labeled data from a multitude of contributors (Howe, 2006), has emerged as a pivotal tool in data collection for deep learning models. It offers a cost-effective alternative to expert labeling, making it especially valuable in today's data-driven research landscape (Nowak and Rüger, 2010). While its application spans various domains, from image labeling to text classification (Venanzi et al., 2014), this paper narrows its focus on span-based sequence labeling tasks, which assign categorical labels to individual words within a sentence (Erdogan, 2010). Notable examples of such tasks include named entity recognition (NER) and opinion expression identification (OEI) (Collobert et al., 2011).

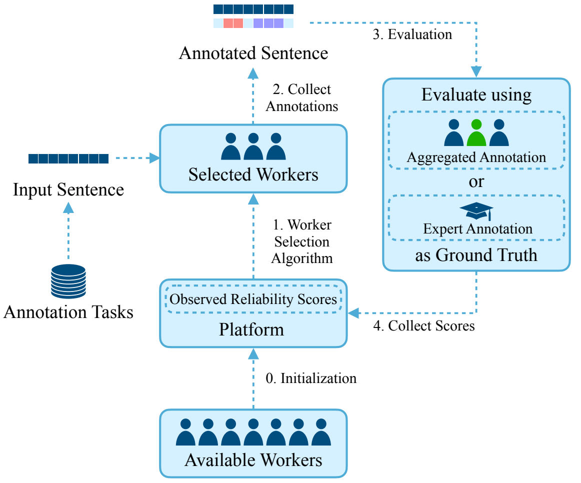


Figure 1: Our online worker selection framework for crowdsourcing.

The inherent complexity of sequence labeling lies in the interdependencies of labels within a sequence. Unlike simpler tasks where labels are independent, sequence labeling requires contextual understanding, making it inherently more challenging (Rodrigues et al., 2014). Consequently, annotations from crowd workers, who might not possess the expertise of trained annotators, often exhibit reduced accuracy. This underscores the imperative to enhance annotation quality, a challenge that this study addresses.

A significant motivation driving this research is the potential application of a mixed feedback mechanism in Reinforcement Learning from Human Feedback (RLHF) systems. RLHF systems traditionally rely heavily on expert feedback, which, while accurate, is expensive and often not scalable (Casper et al., 2023). By integrating feedback from both experts and aggregated crowd workers, we can achieve a balance between accuracy and cost. This hybrid approach not only maintains the quality of feedback but also significantly reduces the financial burden, making RLHF systems more accessible and scalable.

Historically, research in this domain has concentrated on annotation aggregation (Rodrigues et al., 2014; Nguyen et al., 2017; Simpson and Gurevych, 2019), employing methods post data collection. However, given the varied skill levels among crowd workers, a proactive approach that identifies and leverages the most accurate workers during the data collection phase can significantly enhance data quality. Termed as online worker selection, this strategy involves iterative allocation of a set budget across a pool of workers to optimize annotation quality (Chen et al., 2013). This dynamic process grapples with the uncertainty of worker skill levels, necessitating a balance between exploring new workers and exploiting currently identified proficient ones.

In the context of sequence labeling, traditional bandit-based algorithms (Rangi and Franceschetti, 2018) fall short due to the intricacies introduced by label dependencies. These intricacies manifest challenges in both annotation evaluation and aggregation. To address the evaluation challenge, this study employs the span-level F_1 score (Derczynski, 2016), a widely recognized metric, as the feedback signal in the worker selection process. The core **challenge** here is the accurate computation of the F_1 score in the absence of expert annotations as a reference. The objective is to minimize reliance on costly expert annotations. For aggregation, while the majority voting method is employed for its simplicity and effectiveness, its reliability can be compromised when faced with divergent annotations from different annotators.

The overarching **goal** of this research is to maximize the quality of annotations while minimizing costs. This involves strategically replacing expert ground truth labels with aggregated crowd-sourced labels, ensuring that the overall F_1 score remains high. Such replacements are made only when there's a high level of agreement among crowd workers, indicating that expert evaluation might be redundant for that particular sequence. The proposed worker selection algorithm, as illustrated in Figure 1, adopts an iterative approach: tasks are assigned to a subset of workers, their annotations are evaluated, and the resulting scores inform worker selection in subsequent rounds.

However, real-world datasets present challenges due to their imbalanced nature and limited scale (Rodrigues et al., 2014; Zhang et al., 2022). Addressing these challenges, this paper introduces a data augmentation method tailored for span-based sequence labeling datasets. This method, designed to emulate potential human annotation errors, ensures that aggregated annotations remain meaningful. Three specific modifications, namely shifting, expanding, and shrinking, are applied to expert annotations, generating a spectrum of potential human annotations. This augmentation addresses dataset limitations, facilitating the offline evaluation of worker selection algorithms.

In summary, this paper's contributions are manifold[1]:

- It presents the exploration of worker selection for span-based sequence labeling tasks, recognizing the unique challenges they present.
- It employs the span-level F_1 score, evaluated by both experts and crowd workers, as a feedback mechanism, ensuring accurate worker selection.
- It introduces a data augmentation technique to counteract the limitations of real datasets, enabling effective offline simulations.
- Through rigorous experimentation, it demonstrates the efficacy of the proposed method, achieving impressive F_1 scores while significantly reducing expert annotation costs.

To provide a comprehensive understanding of the background of this work, we briefly introduce the related work in Appendix A.

2 Methodology

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 $\boldsymbol{a}_{ij} = a_1 a_2 \dots a_k \dots a_l$ 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.

---

[1] All resources, including source code and datasets, are made available for the broader research community at `https://anonymous.4open.science/r/CostEffi/`

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 2.2. 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 different workers in W_t, and collect their annotations $\boldsymbol{A}_i = \{\boldsymbol{a}_{i1}, \boldsymbol{a}_{i2}, \dots, \boldsymbol{a}_{iK}\}, \forall i \in \{1, 2, \dots, |S_t|\}$. 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.

2.1 Problem Formulation

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

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

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

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.

2.2 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:

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

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.,

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

Expert+MV When workers give very different annotations on the same sentence (usually when the task is difficult), 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 $\mathrm{F}_1^{\text{MV}}(\boldsymbol{a}_{ij}(t))$. Otherwise, the reward is $\mathrm{F}_1^{\text{Exp}}(\boldsymbol{a}_{ij}(t))$. In this way, MV is only used when the crowd workers can reach an agreement. Thus the reward is always calculated based on reliable ground truth. We summarize the reward of worker w_j as:

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

The ϵ-Greedy, Thompson Sampling, and Combinatorial Upper Confidence Bound (CUCB) are three 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:

Measure	Chinese OEI	CoNLL 2003
# of Sentence	8047	4580
# of Worker	70	47
Span Length	5.05	1.51
Max	658	1626
Min	153	48
Range	505	1578
Mean	368	350
Median	332.5	230
SD	135.23	328.01
Variance	18286.52	107589.34
CV	36.71%	93.57%

Table 1: Statistics of the original datasets. Span lengths are averages. The terms SD and CV represent Standard Deviation and Coefficient of Variation respectively. The metrics Max, Min, Range, Mean, Median, SD, Variance, and CV pertain to the number of sentences annotated by each worker, indicating dataset imbalances.

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

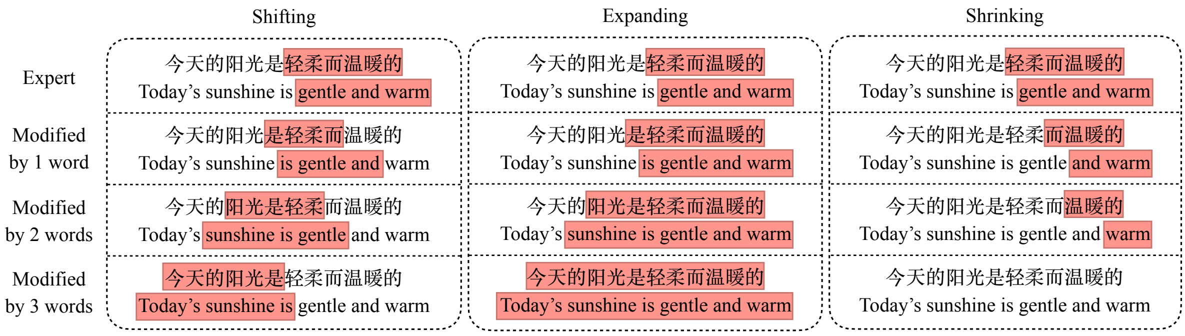
Thompson Sampling samples from gaussian distributions of workers' rewards at each time step t, and select workers which could maximize the total reward. CUCB handles the tradeoff by adding an item considering T_j and t to $\bar{\mu}_j$ like:

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

This makes workers with less annotations more likely to be selected as the algorithm proceeds. We provide a brief analysis in Appendix C. We explain on the application of our worker selection algorithms when building new datasets in Appendix F

2.3 Data Augmentation Method

We propose the data augmentation method to facilitate the offline simulation of the crowdsourcing process, thus evaluating the worker selection algorithms. During offline simulation, when the worker selection strategy selects a certain worker to annotate a certain sentence, we can use the annotation in the original dataset if it exists. But if the selected worker did not annotate the sentence in the original dataset, we need to generate an annotation for the



	Shifting	Expanding	Shrinking
Expert	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm
Modified by 1 word	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm
Modified by 2 words	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm
Modified by 3 words	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm	今天的阳光是轻柔而温暖的 Today’s sunshine is gentle and warm

Figure 2: An example of the three methods to generate annotations. Chinese characters and corresponding English words with red backgrounds indicate annotation spans. Transliteration of the Chinese sentence is *jīn tiān de yáng guāng shì qīng róu ér wēn nuǎn de*.

sentence. And the generated annotation should be in the same quality (depicted in F-score) as the real annotations by the worker. The generated annotation will be then used with the other annotations on the same sentence for majority voting.

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 lack previous work on generating missing crowd annotations for span-based sequence labeling, 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.

Through our statistical analysis and observation on the real datasets, we characterized the 3 most common annotation error patterns. Due to space limitation, we defer the detailed analysis to Appendix D. Based on these analysis results, we propose a data augmentation method as follows: 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.

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.

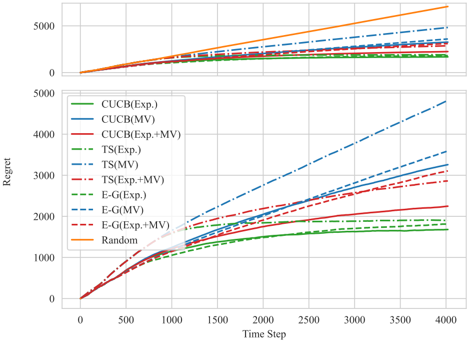
For each sentence $s_i \in S$, s_i may contain multiple annotation spans. We perform modifications on each span in s_i, and find all combinations of spans to form possible sentence annotations. With these methods, we can imitate crowd annotations with different kinds of errors in practice. Next, for 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.

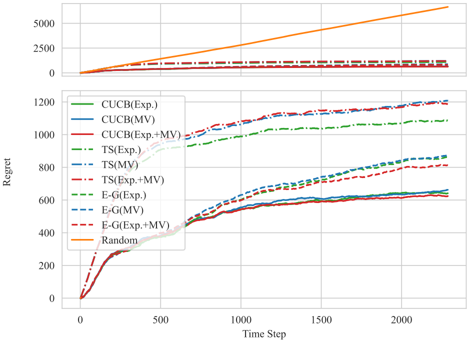
To better illustrate the procedure of the augmentation, we provide a running example in Appendix E.

Worker ID	**Rnd. Gen.** $\lvert\Delta\mathbf{F}_1\rvert$	**SES Only** $\lvert\Delta\mathbf{F}_1\rvert$	**SES +Alg.2** $\lvert\Delta\mathbf{F}_1\rvert$
25	2.83	6.69	0.01
52	8.15	10.83	0.00
46	3.83	13.48	0.00
43	10.02	11.21	0.00
18	9.87	12.84	0.00
50	16.69	10.71	0.00
12	47.18	10.52	0.00
Avg.	14.08	10.90	0.0014

Table 2: Comparisons between different data augmentation methods on the error of span-level exact F_1 score of every crowd worker. The error $|\Delta F_1|$ is calculated as the absolute difference between each worker's F_1 score after augmentation and his real F_1 score. **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 optimal method.



(a) Chinese OEI



(b) CoNLL 2003

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

3 Experiments

3.1 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[2], 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.

3.2 Data Augmentation

We augment both datasets with the method proposed in Section 2.3. According to Table 1, the most hard-working annotator in the OEI dataset provided annotations on 658 sentences, while the least one annotated only 153 sentences. On average, each crowd worker annotated 368 out of 8047 sentences in the Chinese OEI dataset. For the offline simulation of the worker selection process, we want every worker to annotate all 8047 sentences. Therefore we need to generate the missing 8047 - 368 = 7679 annotations for every worker, on average. This also applies similarly to the CoNLL 2003 dataset.

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[3]. Due to space limitation, we present comparisons of different augmentation algorithms with 10 sampled workers in Table 2. The complete results are deferred to Table 7 in the appendix. These results show that our **SES + Alg.2** method clearly outperforms the other baselines, producing almost the same F_1 scores for each worker as their original ones.

---

[2] `https://english.sina.com/weibo/`

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

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**
TS (Exp.)	64.94	79.88	**68.51**	75.64	68.31	**71.57**	85.02	75.71	**80.09**
TS (MV)	64.47	79.19	67.91	74.97	67.54	70.80	84.14	74.21	78.86
TS (Exp.+MV)	64.20	79.09	**67.62**	75.27	67.83	**71.12**	84.77	75.39	**79.81**
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 3: Detailed P, R, and F_1 scores of all methods on the CoNLL 2003 dataset. All our algorithms perform significantly better than the Random (i.e., naive crowdsourcing) baseline.

3.3 Worker Selection

Baselines We test the **Exp.+MV** method with 4 baselines: **Oracle**, **Random**, **Exp.**, and **MV**. **Oracle** always 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, which is equivalent to usual crowdsourcing procedure without worker selection. **Exp.**, **MV**, and **Exp.+MV** are CMAB-based algorithms introduced in Section 2.2. The CMAB-based algorithms are tested with CUCB, Thompson Sampling and ϵ-Greedy as the worker selection criterion respectively.

Regret as a Metric We evaluate our worker selection algorithms using cumulative regret, a metric indicating the performance deviation from the oracle's selection defined as:

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

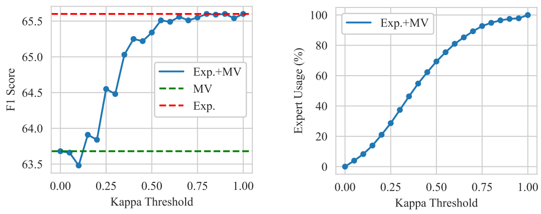
In our experiments, we request 10 annotations per sentence, allowing CMAB-based algorithms to converge, and select 20 workers at each time step t. On the Chinese OEI dataset, setting the kappa threshold τ to 0.4 in **Exp.+MV** results in a 57.02% reduction in expert annotation cost, while a 0.65 threshold on the CoNLL 2003 dataset leads to a 43.83% cost reduction.

Results show **Random** consistently underperforms across datasets. On the Chinese OEI dataset, **Exp.+MV** surpasses **MV**, albeit with higher regret than **Exp.**, justified by the substantial cost savings. On the CoNLL 2003 dataset, **Exp.+MV** even outperforms **Exp.**, suggesting crowd workers can provide valuable input for simpler tasks like NER. Overall, algorithms employing the CUCB criterion demonstrate superior performance, with **CUCB (Exp.+MV)** excelling in balancing cumulative regret and expert cost.

Effect of τ on F_1 and cost 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 Thompson Sampling and ϵ-Greedy on both datasets, we display only the results from CUCB in later experiments.

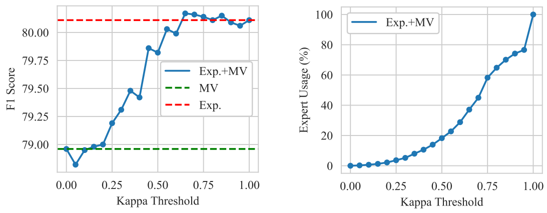
On the Chinese OEI dataset, as illustrated in Figure 4, F_1 increases sharply with $\tau \in [0, 0.4]$. When $\tau = 0.4$, **Exp.+MV** achieves 99.47% F_1 score of **Exp.**, and saves 47.19% of the expert cost. The F_1 score goes up slowly until τ reaches 0.8. When $\tau = 0.8$, the F_1 score of **Exp.+MV** 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 5. Similarly, the F_1 score of



(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 Chinese OEI dataset.



(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 CoNLL 2003 dataset.

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.

Our **CUCB (Exp.+MV)** worker selection algorithm eliminates the need for expert evaluation on every sentence. Instead, we harness crowd intelligence via our kappa-thresholded MV, producing datasets of comparable or even superior quality to those relying solely on expert evaluations.

Extended F_1 Metrics 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 3 and 6. 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.

Feedback Simulator 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. We put the definitions and results in Appendix B.

Effect on ML Models To further show the effect of our worker selection algorithm on the performance of machine learning models, we have run experiments with several widely-accepted models and provide the results in Table 5. We observe a consistent increment of F1 score on the ML models, with our bandit-based worker selection algorithm. This validates that our worker selection algorithm may help improve the performance of ML models while saving budget on data crowdsourcing.

4 Conclusion

In this study, we introduced a CMAB-based worker selection strategy tailored for span-based sequence labeling tasks, leveraging the span-level F_1 with **Exp.+MV** as a feedback mechanism. To address the challenges posed by unbalanced and limited real datasets, we innovated a data augmentation method. This technique not only facilitates offline simulation but also mirrors the genuine annotation behaviors of workers closely.

Our empirical evaluations underscore the efficacy of the proposed method. On the Chinese OEI dataset, our approach achieved an impressive 99.47% F_1 score, translating to a substantial 47.19% reduction in expert costs. Similarly, on the CoNLL 2003 dataset, we observed a remarkable 100.04% F_1 score, with savings of up to 65.97% in expert costs, both benchmarks set against expert-evaluation-only baselines. Furthermore, our method demonstrated its robustness with a 94.86% F_1 score and a 65.97% reduction in expert costs on a data-free simulator. Our approach also boosts ML model performance, optimizing both accuracy and cost.

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. Future work could explore extending our worker selection strategy to other domains and tasks and in online environment, further solidifying its role as a versatile and cost-effective tool in the realm of crowdsourcing and beyond.

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*, pages 1101–1109.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

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*, pages 151–159.

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*, pages 261–266.

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

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*, pages 6242–6269.

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. *Advances in Applied Mathmatics*, 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. *Journal of Machine Learning Research*, 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*.

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*, pages 1221–1235.

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*, pages 299–309.

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*, page 557–566.

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*, pages 1345–1352.

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*, pages 1611–1618.

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*, pages 1093–1104.

Yiwen Song and Haiming Jin. 2021. Minimizing entropy for crowdsourcing with combinatorial multi-armed bandit. In *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*, 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*, page 155–164.

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*, pages 2801–2813.

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

A 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.

B Feedback Simulator

The performance of crowd workers can vary across different types of annotation tasks. To evaluate the **Exp.+MV** worker selection method in more stable conditions without task-specific influence, we do not actually annotate the sentences, but directly use a worker's average F_1 score to simulate his score on each sentence he annotates. The simulated scores are used as the numerical feedback for worker selection. Specifically, for each worker w, we calculate in advance two average F_1 scores for all of their annotations on the original dataset. The two F_1 scores for each worker are calculated using expert and majority vote (MV) evaluation respectively, denoted as $\bar{\varphi}_w^{Exp.}$ and $\bar{\varphi}_w^{MV}$. At each time step t, for every sentence s_i in the sentence set to be annotated S_t, we ask K different workers from the current selected workers W_t to annotate it. Then, we use a random value between 0 and 1 as the agreement level κ. If κ exceeds the threshold value τ (set to 0.4 in **Exp.+MV**), we independently generate feedback for the K workers from a Bernoulli distribution with a probability parameter set to $\bar{\varphi}_w^{MV}$. If

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

Table 4: The overall span-level proportional F_1 scores of all methods with the feedback simulator.

Method	Original	w/ Our Alg.
LSTM-Crowd-cat	52.66	54.27
Bert-BiLSTM-CRF	52.14	54.51
Annotator-Adaptor	53.86	56.16

Table 5: Span-level exact F_1 scores of widely-accepted deep learning models on the Chinese OEI dataset. LSTM-Crowd-cat is from Nguyen et al. (2017). Bert-BiLSTM-CRF and Annotator-Adaptor are from Zhang et al. (2022). We provide results with and without our worker selection algorithm.

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 $\{W_{t1}, \dots, W_{ti}, \dots, W_{tn}\}$, 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 $F_1^{MV}(\boldsymbol{a}_{ij}(t))$
        else
            Update $T_i$ and $\bar{\mu}_j$ with $F_1^{Exp}(\boldsymbol{a}_{ij}(t))$
        end if
    end for
end while
```

not, the feedback is generated from a Bernoulli distribution with a probability parameter set to $\bar{\varphi}_w^{Exp.}$. The span-level average F_1 scores of the annotated dataset using different worker selection algorithm are shown in Table 4. Our feedback mechanism **Exp.+MV** for worker selection achieved comparable performance to the expert-only mechanism **Exp.** (68.29 versus 69.78), while in the same time reduced expert involvement in evaluation by 59.88% under the dataset-independent conditions.

C 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:*

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

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 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 $\mu = \mathbb{E}(X),\ \delta = \frac{N(\bar{p} - \frac{1}{2})}{\frac{N}{2} + N(\bar{p} - \frac{1}{2})}$,

Algorithm 2 The annotation selection algorithm.

```
For each worker $w_j \in W$, maintain (1)a variable $\hat{\varphi}_j$ as the average $\mathrm{F}_1$ 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 $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{1k})$ for each $\boldsymbol{a}_{1k} \in \boldsymbol{A}_1^p$
For each $w \in W$, initialize $\hat{\varphi}_j$ with the $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{1k})$ 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 $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{ik})$ 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 $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{ik})$ less than $\bar{\varphi}_j$, and append $\boldsymbol{a}_{ik}$ to $\boldsymbol{A}^j$
        else
            Update $\hat{\varphi}_j$ with the minimal $\mathrm{F}_1^{\mathrm{Exp}}(\boldsymbol{a}_{ik})$ greater than $\bar{\varphi}_j$, and append $\boldsymbol{a}_{ik}$ to $\boldsymbol{A}^j$
        end if
    end for
end for
```

$x_l = 0, x_h = 1$, and get the following result:

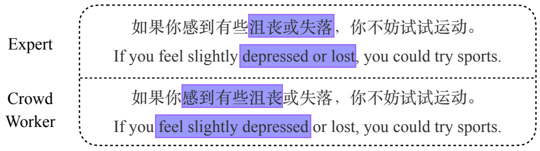
$$
\begin{aligned}
P_{\mathrm{MV}} &= P\left(X \geq \frac{N}{2}\right) = 1 - P\left(X \leq \frac{N}{2}\right) \\
&\geq 1 - e^{-\frac{\delta^2 \mu^2}{N}} && (11) \\
&= 1 - e^{-\frac{\frac{N^2(\bar{p}-\frac{1}{2})^2}{[\frac{N}{2}+N(\bar{p}-\frac{1}{2})]^2}[\frac{N}{2}+N(\bar{p}-\frac{1}{2})]^2}{N}} &&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

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 $\tilde{O}(T^{2/3})$.

D Case Study of Annotation Errors

Based on our statistical analysis of the Chinese OEI dataset, we find that 74.80% of annotations have different types of errors. And these annotation errors could be decomposed to three basic error types, namely Shifting, Expanding, and Shrinking (SES). In our data augmentation algorithm, we reversely used SES modifications and their combinations on the ground truth annotations to generate annotations with varying errors made by crowd workers. In this section, we provide a detailed characterization of human-made errors observed on annotated data with real cases to better motivate these modifications.

Shifting Some crowd annotation spans are as long as expert ones, but their positions are wrong. *Shifting* simulates this type of error. As depicted in Figure 6, both the expert span and the crowd span are three words long and of negative polarity. The difference is that the crowd span is shifted to the left by 2 words compared with the expert span. This type of error can be generated with *Shifting* on the expert annotations.



如果你感到有些沮丧或失落，你不妨试试运动。

If you feel slightly depressed or lost, you could try sports.

如果你感到有些沮丧或失落，你不妨试试运动。

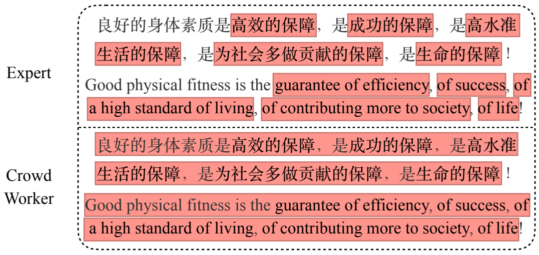
If you feel slightly depressed or lost, you could try sports.

Figure 6: A case in which the crowd worker annotates a span with correct length and polarity but incorrect position.

Expanding *Expanding* is used to generate longer (than expert span) error spans. It might be intuitive that annotators barely make errors such as expanding to a very long span. However, in the case illustrated in Figure 7, the expert annotates five short spans separated by commas, while the crowd worker uses a very long span that covers the whole sentence, which is obviously not accurate. To simulate such human-made errors, we can expand an expert span to cover unnecessary words. Statistically, 4.03% of annotation errors are very long spans with more than 15 Chinese characters. So we do not set an upper bound of span length in *Expanding*.

Shrinking *Shrinking* is useful since crowd workers often ignore some words when annotating. As shown in Figure 8, the crowd worker failed to find all words expressing positive opinions.

Sometimes crowd workers ignore a whole span in expert annotations. This is why we set the lower


Expert
Crowd Worker

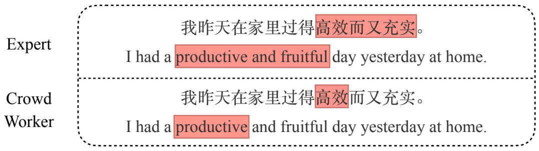
良好的身体素质是高效的保障，是成功的保障，是高水准生活的保障，是为社会多做贡献的保障，是生命的保障！

Good physical fitness is the guarantee of efficiency, of success, of a high standard of living, of contributing more to society, of life!

良好的身体素质是高效的保障，是成功的保障，是高水准生活的保障，是为社会多做贡献的保障，是生命的保障！

Good physical fitness is the guarantee of efficiency, of success, of a high standard of living, of contributing more to society, of life!

Figure 7: A case in which the crowd worker uses a very long span that covers the whole sentence.


Expert
Crowd Worker

我昨天在家里过得高效而又充实。

I had a productive and fruitful day yesterday at home.

我昨天在家里过得高效而又充实。

I had a productive and fruitful day yesterday at home.

Figure 8: A case in which the crowd worker does not annotate all words with polarity.

bound of span length to zero in *Shrinking*, which means we can shrink a span into no span.

These three types of errors may occur separately or combined in real crowd annotations. Such that an error could be both shifted and shrunk. This is why we use the combination of these three types of modifications to simulate human-made errors in our data augmentation algorithm.

E A Running Example of Data Augmentation

We here provide a running example to illustrate how an annotation for a certain worker on a certain sentence is generated with our proposed augmentation method. Suppose we have an English sentence:

> Although he looked very depressed yesterday, he has already become much more cheerful now.

And an expert annotation:

> Although he looked **[NEGATIVE: very depressed]** yesterday, he has already become **[POSITIVE: much more cheerful]** now.

If the crowd worker Sam has an annotation on this sentence in the original dataset, we use it directly in the augmented dataset. Otherwise, we generate an annotation for Sam with our data augmentation method.

When generating annotation for Sam, we follow the steps below:

1. For each span in the expert annotation, we apply the Shifting, Expanding, and Shrinking (SES) modifications on it. After this step, we have several lists of annotation, each list contain annotations with only one modified span:
    - List 1, modifications of the first span, containing N_1 annotations:
        - Although he looked **[NEGATIVE: very depressed]** yesterday, he has already become much more cheerful now. *# Unmodified, span-level proportional F1 = 1.0*
        - Although he looked very **[NEGATIVE: depressed yesterday]**, he has already become much more cheerful now. *# Shifting, F1 = 0.5*
        - Although he looked very depressed **[NEGATIVE: yesterday ,]** he has already become much more cheerful now. *# Shifting, F1 = 0*

        ... *# Other Shifting modifications*
        - Although he **[NEGATIVE: looked very depressed]** yesterday, he has already become much more cheerful now. *# Expanding, F1 = 1.0*

        ... *# Other Expanding modifications*
        - Although he looked very **[NEGATIVE: depressed]** yesterday, he has already become much more cheerful now. *# Shrinking, F1 = 0.5*

        ... *# Other Shrinking modifications*
    - List 2, modifications of the second span, containing N_2 annotations:
        - Although he looked very depressed yesterday, he has already become **[POSITIVE: much more cheerful]** now. *# Unmodified, span-level proportional F1 = 1.0*
        - Although he looked very depressed yesterday, he has already become much **[POSITIVE: more cheerful now]**. *# Shifting, F1 = 0.6667*
        - Although he looked very depressed yesterday, he has already become much more **[POSITIVE: cheerful now .]** *# Shifting, F1 = 0.3334*

        ... *# Other Shifting modifications*
        - Although he looked very depressed yesterday, he has already **[POSI-**

TIVE: become much more cheerful] now. *# Expanding, F1 = 1.0*

... *# Other Expanding modifications*

- Although he looked very depressed yesterday, he has already become much **[POSITIVE: more cheerful]** now. *# Shrinking, F1 = 0.6667*

... *# Other Shrinking modifications*

2. We choose one annotation from each list, and combine them to generate an annotation with 2 spans. This is done for all combinations of the annotations in the two lists. Note that if the two spans overlay with each other, we merge them into one span. After step 2, we have one list of annotations:

   Combined List, containing less than or equal to $N_1 \times N_2$ annotations:

   - Although he looked **[NEGATIVE: very depressed]** yesterday, he has already become **[POSITIVE: much more cheerful]** now. *# span-level proportional F1 = 1.0*
   - Although he looked very **[NEGATIVE: depressed yesterday]** , he has already become **[POSITIVE: much more cheerful]** now. *# span-level proportional F1 = 0.75*
   - Although he looked very depressed **[NEGATIVE: yesterday ,]** he has already become **[POSITIVE: much more cheerful]** now. *# span-level proportional F1 = 0.5*

   ... *# Other combinations with F1 ranging from 0 to 1.0*

3. We choose one annotation from the combined list as Sam's annotation on this sentence, according to the following procedure:
   (a) Sam has an average F1 score $F_{\text{ori}} = 0.57$ on the original (real) dataset.
   (b) We have already got 10 annotations for Sam in the augmented dataset, which has an average F1 score $F_{\text{aug\_10}} = 0.54$.
   (c) We are choosing annotation on the 11th sentence for Sam.
   (d) We firstly select two annotations with the closest F1 scores to F_{ori} from the combined list, one higher than F_{ori}, and one lower than F_{ori}, as candidate annotations. In this case, the two annotations could have F1 scores of 0.58 and 0.52 respectively.
   (e) If $F_{\text{aug\_10}} > F_{\text{ori}}$, we choose the annotation with the lower F1 score (0.52) as Sam's annotation on this sentence. Otherwise, we choose the annotation with the higher F1 score (0.58). This is to ensure that the average F1 score of Sam's annotations in the whole augmented dataset, F_{aug}, is as close to F_{ori} as possible, which reflects Sam's reliability (i.e., performance). In this case, we choose the annotation with F1 score of 0.58.

By generating the missing annotations in the original dataset with the method above, we could have an augmented dataset.

F Explanation of Worker Selection on Building New Datasets

When creating new datasets, we expect to have a few (e.g. five) experts and a relatively large group of (e.g. a hundred) crowd workers available for annotation.

At each time step, we select a group of (e.g. 20) crowd workers, and request them to annotate a few (e.g. 5) sentences, resulting in 4 crowd annotations on each sentence. Now we calculate the agreement of the annotations on each sentence, if the agreement is high (e.g. greater than 0.4), we use the MV aggregation of the crowd annotations as the ground truth, and calculate the F1 scores of each worker's annotation. Otherwise, we ask an expert to give an annotation on the sentence, and calculate the F1 score of each worker on the expert annotation. Note that the expert annotates only when the agreement is low. After this time step, we have crowd annotations on the sentences and their F-scores, which can be used to update the average score of each worker. This procedure is repeated until we have enough annotations on every sentence.

In other words, the Expert+MV approach works with both crowd workers and experts available (e.g. on an online system) when building datasets. The Expert+MV is an iterative approach in which the expert annotates when needed. And it saves the

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**
TS (Exp.)	62.66	64.91	**63.20**	49.76	42.34	**45.69**	72.15	60.20	**65.63**
TS (MV)	59.82	61.90	60.25	44.81	40.71	42.36	67.72	56.05	61.34
TS (Exp.+MV)	61.66	64.03	**62.23**	47.20	42.36	**44.49**	70.66	59.07	**64.35**
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 6: Detailed P, R, and F$_1$ scores of all methods on the Chinese OEI dataset. All our algorithms perform significantly better than the Random (i.e., naive crowdsourcing) baseline.

cost of expert annotations by using the MV aggregation of crowd annotations as the ground truth when possible. Our experiment results show that the Expert+MV approach can save 47.19% of the cost of expert annotations on the Chinese OEI dataset, and 65.97% on the CoNLL'03 dataset respectively.

However, even in the case that no expert is available, which means that Expert+MV falls back to MV, we can still observe that the MV approach outperforms the Random baseline (which is an equivalent of normal crowdsourcing procedure which assigns an equal amount of sentences to each worker randomly) by a large gap. In this case, the MV approach saves 100% of expert annotation cost, but still produced crowd annotation with good quality. Please refer to Table 3 and Table 6 for more detailed results.

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 7: 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.