CROWD-IN-THE-LOOP: A Hybrid Approach for Annotating Semantic Roles

Chenguang Wang[†] **, Alan Akbik**[‡*]**, Laura Chiticariu**[†] **, Yunyao Li**[†] **, Fei Xia**[§] **, Anbang Xu**[†]
[†]IBM Research - Almaden
[‡]Zalando Research, Berlin
[§]Department of Linguistics, University of Washington
chenguang.wang@ibm.com, alan.akbik@zalando.de
{chiti, yunyaoli, anbangxu}@us.ibm.com, fxia@uw.edu

Abstract

Crowdsourcing has proven to be an effective method for generating labeled data for a range of NLP tasks. However, multiple recent attempts of using crowdsourcing to generate gold-labeled training data for semantic role labeling (SRL) reported only modest results, indicating that SRL is perhaps too difficult a task to be effectively crowdsourced. In this paper, we postulate that while producing SRL annotation does require expert involvement in general, a large subset of SRL labeling tasks is in fact appropriate for the crowd. We present a novel workflow in which we employ a classifier to identify difficult annotation tasks and route each task either to experts or crowd workers according to their difficulties. Our experimental evaluation shows that the proposed approach reduces the workload for experts by over two-thirds, and thus significantly reduces the cost of producing SRL annotation at little loss in quality.

1 Introduction

Semantic role labeling (SRL) is the task of labeling the predicate-argument structures of sentences with semantic *frames* and their *roles* (Baker et al., 1998; Palmer et al., 2005). It has been found useful for a wide variety of NLP tasks such as question-answering (Shen and Lapata, 2007), information extraction (Fader et al., 2011) and machine translation (Lo et al., 2013). A major bottleneck impeding the wide adoption of SRL is the need for large amounts of labeled training data to capture broad-coverage semantics. Such data requires trained experts and is highly costly to produce (Hovy et al., 2006).

Crowdsourcing SRL *Crowdsourcing* has shown its effectiveness to generate labeled data for a range of NLP tasks (Snow et al., 2008; Hong and Baker, 2011; Franklin et al., 2011). A core advantage of crowdsourcing is that it allows the annotation workload to be scaled out among large numbers of inexpensive *crowd workers*. Not surprisingly, a number of recent SRL works have also attempted to leverage crowdsourcing to generate labeled training data for SRL and investigated a variety of ways of formulating crowdsourcing tasks (Fossati et al., 2013; Pavlick et al., 2015; Akbik et al., 2016). All have found that crowd feedback generally suffers from low inter-annotator agreement scores and often produces incorrect labels. These results seem to indicate that, regardless of the design of the task, SRL is simply too difficult to be effectively crowdsourced.

Proposed Approach We observe that there are significant differences in difficulties among SRL annotation tasks, depending on factors such as the complexity of a specific sentence or the difficulty of a specific semantic role. We therefore postulate that a subset of annotation tasks is in fact suitable for crowd workers, while others require expert involvement. We also postulate that it is possible to use a classifier to *predict* whether a specific task is easy enough for crowd workers.

Based on these intuitions, we propose CROWD-IN-THE-LOOP, a hybrid annotation approach that involves both crowd workers and experts: All annotation tasks are passed through a decision function (referred to as TASKROUTER) that classifies them as either *crowd-appropriate* or *expert-required*, and sent to crowd or expert annotators accordingly. Refer to Figure 1 for an illustration of this workflow.

[*]The work was done while the author was at IBM Research - Almaden.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1913–1922
Copenhagen, Denmark, September 7–11, 2017.

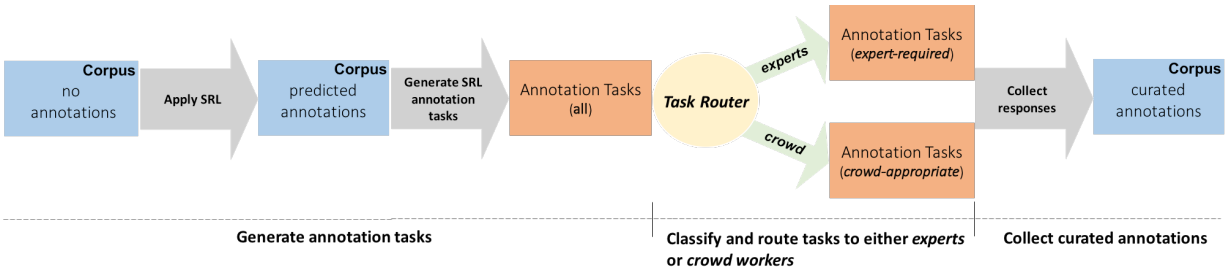

Figure 1: Overview of proposed CROWD-IN-THE-LOOP approach for curating SRL annotations.

We conduct an experimental evaluation that shows (1) that we are able to design a classifier that can distinguish between crowd-appropriate and expert-required tasks at very high accuracy (96%), and (2) that our proposed workflow allows us to pass over two-thirds of the annotation workload to crowd workers, thereby significantly reducing the need for costly expert involvement.

Contributions In detail, our contributions are:

- We propose CROWD-IN-THE-LOOP, a novel approach for creating annotated SRL data with both crowd workers and experts. It reduces overall labeling costs by leveraging the crowd whenever possible, and maintains annotation quality by involving experts whenever necessary.
- We propose TASKROUTER, an *annotation task decision function* (or *classifier*), that classifies each annotation task into one of two categories: *expert-required* or *crowd-appropriate*. We carefully define the classification task, discuss features and evaluate different classification models.
- We conduct a detailed experimental evaluation of the proposed workflow against several baselines including standard crowdsourcing and other hybrid annotation approaches. We analyze the strengths and weaknesses of each approach and illustrate how expert involvement is required to address errors made by crowd workers.

Outline This paper is organized as follows: We first conduct a baseline study of crowdsourcing SRL annotation, and analyze the difficulties of relying solely on crowd workers (Section 2). Based on this analysis, we define the classification problem for CROWD-IN-THE-LOOP, discuss the design of our classifier, and evaluate its accuracy (Section 3). We then employ this classifier in the proposed CROWD-IN-THE-LOOP approach and comparatively evaluate it against a number of crowdsourcing and hybrid workflows (Section 4). We discuss related work (Section 5) and conclude the study in Section 6.

2 Crowdsourcing SRL

We first conduct a baseline study of crowdsourcing SRL. We illustrate how we design and create annotation tasks, how we gather and interpret crowd feedback, and analyze the results of the study to determine the applicability of crowdsourcing for producing SRL annotation.

SRL formalism. In this study, and throughout the paper, we use the PROPBANK formalism of SRL (Palmer et al., 2005), which defines verb-specific frames (BUY.01, BUY.02), frame-specific core roles (**A0** to **A5**), and frame-independent non-core roles (for temporal, location and other contexts).

2.1 Annotation Task Design

To design the annotation task, we replicate a setup proposed in previous work (Akbik et al., 2016) in which crowd workers are employed to *curate* the output of a statistical SRL system. This setup generates annotation tasks as following:

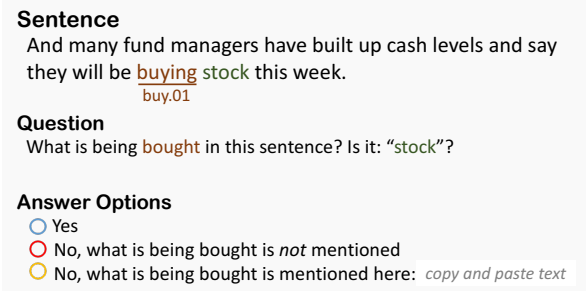

Figure 2: Example annotation task, consisting of a *sentence* with predicted role labels, a human readable *question* regarding to one label, and a set of answer *options*. By answering, crowd workers curate a prediction made by the SRL.

Step 1. We use a statistical SRL system to predict SRL labels for a set of sentences (see Figure 1). While state-of-the-art SRL will predict many correct labels, some predicted labels will be incorrect, and some labels will be missing. Annotation tasks are therefore designed to detect and correct precision and recall errors.

Step 2. We generate two types of annotation tasks for the study, namely CONFIRMPREDICTION and ADDMISSING tasks: (1) The first, CONFIRMPREDICTION tasks, ask users to confirm, reject or correct each predicted frame or role. This type of task addresses *precision* issues in the SRL. We present to workers a human-readable question-answer pair (He et al., 2015) for each predicted label, an example of which is illustrated in Figure 2. (2) The second, ADDMISSING tasks, address potentially missing annotation, i.e. *recall* issues in the SRL. We generate a question without a suggested answer and ask workers to either confirm that this role does not appear in the sentence, or supply the correct span. We identify potentially missing annotation using PropBank frame definitions; any unseen core role in a sentence is considered potentially missing.

We use a manually created mapping of frame-roles to questions to generate these tasks. See Table 1 for a mapping of the roles of the BUY.01 frame to questions.

Step 3. Each question is presented to crowd workers together with the sentence and a set of answer options. Example annotation tasks are illustrated in Figures 2 and 3. A task thus is defined as follows:

Definition 1 Annotation Task*: A task consists of a sentence, a human readable question regarding a predicted label, and a set of answer options.*

We collect worker responses to these tasks. If the majority of crowd workers agrees on a correction, we remove or correct incorrectly predicted labels for CONFIRMPREDICTION tasks and add new labels for ADDMISSING tasks.

Frame: BUY.01 (*purchase*)		
Role	Description	Question
A0	buyer	Who is buying something?
A1	thing bought	What is being bought?
A2	seller	From whom is something bought?
A3	price paid	What is the price paid?
A4	benefactive	For whom is something bought?

Table 1: Examples of mapping between semantic labels and question phrases of frame BUY.01. The description column lists the textual role descriptions from PropBank frame files.

Agreement	#Tasks	#Correct	#Incorrect	Precision
all 5 agree on answer	1,801	1,679	122	**0.93**
4 out of 5 agree	436	376	60	0.86
3 out of 5 agree	278	187	91	0.67
no majority answer	34	0	34	0.0
total	2,549	2,242	307	0.88

Table 2: Tasks in our crowdsourcing study by ratio of how many workers agreed on an answer. If all five workers agree, the majority answer is correct in 93% of cases. If fewer workers agree, the precision of the majority answer decreases.

2.2 Crowdsourcing Study

We conduct a crowdsourcing study consisting of 2,549 annotation tasks, generated by running a state-of-the-art SRL system (Akbik and Li, 2016) over 250 randomly selected gold-labeled sentences from the English training dataset in the CoNLL-2009 shared task (Hajič et al., 2009). We generated tasks using our question mappings from the predicted labels. This setup allows us to compare crowd feedback to gold labels and determine how often the crowd provides incorrect answers.

Human Annotators For *crowd annotators*, we employ five native speakers of English from UPWORK[1], selected using the following procedure: We required workers to complete a short tutorial[2], followed by 20 annotation tasks, which we evaluated against the gold data. We used the results to select the best-scoring 5 of 7 applicants. We then asked them to complete the remaining labeling tasks. The study was conducted in a span of three weeks. Crowd workers were paid a fixed sum for the completion of the study, which resulted in a cost of 2 cents per worker per task. In total, workers estimated an average of 9 hours to complete the full task.

2.3 Analysis

We gather crowd feedback and compare the majority answer for each task with the gold label. Refer to Table 2 for an overview of results. We make several observations:

The more workers agree, the better the answer. Generally, we note that majority answers tend to be more often correct if more workers agree. Specifically, as Table 2 shows, all 5 annota-

[1] https://www.upwork.com/

[2] The tutorial is available upon request.

Type		Frame	A0	A1	A2	A3	A4	LOC	TMP
CONFIRM-PREDICTION	Expert-required	**134 (38%)**	280 (32%)	382 (32%)	73 (33%)	**6 (43%)**	**8 (50%)**	36 (35%)	120 (36%)
	Crowd-appropriate	222 (62%)	608 (68%)	797 (68%)	146 (67%)	8 (57%)	8 (50%)	67 (65%)	211 (64%)
ADD-MISSING	Expert-required	0	82 (31%)	54 (33%)	**240 (37%)**	99 (34%)	**72 (38%)**	0	0
	Crowd-appropriate	0	186 (69%)	111 (67%)	405 (63%)	190 (66%)	120 (62%)	0	0

Table 3: Breakdown of annotation tasks by question types and semantic labels, and proportion of expert-required tasks (formally defined in Section 3). Percentages in each cell add up to 100%. On average, 34% of tasks are expert required. Task types that lie above this average are highlighted bold. For instance, 38% of all frame confirmation questions are expert-required, indicating that this question type is of above-average difficulty.

tors agreed in 1,801 out of all 2,549 tasks (71%). Of these tasks, the majority answer was correct in 1,679 cases, and incorrect in 122, yielding a precision of 93% for tasks in full agreement. If only 4 out of 5 agree (i.e. one annotator provided a different answer), the precision drops to 86%. If only three annotators agree on an answer, the precision is even lower, at 67%. Furthermore, we note 34 cases in which there was no majority answer (no agreement by at least 3 workers). We therefore see a direct correlation between agreement scores and the validity of majority answers.

Even if all workers agree, errors are made. We also note that all 5 crowd workers sometimes unanimously agree an *incorrect annotation*, in a total of 122 cases. To illustrate such a case, consider the example in Figure 3: In our study, all 5 workers incorrectly selected **yes** as answer. However, (perhaps somewhat counterintuitively to non-experts) under the PropBank paradigm it is the "phone representative" that provide explicit help in this sentence, not "Vanguard."

Characteristics of difficult annotation tasks. As illustrated in Table 3, we break down annotation tasks by question types and semantic labels to gain a better understanding of which tasks are difficult for the crowd. The first row in the table lists results for CONFIRMPREDICTION tasks. We note that some tasks of this type require above-average expert involvement, such as confirmation questions that pertain to the frame label or higher numbered roles (A3 and A4). The second row lists results for ADDMISSING tasks. Here, we note that again higher order roles tend to be above average expert-required[3]. However, while the breakdown in Table 3 indicates some general trends for the difficulty of annotation tasks, the question type itself does not suffice to determine whether an individual instance requires expert involvement or not.

Summary. Our crowdsourcing study supports the initial hypothesis that a portion of SRL tasks is in fact appropriate for crowd workers, but also shows that identifying such tasks is not straightforward since neither crowd agreement scores nor the annotation task type is sufficient indicators of difficult tasks. We investigate this problem further in the next section.

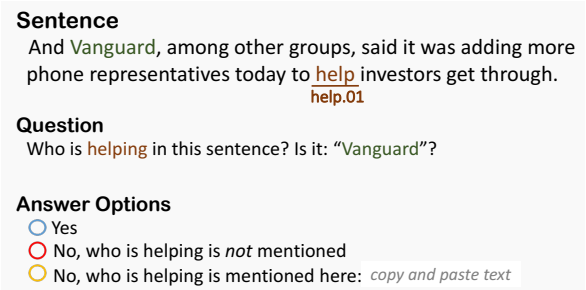

Figure 3: Example of an annotation task where crowd workers unanimously provided an incorrect answer in our study (see 2.3). This task is classified as *expert-required*.

3 TASKROUTER: Annotation Task Classification

Our study shows that some annotation tasks are appropriate for crowd workers, while others are not. In this section, we define a classification problem in which we wish to classify each task into one of the two following classes:

Definition 2 Crowd-appropriate: *A task for which: (1) All crowd workers agree on the answer. (2) The agreed-upon answer is correct.*

Definition 3 Expert-required: *A task that is not crowd-appropriate.*

According to these definitions, our crowdsourcing study found that the task in Figure 2 is *crowd-appropriate*, i.e. easy enough for the crowd to provide correct and consistent answers, while the task in Figure 3 is considered *expert-required*.

[3]Note that there are no ADDMISSING questions for frames since our SRL predicts a label for each verb in a sentence. Also there are no missing optional arguments since we ask missing argument questions only for core roles.

3.1 Features

To solve the task classification problem, we note two groups of distinct features (see Table 4):

Task-level features $\mathbf{X}^g$ capture the general difficulty of a labeling task, as defined by a frame or role type. The intuition here is that certain frames/roles are inherently difficult for non-experts, and that annotation tasks related to such frames/roles should be handled by experts. In the BUY.01 frame for instance, *buyer* (**A0**) is a simple crowd-appropriate semantic concept, while *benefactive* (**A4**) generally produces lower agreement scores. Task-level features therefore include the frame and role labels themselves, as well as the complexity of each question, measured in features such as the question word (*what*, *how*, *when* etc.), its length measured in number of tokens, and all tokens, lemmas and POS-tags in the question.

Sentence-level features $\mathbf{X}^l$ capture complexity associated with the specific *task instance*. The intuition is that some sentences are more complex and more difficult to understand than others. In such sentences, even roles with generally crowd-appropriate definitions might be incorrectly answered by non-experts. We capture the complexity of a sentence with features such as its length (number of tokens in sentence), the numbers of frames, roles, verbs, and nouns in the sentence, as well as all tokens, lemmas and POS-tags.

3.2 Classification Model

In addition to task- and sentence-level features, we present a classifier that also models the interplay between multiple annotation tasks generated from the same sentence. The intuition here is that there is an interdependence between labeling decisions in the same sentence. For instance, the presence of a difficult role may alter the interpretation of a sentence and make other labeling decisions more complicated. We thus propose a fuzzy classification model with two layers (Ishibuchi et al., 1995) of SVM classifiers (Wang et al., 2016), which introduces the context of the task using fuzzy indicators to model the interplay between the two groups of features.

Type	Features
Task-level features	Frame label; role label; question type; length of question in # tokens; Wh-word; tokens, lemmas, POS tags of all words in question.
Sentence-level features	# of questions for sentence, # of question types for sentence; # of verbs, # of nouns, # of frames, # of roles in sentence; length of sentence in # tokens; tokens, lemmas and POS tags of all words in sentence; head word and dependency relation to head word.

Table 4: Features for annotation task classification.

Specifically, we train a *local-layer SVM classifier* $\mathcal{L}^l$ using the sentence-level features $\mathbf{X}^l$ (computed from sentences). We also train a *global-layer SVM classifier* $\mathcal{L}^g$ using the task-level features $\mathbf{X}^g$ (computed from tasks). We refer to the predictions of the local and global classifiers as *fuzzy indicators* and we incorporate them as additional features to the fuzzy two-layer SVM classifier $\mathcal{L}^f$ as follows. Given task a_i among all tasks a_1 to a_n for a sentence s, the first layer of the fuzzy classifier, consists of applying the local-layer classifier using the sentence-level features of s. The second layer of the fuzzy classifier consists in applying the global-layer classifier n times, each time using task-level features for task $a_j, 1 \leq j \leq n$, resulting in $n+1$ values: one *local-layer indicator*, and n *global-layer indicators*. Our final fuzzy classifier model uses the $n+1$ local and global indicators as features, in addition to the sentence- and task-level features of a_i.

Note that the classification of task a_i considers features from other tasks a_j from the same sentence, but more efficiently than placing all task-level features of all tasks into a single feature vector. Formally, the objective function of the fuzzy two-layer SVM classification model $\mathcal{L}^f$ is:

$$\max_{\alpha} \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Y}\mathbf{K}(\mathbf{X}^{f^T}\mathbf{X}^f)\mathbf{Y}\boldsymbol{\alpha} \quad (1)$$
$$s.t. \quad \boldsymbol{y}^T\boldsymbol{\alpha} = 0, 0 \leq \boldsymbol{\alpha} \leq C\mathbf{1}.$$

where $\mathbf{K}(\mathbf{X}^{f^T}\mathbf{X}^f)$ is the fuzzy two-layer RBF kernel function, $\mathbf{X}^f = [\mathbf{X}^{g^T}, \mathbf{X}^{l^T}, \mathbf{Y}_1{}^{g^T}, \cdots, \mathbf{Y}_j{}^{g^T}, \cdots, \mathbf{Y}_n{}^{g^T}, \mathbf{Y}^{l^T}]$ is the fuzzy two-layer feature matrix, n is the number of annotation tasks generated from a sentence, $\mathbf{Y}_j{}^g$ represents the j-th fuzzy indicator generated by the j-th global classifier $\mathcal{L}^g{}_j$, $\mathbf{Y}^l$ is the fuzzy indicator generated by the local classifier $\mathcal{L}^l$, $\mathbf{Y}$ is the label matrix, $\mathbf{1}$ is a vector of all ones and C is a positive trade-off parameter.

3.3 Evaluation

To evaluate the accuracy of TASKROUTER we use the standard measure of *accuracy* for binary classifiers. As Table 5 shows, we evaluate four setups

Approach	Accuracy
SVM_{task}: Task features only	0.91
$\text{SVM}_{sentence}$: Sentence features only	0.87
$\text{SVM}_{task+sentence}$: All features	0.94
TASKROUTER: Fuzzy two-layer	**0.96***

Table 5: Performance of classifiers trained with five-fold cross validation on training set. The improvements of TASKROUTER over other classifiers are significant at the * 0.05 level, paired t-test.

in which we train an SVM with (1) task-level features, (2) sentence-level features, (3) all features, and (4) our proposed fuzzy two-layer classifier.

Data. We use the dataset created in our crowdsourcing study (see Section 2.2), which consists of 2,549 annotation tasks labeled as either *expert-required* or *crowd-appropriate* according to our definitions and the results of the study. We leverage five-fold cross validation to train the classifiers over a training split (80%).

Results. The cross validation results are listed in Table 5. Our proposed classifier outperforms all baselines and reaches a classification accuracy of **0.96**. Interestingly, we also note that task-level features are significantly more important than sentence-level features, as the setup SVM_{task} outperforms $\text{SVM}_{sentence}$ by 6 accuracy points. Furthermore, our proposed approach outperforms $\text{SVM}_{task+sentence}$, indicating a positive impact of modeling the global interplay of annotation tasks.

These experiments confirm our initial postulation that it is possible to train a high quality classifier to detect expert-required tasks. We refer to the best performing setup as TASKROUTER.

4 CROWD-IN-THE-LOOP Study

Having created TASKROUTER, we now execute our proposed CROWD-IN-THE-LOOP workflow and comparatively evaluate it against a number of crowdsourcing and hybrid approaches. We wish to determine (1) to what degree does having the crowd in the loop reduce the workload of experts? (2) How does the quality of the produced annotated data compare to purely crowdsourced or expert annotations?

4.1 Approaches

We evaluate the following approaches:

1. Baseline without curation The first is a simple baseline in which we use the output of SRL as-is, i.e. with no additional curation either by the crowd or experts. We list this method to show the quality of the starting point for the curation workload.

2. CROWD (Crowdsourcing) The second baseline is a standard crowdsourcing approach as described in Section 2, i.e. without experts. We send all annotation tasks (100%) to the crowd and gather crowd feedback to correct labels in three different settings. We correct all labels based on majority vote, i.e., if at least 3 (CROWD_{min3}), 4 (CROWD_{min4}) or all 5 (CROWD_{all5}) out of 5 annotators agree on an answer.

3. HYBRID (Crowdsourcing + Expert curation) In this setting, we replicate the approach proposed by (Akbik et al., 2016): After first executing crowdsourcing (i.e. sending 100% of the tasks to the crowd), we identify all tasks in which crowd workers provided conflicting answers. These tasks are sent to experts for additional curation (expert answers are used for curation instead of the crowd response). We use three definitions of what constitutes a conflicting answer: (1) We consider all answers in which at least a majority (3 out of 5) agreed as crowd-appropriate and send the rest (2.2%) to experts. We refer to this setup as HYBRID_{min3}. (2) Only tasks where 4 out of 5 agreed are crowd-appropriate, the remaining 9.9% go to experts (HYBRID_{min4}). (3) Any task in which there is no unanimous agreement (27.3%) is deemed expert-required (HYBRID_{all5}).

4. CROWD-IN-THE-LOOP This setup is *the proposed approach* in which we use TASKROUTER trained over a holdout training set to split annotation tasks into crowd and expert groups. In our experiments, TASKROUTER determines the following partitions: 66.4% of tasks to the crowd, the remaining 33.6% to experts. To give an indication of the lower bound of the approach given these partitions, we list results for two settings: (1) $\text{CROWD-IN-THE-LOOP}_{Random}$, a lower bound setting in which we randomly split into these partitions. (2) $\text{CROWD-IN-THE-LOOP}_{TaskRouter}$, the proposed setting in which we use TASKROUTER to perform this split.

Refer to Table 6 for an overview of these experiments. The WORKLOAD columns indicate what percentage of tasks is sent to crowd and experts.

4.2 Experimental Setup

Data We use the dataset created in the crowdsourcing study in Section 2, consisting of 2,549 annotation tasks labeled either as expert-required

Approach	Annotation Quality			Workload		Correctness	
	P	R	F1	crowd	expert	crowd-only	hybrid
Baseline without curation	0.86	0.83	0.85	0%	0%	-	-
Crowd$_{min3}$	0.92	0.88	0.90	100.0%	0%	0.84	0.84
Crowd$_{min4}$	0.89	0.85	0.87	100.0%	0%	0.84	0.84
Crowd$_{all5}$	0.87	0.84	0.85	100.0%	0%	0.84	0.84
Hybrid$_{min3}$	0.90	0.86	0.88	100.0%	2.2%	0.84	0.84
Hybrid$_{min4}$	0.91	0.87	0.89	100.0%	9.9%	0.84	0.86
Hybrid$_{all5}$	0.93	0.89	0.91	100.0%	27.3%	0.84	0.88
Crowd-in-the-Loop$_{Random}$	0.92	0.88	0.90	**66.4%**	**33.6%**	0.83	0.89
Crowd-in-the-Loop$_{TaskRouter}$	**0.96***	**0.92***	**0.94***	**66.4%**	**33.6%**	**0.92***	**0.95***

Table 6: Comparative evaluation of different approaches for generating gold-standard SRL annotation. The improvements of Crowd-in-the-Loop$_{TaskRouter}$ over other approaches are significant at the * 0.05 level, paired t-test.

or crowd-appropriate [4]. As shown in Section 3.3, we use 80% of the dataset to train TaskRouter under cross validation, and conduct the comparative evaluation using the remaining 20%.

Human annotators & curation We simulate an *expert annotator* using the CoNLL-2009 gold SRL labels and reuse the crowd answers from the study for *crowd annotators*. For each setting, we gather crowd and expert answers to the annotation tasks, and interpret the answers to curate the SRL labels that were produced by the statistical SRL system. After curation, we evaluate the resulting labeled sentences against gold-labeled data to determine the annotation quality in terms of precision, recall and F_1-score.

Evaluation Metrics Next to the quality of resulting annotations, we are interested to evaluate how effectively we integrate the crowd. We measure this in two metrics. (1) One is the percentage of tasks that go to the crowd and to experts respectively. Note that in the Hybrid setup, some tasks go to both crowd workers and experts, so that the percentages can add up to over a hundred percent. This information is illustrated in column Workload in Table 6. (2) The second is the overall validity of crowd feedback, referred to as *correctness*, measured as the ratio of correct answers among all answers retrieved from the crowd. We provide two values for correctness in Table 6, under column Correctness: The first is the correctness only over crowd feedback. Note that this value is the same for all Crowd and Hybrid setups since in these approaches 100% of annotation tasks are passed to the crowd. The second named *hybrid* is the overall correctness of the resolved answers with both expert and crowd feedback.

[4] We will release the dataset soon.

4.3 Experimental Results

The results of our experiments are summarized in Table 6. We make the following observations:

Crowd-in-the-Loop significantly increases annotation quality. Our evaluation shows that Crowd-in-the-Loop produces SRL annotation with significantly higher quality compared to crowdsourcing or hybrid scenarios. With a resulting F_1-score of 0.94, it outperforms the best performing crowdsourcing setup (0.90) by 4 points. More importantly, our proposed approach also outperforms other hybrid approaches that partially leverage experts. It outperforms the best hybrid approach (0.91) by 3 points, indicating that TaskRouter is better to select expert-required tasks than a method with only crowd agreement.

Significantly less expert involvement required. In our experiments, more than two-thirds of all tasks were determined to be crowd-appropriate by TaskRouter. This considerably reduces the need for expert involvement compared to expert labeling, while still maintaining relatively high annotation quality. In particular, our approach compares favorably to other hybrid setups in which a similar partition of tasks is completed by experts. Since TaskRouter is more capable to choose expert-required tasks than previous approaches, we achieve higher overall quality at similar levels of expert involvement.

Crowd workers more effective. As the correctness column in Table 6 shows, the selection of tasks by TaskRouter is more appropriate for the crowd in general. Their average correctness increases to 0.92, compared to 0.84 if the crowd completes 100% of the tasks.

4.4 Discussion and Outlook

The proposed approach far outperforms crowdsourcing and hybrid approaches in terms of annotation quality. In particular, even at similar levels of expert involvement, it outperforms the HYBRID$_{all5}$ approach. However, we also note that with an F_1-score of 0.94, our approach does not yet reach the quality of gold annotated data.

Insights for further improving quality. To further improve the quality of generated SRL training data, future work may (1) investigate additional features (Wang et al., 2015) and classification models to improve the TASKROUTER to better distinguish between crowd-appropriate and expert-required tasks, and (2) experiment with other SRL crowdsourcing designs to make more tasks crowd-appropriate. Nevertheless, we suspect that a small decrease in quality cannot be fully avoided if large amounts of non-experts are involved in a labeling task such as SRL. Given such involvement of non-experts, we believe that our proposed approach is a compelling way for increasing crowdsourcing quality while keeping expert costs relatively low.

Flexible trade-off of costs vs quality. Another avenue for research is to experiment with classifier parameters that allow us to more flexibly control the trade-off between how many experts we wish to involve and what annotation quality we desire (e.g., active learning (Wang et al., 2017)). This may be helpful to scenarios in which costs are fixed, or where one aims to compute the costs for generating annotated data of specific quality.

Use for SRL domain adaptation. One intended avenue for study is to apply our approach to generate training data for a specific textual domain for which little or no SRL training data currently exists. We believe that due to its relatively lower costs, our approach may be an ideal candidate for practical domain adaptation of SRL.

Applicability to other NLP crowdsourcing tasks. Finally, while in this paper we focused on the task of generating labeled training data for SRL, we believe that our proposed approach may be applicable to other NLP tasks that have only reported moderate results to-date. To study this applicability, one would first need to conduct a similar study as in Section 2 to identify crowd-appropriate and expert-required tasks and attempt the training of a classifier.

5 Related Work

Crowdsourcing SRL Annotation Different approaches have been adapted to formulate SRL tasks for non-expert crowd workers (Hong and Baker, 2011). Typical tasks include selecting answers from a set of candidates (Fossati et al., 2013), marking text passages that contain specific semantic roles (Feizabadi and Padó, 2014), and constructing question-answer pairs (He et al., 2015, 2016). However, a particular challenge is that SRL annotation tasks are often complex and crowdsourcing inevitably leads to low-quality annotations (Pavlick et al., 2015).

Instead of attempting to design a better annotation task, our proposed approach addresses this problem by accepting that a certain portion of annotation tasks is too difficult for the crowd. We create a classifier to identify such tasks and involve experts whenever necessary.

Routing Tasks Recent approaches have been developed to address the task routing problem in crowdsourcing (Bragg et al., 2014; Bozzon et al., 2013; Hassan and Curry, 2013). As workers vary in skill and tasks vary in difficulty, prior recommended approaches often consider the match between the task content and workers' profiles. However, these approaches are difficult to apply to the particular context of SRL annotation since we only distinguish between either experts familiar with PropBank, or non-expert crowd workers.

Rather than routing tasks to the most appropriate workers, our proposed approach determines which SRL tasks are appropriate for crowdsourcing, and sends the remaining ones to experts.

Human-in-the-loop Methods Our method is similar in the spirit of human-in-the-loop learning (Fung et al., 1992; Li et al., 2016). Human-in-the-loop learning generally aims to leverage humans to complete easy commonsense tasks, such as the recognition of objects in images (Patterson et al., 2013). Recent work also proposed human-in-the-loop parsing (He et al., 2016) to include human feedback into parsing. However, unlike these approaches, we aim to combine both experts and non-experts to address the difficulty of some SRL annotation tasks, while leveraging the crowd for the majority of tasks.

6 Conclusion

In this paper, we proposed CROWD-IN-THE-LOOP an approach for creating high-quality annotated

data for SRL that leverages both crowd and expert workers. We conducted a crowdsourcing study and analyzed its results to design a classifier to distinguish between crowd-appropriate and expert-required tasks. Our experimental evaluation showed that our proposed approach significantly outperforms baseline crowdsourcing and hybrid approaches, and successfully limits the need for expert involvement while achieving high annotation quality.

References

Alan Akbik and Yunyao Li. 2016. K-srl: Instance-based learning for semantic role labeling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, pages 599–608.

Alan Akbik, Kumar Vishwajeet, and Yunyao Li. 2016. Towards semi-automatic generation of proposition banks for low-resource languages. In *Conference on Empirical Methods on Natural Language Processing*, pages 993–998.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Annual Meeting of the Association for Computational Linguistics*, pages 86–90.

Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci. 2013. Choosing the right crowd: expert finding in social networks. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 637–648.

Jonathan Bragg, Andrey Kolobov, Mausam Mausam, and Daniel S Weld. 2014. Parallel task routing for crowdsourcing. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Conference on Empirical Methods on Natural Language Processing*, pages 1535–1545.

Parvin Sadat Feizabadi and Sebastian Padó. 2014. Crowdsourcing annotation of non-local semantic roles. In *European Chapter of the Association for Computational Linguistics*, pages 226–230.

Marco Fossati, Claudio Giuliano, and Sara Tonelli. 2013. Outsourcing framenet to the crowd. In *Annual Meeting of the Association for Computational Linguistics(2)*, pages 742–747.

Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. 2011. Crowddb: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 61–72.

Patrick T Fung, Graham Norgate, Timothy A Dilts, Andrew S Jones, and Rangaswamy Ravindran. 1992. Human-in-the-loop machine control loop. US Patent 5,116,180.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18.

Umairul Hassan and Edward Curry. 2013. A capability requirements approach for predicting worker performance in crowdsourcing. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, pages 429–437.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Conference on Empirical Methods on Natural Language Processing*, pages 643–653.

Luheng He, Julian Michael, Mike Lewis, and Luke Zettlemoyer. 2016. Human-in-the-loop parsing. In *Conference on Empirical Methods in Natural Language Processing*.

Jisup Hong and Collin F Baker. 2011. How good is the crowd at real wsd? In *Proceedings of the 5th linguistic annotation workshop*, pages 30–37.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Human Language Technology Conference of the NAACL*, pages 57–60.

Hisao Ishibuchi, Ken Nozaki, Naohisa Yamamoto, and Hideo Tanaka. 1995. Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on fuzzy systems*, 3(3):260–270.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

Chi-kiu Lo, Meriem Beloucif, and Dekai Wu. 2013. Improving machine translation into chinese by tuning against chinese meant. In *IWSLT 2013, 10th International Workshop on Spoken Language Translation*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Genevieve Patterson, Grant Van, Horn Serge, Belongie Pietro, and Perona James Hays. 2013. Bootstrapping fine-grained classifiers: Active learning with a crowd in the loop. In *Neural Information Processing Systems (Workshop)*.

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. 2015. Framenet+: Fast paraphrastic tripling of framenet. pages 408–413.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Conference on Empirical Methods on Natural Language Processing-CoNLL*, pages 12–21.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263.

Chenguang Wang, Laura Chiticariu, and Yunyao Li. 2017. Active learning for black-box semantic role labeling with neural factors. In *IJCAI*, page to appear.

Chenguang Wang, Yangqiu Song, Ahmed El-Kishky, Dan Roth, Ming Zhang, and Jiawei Han. 2015. Incorporating world knowledge to document clustering via heterogeneous information networks. In *KDD*, pages 1215–1224.

Chenguang Wang, Yangqiu Song, Haoran Li, Ming Zhang, and Jiawei Han. 2016. Text classification with heterogeneous information network kernels. In *AAAI*, pages 2130–2136.