Effective Crowd Annotation for Relation Extraction

Angli Liu, Stephen Soderland, Jonathan Bragg,
Christopher H. Lin, Xiao Ling, and Daniel S. Weld
Turing Center, Department of Computer Science and Engineering
Box 352350
University of Washington
Seattle, WA 98195, USA
{anglil, soderlan, jbragg, chrislin, xiaoling, weld} at cs.washington.edu

Abstract

Can crowdsourced annotation of training data boost performance for relation extraction over methods based solely on distant supervision? While crowdsourcing has been shown effective for many NLP tasks, previous researchers found only minimal improvement when applying the method to relation extraction. This paper demonstrates that a much larger boost is possible, e.g., raising F1 from 0.40 to 0.60. Furthermore, the gains are due to a simple, generalizable technique, *Gated Instruction*, which combines an interactive tutorial, feedback to correct errors during training, and improved screening.

1 Introduction

Relation extraction (RE) is the task of identifying instances of relations, such as *nationality (person, country)* or *place_of_birth (person, location)*, in passages of natural text. Since RE enables a broad range of applications — including question answering and knowledge base population — it has attracted attention from many researchers. Many approaches to RE use supervised machine learning, e.g., (Soderland et al., 1995; Califf and Mooney, 1997; Lafferty et al., 2001), but these methods require a large, human-annotated training corpus that may be unavailable.

In response, researchers developed methods for *distant supervision* (DS) in which a knowledge base such as Wikipedia or Freebase is used to automatically tag training examples from a text corpus (Wu and Weld, 2007; Mintz et al., 2009). Indeed, virtually all entries to recent TAC KBP relation extraction competitions use distant supervision (Ji and Grishman, 2011). However, distant supervision provides noisy training data with many false positives, and this limits the precision of the resulting extractors (see Section 2). A natural assumption is that human-annotated training data, either alone or in conjunction with distant supervision, would give better precision. In particular, Snow et al. (2008) showed that, for many NLP tasks, crowdsourced data is as good as or better than that annotated by experts.

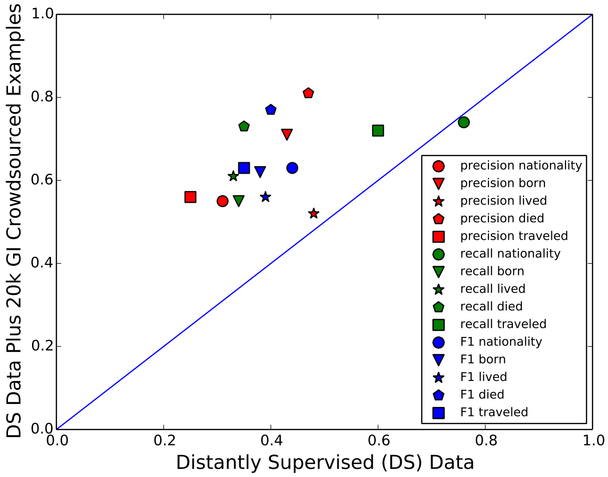


Figure 1: Adding 20K crowdsourced instances, acquired using Gated Instruction, to 700K examples from distant supervision raises precision, recall, and F1 for nearly all relations and raises overall F1 from 0.40 to 0.60 with MIML-RE learning.

It is quite surprising, therefore, that researchers who have applied crowdsourced annotation to relation extraction argue the *opposite*, that crowdsourcing provides only minor improvement:

- Zhang et al. (2012) conclude that “Human feedback has relatively small impact on precision and recall.” Instead, they advise applying distant supervision to vastly more data.
- Pershina et al. (2014) assert “Simply taking the union of the hand-labeled data and the corpus labeled by distant supervision is not effective since hand-labeled data will be swamped by a larger amount of distantly labeled data.” Instead, they introduce a complex feature-creation approach which improves the F1-score of MIML-RE, a state-of-the-art extractor (Surdeanu et al., 2012), just 4%, from 0.28 to 0.32 on a set of 41 TAC KBP relations.
- Angeli et al. (2014) explored a novel active learning method to control crowdsourcing, but found no improvement from adding the crowdsourced training to distant supervision using the default settings of MIML-RE, and only a 0.04 improvement in F1 when they initialized MIML-RE using the crowdsourced training.

This paper reports quite a different result, showing up to a 0.20 boost to F1. By carefully designing a quality-controlled crowdsourcing workflow that uses *Gated Instruction* (GI), we are able to create much more accurate annotations than those produced by previous crowdsourcing methods. GI (summarized in Figure 2) includes an interactive tutorial to train workers, providing immediate feedback to correct mistakes during training. Workers are then screened by their accuracy on gold-standard questions while doing the annotation. We show that GI generates much better training data than crowdsourcing used by other researchers, and that this leads to dramatically improved extractors.

Adding GI-crowdsourced annotations of the example sentences selected by Angeli et al.'s active learning method provides a much larger boost to the performance of the learned extractors than when their traditional crowdsourcing methods are used. In fact, the improvement due to our crowdsourcing method substantially outweighs the benefits of Angeli et al.'s active learning strategy as well. In total, this paper makes the following contributions:

- We present the design of the Gated Instruction crowdsourcing workflow with worker training and screening that ensures high-precision annotations for relation extraction training data in the presence of unreliable workers.
- We demonstrate that Gated Instruction increases the annotation quality of crowdsourced training data, raising precision from 0.50 to 0.77 and recall from 0.70 to 0.78, compared to Angeli et al.'s crowdsourced tagging of the same sentences. We make the data available for future research (Section 4.1).
- Augmenting distant supervision with 10K of Angeli et al.'s training examples annotated using Gated Instruction boosts F1 from 0.40 to 0.47, compared to 0.43, the result from using Angeli et al.'s crowdsourced annotations.
- We demonstrate that improved crowdsourcing has a greater effect than Angeli et al.'s active learning approach. Adding 10K *randomly selected* sentences, labeled using Gated Instruction, to distantly supervised data raises F1 by 6 points, compared to the 3 point gain from adding Angeli et al.'s crowdsourced labels on their active-learning sample.
- In contradiction to Zhang et al.'s prior claims, we show that increasing amounts of crowdsourced data can dramatically improve extractor performance. When we augmented distant supervision with 20K instances using Gated Instruction, we show that F1 is raised from 0.40 to 0.60.
- Gated Instruction may also reduce the cost of crowdsourcing. We show that with the high quality Gated Instruction annotations, a single annotation is more effective than majority vote over multiple annotators.

Our results provide a clear lesson for future researchers hoping to use crowdsourced data for NLP tasks. Extreme care must be exercised in the details of the workflow design to ensure quality data and useful results.

2 Background and Related Work

Distant supervision (DS) is a method for training extractors that obviates the need for human-labeled training data by heuristically matching facts from a background knowledge base (KB) to a large textual corpus. Originally developed to extract biological relations (Craven and Kumlien, 1999), DS was later extended to extract relations from Wikipedia infoboxes (Wu and Weld, 2007) and Freebase (Mintz et al., 2009). Specifically, distant supervision uses the KB to find pairs of entities E_1 and E_2 for which a relation R holds. Distant supervision then makes that assumption that any sentence that contains a mention of both E_1 and E_2 is a positive training instance for $R(E_1, E_2)$.

Unfortunately, this assumption leads to a large proportion of false positive training instances. For example, Freebase asserts that Nicolas Sarkozy was born in Paris, but nearly all sentences in a news corpus that mention Sarkozy and Paris do not give evidence for a *place_of_birth* relation. To address this shortcoming, there have been attempts to model the relation dependencies as multi-instance multi-class (Bunescu and Mooney, 2007; Riedel et al., 2010) leading to state-of-the art extraction learners MultiR (Hoffmann et al., 2011) and MIML-RE (Surdeanu et al., 2012).

Additionally, other techniques developed to study the relation extraction problem have achieved certain success, including universal schemas (Riedel et al., 2013), and deep learning (Nguyen and Grishman, 2014). Despite these technical innovations, the best systems at the TAC-KBP evaluation[1] still require substantial human effort, typically hand-written extraction rules (Surdeanu and Ji, 2014).

Recently researchers have explored the idea of augmenting distant supervision with a small amount of crowdsourced annotated data in an effort to improve relation extraction performance (Angeli et al., 2014; Zhang et al., 2012; Pershina et al., 2014).

Zhang et al. (2012) studied how the size of the crowdsourcing training corpus and distant supervision corpus affect the performance of the relation extractor. They considered the 20 TAC KBP relations that had a corresponding Freebase relation. They added up to 20K instances of crowd data to 1.8M DS instances using sparse logistic regression, tuning the relative weight of crowdsourced and DS training. However, they saw only a marginal improvement from F1 0.20 to 0.22 when adding crowdsourced training to DS training, and conclude that human feedback has little impact.

Angeli et al. (2014) also investigated methods for infusing distant supervision with crowdsourced annotations in the Stanford TAC-KBP system. They experimented with several methods, including adding a random sample of annotated sentences to the training mix, and using active learning to select which sentences should be annotated by humans. Their best results were what they termed "Sample JS," training a committee of MIML-RE classifiers and then sampling the sentences to be crowdsourced weighted by the divergence of classifications.

Surprisingly, they found that the simple approach of just adding crowdsourced data to the training mix *hurt* extractor performance slightly. They conclude that the most important use for crowdsourced annotations is as a way to *initialize* MIML-RE, mitigating the problem of local minima during learning. When they initialized MIML-RE with 10K Sample JS crowdsourced instances and then trained on a combination of Sample JS crowdsourced and DS instances, this raised F1 from 0.34 to 0.38.

Pershina et al. (2014) also exploited a small set of highly informative hand-labeled training data to improve distant supervision. Rather than crowdsourcing, they used the set of 2,500 labeled instances from the KBP 2012 assessment data. They state that "Simply taking the union of the hand-labeled data and the corpus labeled by distant supervision is not effective since hand-labeled data will be swamped by a larger amount of distantly labeled data." Instead they use the hand-annotated data to learn *guidelines* that are included in a graphical prediction model that extends MIML-RE, trained using distant supervision. This raised F1 from 0.28 to 0.32 over a comparison system without the learned guidelines.

Gormley et al. (2010) filtered crowdsourced workers by agreement with gold questions and by noting which workers took fewer than three seconds per question. They reported good inter-annotator agreement, but did not build a relation extractor from their data.

Both Zhang et al. and Angeli et al. used traditional methods to ensure the quality of their crowdsourced data. Zhang et al. replicated each question three times and included a gold question (i.e., one with a known answer) in each set of five questions. They only used answers from workers who answered at least 80% of the gold-standard questions correctly.

[1] http://www.nist.gov/tac/

Angeli et al. included two gold-standard questions in every set of 15. They discarded sets in which both controls were answered incorrectly, and additionally discarded all submissions from workers who failed the controls on more than one third of their submissions. They collected five annotations for each example, and used the majority vote as the ground truth in their training. They did not report the resulting quality of their crowdsourced annotations, but did release their data, allowing us to measure its precision and recall (see Section 4.1).

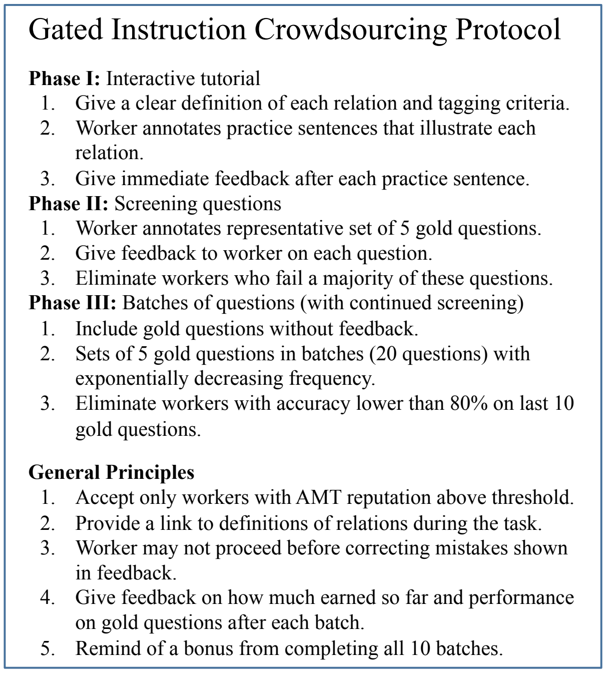
We argue that all these systems would have gotten better performance by focusing attention on the quality of their crowdsourced annotation. We demonstrate that by improving the crowdsourcing workflow, we achieve a higher F1 score, both with the crowdsourced training alone and in combination with distant supervision.

Our work adds to the existing large body of work that shows that crowdsourcing can be and is an effective and efficient method for training machine learning algorithms. Snow et al. (2008) showed that multiple workers can simulate an expert worker in a variety of natural language tasks. Many researchers (e.g., (Dawid and Skene, 1979; Whitehill et al., 2009)) have designed methods to aggregate crowd labels in order to reduce noise, and Sheng et al. (2008) showed that paying multiple crowd workers to relabel examples, as opposed to labeling new ones, can increase the accuracy of a classifier.

The effectiveness of crowdsourcing is dependent on a number of human factors. Several researchers have studied how worker retention is affected by payment schemes (Mao et al., 2013), recruitment techniques (Ipeirotis and Gabrilovich, 2014), or attention diversions (Dai et al., 2015). Ipeirotis and Gabrilovich show that volunteer workers may provide higher quality work. By contrast, we show that paid workers, too, can produce high quality work through careful attention to worker training and testing.

3 Gated Instruction Crowdsourcing

We used Amazon Mechanical Turk for our crowdsourcing, but designed our own website to implement the Gated Instruction (GI) protocol, rather than use the platform Amazon provides directly. This allowed us greater control over the UI and the worker experience. The primary benefit of GI is worker training, which is necessary across platforms, so we expect to see comparable results on other platforms, such as CrowdFlower.



Gated Instruction Crowdsourcing Protocol

Phase I: Interactive tutorial

1. Give a clear definition of each relation and tagging criteria.
2. Worker annotates practice sentences that illustrate each relation.
3. Give immediate feedback after each practice sentence.

Phase II: Screening questions

1. Worker annotates representative set of 5 gold questions.
2. Give feedback to worker on each question.
3. Eliminate workers who fail a majority of these questions.

Phase III: Batches of questions (with continued screening)

1. Include gold questions without feedback.
2. Sets of 5 gold questions in batches (20 questions) with exponentially decreasing frequency.
3. Eliminate workers with accuracy lower than 80% on last 10 gold questions.

General Principles

1. Accept only workers with AMT reputation above threshold.
2. Provide a link to definitions of relations during the task.
3. Worker may not proceed before correcting mistakes shown in feedback.
4. Give feedback on how much earned so far and performance on gold questions after each batch.
5. Remind of a bonus from completing all 10 batches.

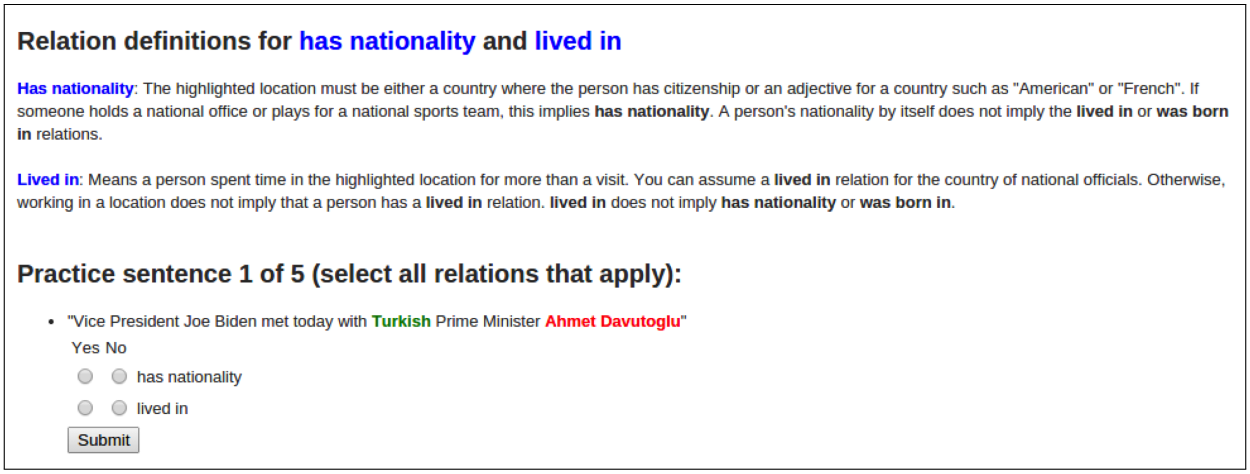
Figure 2: Architecture of the Gated Instruction protocol.

The ideas behind Gated Instruction are summarized in Figure 2. The workflow proceeds in three phases: tutorial, weed-out, and work (described below) with a focus on well-known user interface principles (rapid feedback and availability of extra help). While conceptually simple, we show this approach has a much bigger effect on the resulting learned NLP system than a more complex graphical model.

3.1 Interactive Tutorial Design

The most important step in crowdsourcing is ensuring that workers understand the task. To this end we required workers to complete an interactive tutorial to learn the criteria for the relations to be annotated.

Since we wanted to test our extractor against official answers for the TAC-KBP Slot Filling evaluation, our tutorial taught workers to follow the official KBP guidelines. These guidelines require tagging only relations directly stated in the sentence, and discourage plausible inferences. For example, if a sentence states only that a person works in a city, then annotating a *place_of_residence* relation with



Relation definitions for has nationality and lived in

Has nationality: The highlighted location must be either a country where the person has citizenship or an adjective for a country such as "American" or "French". If someone holds a national office or plays for a national sports team, this implies **has nationality**. A person's nationality by itself does not imply the **lived in** or **was born in** relations.

Lived in: Means a person spent time in the highlighted location for more than a visit. You can assume a **lived in** relation for the country of national officials. Otherwise, working in a location does not imply that a person has a **lived in** relation. **lived in** does not imply **has nationality** or **was born in**.

Practice sentence 1 of 5 (select all relations that apply):

- "Vice President Joe Biden met today with **Turkish** Prime Minister **Ahmet Davutoglu**"

Yes No
○ ○ has nationality
○ ○ lived in

Figure 3: Tutorial page that teaches guidelines for *nationality* and *lived_in*. The worker answers practice sentences with immediate feedback that teach each relation.

that city is counted as an error, even if it is probable that the person lives there.

Figure 3 shows a page from the tutorial that explains annotation guidelines for *nationality* and *lived_in* (i.e., *place_of_residence*). This figure shows the first page of the tutorial — as more relations are taught, those relations are added to the question. The real questions are asked in the same format later on for consistency. The worker can click on a link to see the relation definitions at any time during the tutorial or while doing the actual task. If workers make a mistake during the tutorial, they are given immediate feedback along with an explanation for the correct answer. The workers cannot proceed without correcting all errors on all problems in the tutorial.

3.2 Adaptive Worker Screening

After examining worker mistakes in a preliminary experiment, we manually selected a set of gold questions (i.e., questions with unambiguous, known answers) that workers are likely to get wrong if they don't clearly understand the annotation criteria. The gold questions are grouped into sets of 5 questions that represent all relations being annotated. The first 5 questions (the screening phase) are used to eliminate spammers and careless workers early on. These questions look no different from normal questions, but we give feedback to workers with the right answers if workers give wrong answers to any of these questions. If a worker fails a majority of such questions, the worker is disqualified from the task.

We then place additional sets of gold questions among real test questions without feedback in order to spot-check workers' responses. In our experience, workers who start out with high accuracy maintain that accuracy throughout the entire session. Therefore, we place the gold questions in exponentially decreasing frequency among the batches of 20 questions (5 gold questions in batches 2, 4, 8, etc.), and allow only workers who maintain at least 80% accuracy on the most recent 10 gold questions to continue with the task. Our task was not large enough to attract problems of collusion, but more lucrative or long-running tasks may require continual generation of new gold questions in order to combat sharing of answers among workers (Oleson et al., 2011). Techniques such as expectation maximization (Dawid and Skene, 1979) can be used to produce new gold questions based on worker answers.

3.3 Motivational Feedback

We want workers to stay motivated, so our crowdsourcing system also provides feedback to workers. In particular, workers receive adaptive per-batch message feedback at the end of each batch of questions (every 20 questions) about how well they did on the gold questions in the past batches, how much they have earned so far, and a reminder of the bonus for finishing all 10 batches. We paid workers $0.50 for each batch of 20 questions with a bonus of $1.00 for finishing 10 batches.

4 Experimental Results

In this section, we address the following questions:

- Does Gated Instruction produce training data with higher precision and recall than other research in crowdsourcing for relation extraction?
- Does higher quality crowdsourced training data result in higher extractor performance when adding crowdsourcing to distant supervision?
- How does the boost in extractor performance on random training instances labeled with Gated Instruction compare to that with instances labeled using traditional crowdsourcing techniques selected with active learning?
- How does extractor performance increase with larger amounts of Gated Instruction training data?
- What's the most cost-effective way to aggregate worker votes? Are multiple annotations needed, given high quality crowdsourcing?

4.1 Quality of Gated Instruction Training

We took the best training set of 10,000 instances from Angeli et al.'s 2014 system that selected training instances using active learning (their Sample JS data). In order to focus on the effect of crowdsourcing, we restricted our attention to four distinct relations between *person* and *location* that were used by previous researchers: *nationality*, *place_of_birth*, *place_of_residence*, and *place_of_death*[2]. We then sent these sentences to crowdsourced workers using the Gated Instruction protocol.

To evaluate the crowdsourced training data quality, we hand-tagged the crowdsourced annotations from both our Gated Instruction system and Angeli et al.'s work on 200 random instances. Annotations were considered correct if they followed the TAC-KBP annotation guidelines. Two authors tagged the sample with 87% agreement and then reconciled opinions to agree on consensus labels.

The training precision, recall, and F1 are shown in Figure 4. In this and all other experiments, aggregate statistics are macro-averaged across relations. We also include the training quality from Zhang et al., although this is on a different set of sentences and only for *place_of_birth*, *place_of_residence*, and *place_of_death*.

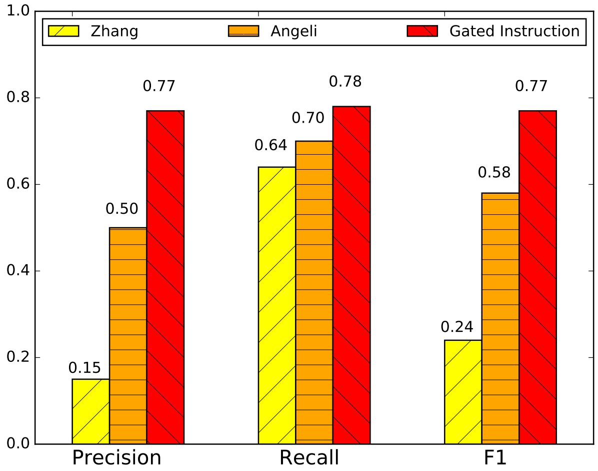


Figure 4: The training data produced by Gated Instruction has much higher precision and somewhat higher recall than that of Angeli et al. or Zhang et al.

Our Gated Instruction protocol gives higher F1 for the training set of each of the four relations we compared with Angeli's crowdsourcing on the same sentences. Our overall F1 was 0.77, compared to 0.58 for Angeli et al. and 0.24 for Zhang et al. The difference in precision is most dramatic, with our system achieving 0.77 compared to 0.50 and 0.15.

Worker agreement with GI was surprisingly high. Two workers agreed on between 78% to 97% of the instances, depending on the relation. The average agreement was 88%. The data is available for research purposes.[3]

4.2 Integrating Crowdsourced Data with the Relation Extraction Pipeline

The pipeline of our relation extraction system is as follows. First we collected sentences of training data from the TAC-KBP newswire corpus that contain a person and a location according to the Stanford NER tagger (Finkel et al., 2005). We represent them using the features described by Mintz et al. (2009). These features include NER tags of the two arguments, the dependency path between two designated arguments, the sequence of tokens between the ar-

[2] We collapsed the KBP relations *per:city_of_**, *per:stateorprovince_of_**, and *per:country_of_** into a single relation *place_of_**.

[3] `https://www.cs.washington.edu/ai/gated_instructions/naacl_data.zip`

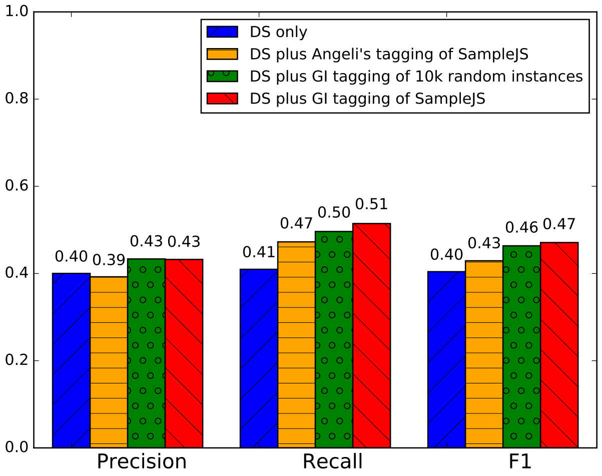


Figure 5: Adding 10K instances with Gated Instruction to 700K DS instances boosts F1 more than that of the original Sample JS annotations. Furthermore, GI applied to 10K randomly-selected instances outperforms active learning with traditional annotation.

guments, and the order of the arguments in the sentence.

We then split the data into 700K used for distant supervision and much smaller sets for crowdsourcing and for a held-out test set. For the experiments presented, unless otherwise noted, we used a variant of majority vote to create a training set. We obtained annotations from two workers for each example sentence and kept the instances where both agreed as our training data.

Finally, we ran a learning algorithm on the distant supervision training data, the crowdsourced training data, and a combination of the two. The results were evaluated on the hand-labeled test set.

4.3 Effect of Data Quality on Extractor Performance

We now study how the higher quality training data from our crowdsourcing protocol affects extractor performance, when it is added to a large amount of distantly-supervised data.

We compared adding the 10K crowdsourced instances from the previous experiment to 700K instances from distant supervision, where the crowdsourced data had tags from either Gated Instruction or the original crowdsourcing from Angeli et al. We compare only with Angeli et al. as we did not have annotations from Zhang et al. for the same training sentences.

We experimented using three learning algorithms: logistic regression, MultiR, and MIML-RE. We found that logistic regression gives the best results when applied to the crowdsourced training alone. With logistic regression, training on the 10K Sample JS instances gave F1 of 0.31 with Angeli et al.'s crowdsourced labels and 0.40 with Gated Instruction. Logistic regression is not a good fit for distant supervision — we had F1 of 0.34 from logistic regression trained on DS only.

MultiR and MIML-RE gave the best results for *combining* crowdsourcing with distant supervision. Each of these multi-instance multi-class learners had similar results, so we present results only for MIML-RE in the remainder of our experiments, as it is the learning algorithm used by other researchers.

We included no special mechanisms to prevent distant supervision data from swamping the smaller amount of crowdsourced data. MIML-RE has a built-in mechanism to combine supervised and distant supervision. It automatically builds a classifier from the supervised instances, uses this to initialize the distant supervision instance labels, and locks the supervised labels. With MultiR, we put the crowdsourced instances in separate singleton "bags" of training instances, since MultiR always takes at least one instance in each bag as truth.

As Angeli et al. found, it is important to use the crowdsourced training to initialize MIML-RE. With the default initialization, Angeli et al. report no gain in F1. We found a small gain in F1 even with the default initialization, but larger gains with crowdsourced initialization, which we use for the following experiments.

To see how much of the boost over distant supervision comes from the active learning that went into Angeli et al.'s sample JS training, we also used Gated Instruction on a *randomly selected* set of 10K newswire instances from the TAC KBP 2010 corpus (LDC2010E12) that contained at least one NER tag for *person* and one for *location*.

As Figure 5 shows, adding the Sample JS training with Gated Instruction crowdsourcing had a positive impact on performance, increasing precision from 0.40 to 0.43, recall from 0.41 to 0.51, and F1 from 0.40 to 0.47. With the original crowdsourced tag-

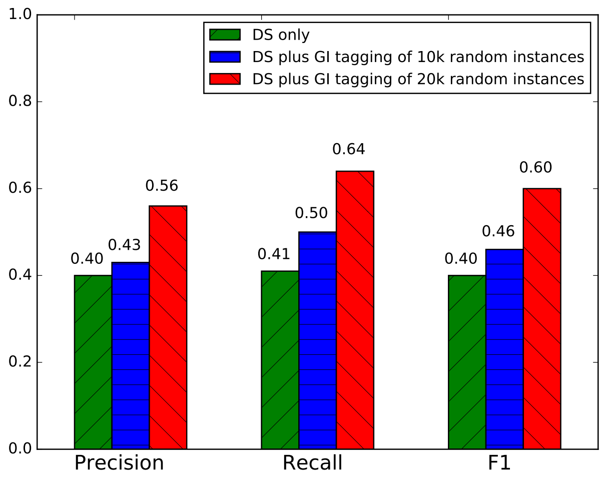


Figure 6: Adding 20K instances with Gated Instruction to DS gives a large boost to both precision and recall, raising F1 from 0.40 to 0.60.

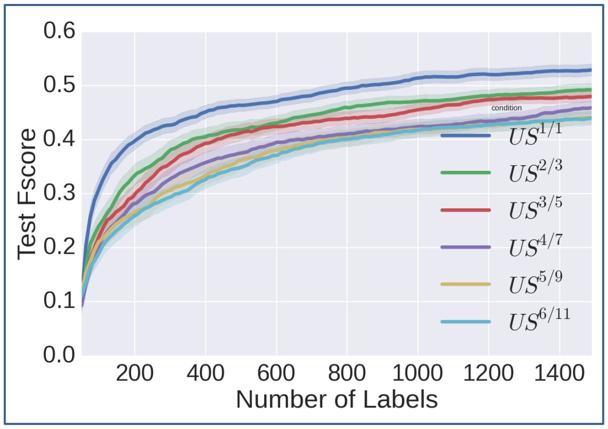


Figure 7: With high quality crowdsourcing, the simple policy of requesting a single annotation performs better than majority-vote of 3, 5, 7, 9, or 11 annotations (holding the annotation budget constant), since the increase in the *number* of data points outweighs the reduction in noise.

ging from Angeli et al., adding the crowdsourced instances actually caused a small *drop* in precision, a smaller gain in recall than Gated Instruction, and F1 of 0.43 — substantially less than achieved with labels from Gated Instruction.

Furthermore, in an apples-to-oranges comparison, we found that our improved crowdsourcing protocol had a much bigger impact than Angeli et al.'s active learning mechanism. Adding 10K *randomly* selected newswire instances tagged with Gated Instruction gave higher precision (0.43), recall (0.51), and F1 (0.46) than adding instances selected by active learning (Sample JS) when labeled using Angeli et al.'s protocol. In fact Gated Instruction gave double the improvement (6 points gain in F1 vs. 3). Of course, both of these numbers are small — bigger gains come from using the techniques together, and especially from using more crowdsourced data.

4.4 Effect of Data Quantity on Extractor Performance

Zhang et al. reported negligible improvement in F1 from adding 20K instances with their crowdsourcing to distant supervision, and Angeli et al. reported a gain of 0.04 F1 from adding 10K instances with active learning and their crowdsourcing.

As Figure 6 shows, Gated Instruction can raise F1 from 0.40 to 0.60 over distant supervision alone from adding 20K random newswire instances. This experiment uses all five relations that we crowdsourced, adding *travel_to* to the relations from Figure 5 that we had in common with Angeli et al. The results for DS only and 10K random instances are not significantly different from those in Figure 5 in which *travel_to* was omitted.

4.5 Comparison between Ways to Aggregate Annotations

In this section we explore the cost-effectiveness of alternate methods of creating training from Gated Instruction annotations. We compare a policy of using the majority vote of two out of three, or three out of five workers, and so forth, as opposed to soliciting a single annotation for each training sentence (unilabeling). Lin et al. (2014) show that in many settings, unilabeling is better because some classifiers are able to learn more accurately with a larger, noisier training set than a smaller, cleaner one.

With a given budget, single annotation gives three times as many training instances as the policy that uses three votes and five times as many as the policy that requires five votes, and so forth. Is the quality of data produced by Gated Instruction high enough to rely on just one annotation per instance?

We randomly select 2K examples from the 20K newswire instances and use Gated Instruction to acquire labels from 10 workers for each sentence. Figure 7 shows that when training a logistic regression classifier with high quality crowdsourcing data, a single annotation is, indeed, more cost effective than using a simple majority of three, five, or more annotations (given a fixed budget). The learning curves in Figure 7 use uncertainty sampling (US) to select examples from the 2000 available with the curves labeled US 1/1 for single votes, US 2/3 for two out of three, and so forth.

This is not to say that a single vote is always the best policy. It is another example of the impact of GI's high quality annotation. In the same domain of relation extraction, Lin et al. (2016) also show that with a more intelligent and dynamic relabeling policy, relabeling certain examples can still help.

5 Conclusion

This paper describes the design of Gated Instruction, a crowdsourcing protocol that produces high quality training data. GI uses an interactive tutorial to teach the annotation task, provides feedback during training so workers understand their errors, refuses to let workers annotate new sentences until they have demonstrated competence, and adaptively screens low-accuracy workers with a schedule of test questions. While we demonstrate GI for the task of relation extraction, the method is general and may improve annotation for many other NLP tasks.

Higher quality training data produces higher extractor performance for a variety of learning algorithms: logistic regression, MultiR, and MIML-RE. Contrary to past claims, augmenting distant supervision with a relatively small amount of high-quality crowdsourced training data gives a sizeable boost in performance. Adding 10K instances that Angeli et al. selected by active learning, annotated with Gated Instruction, raised F1 from 0.40 to 0.47 — substantially higher than the 0.43 F1 provided by Angeli et al.'s annotations. We also find that Gated Instruction is more effective than a complicated active learning strategy. Adding 10K randomly selected instances raises F1 to 0.46, and adding 20K random instances gave F1 of 0.60.

Our experimental results yield two main takeaway messages. First, we show that in contrast to prior work, adding crowdsourced training data substantially improves the performance of the resulting extractor as long as care is taken to ensure high quality crowdsourced annotations. We haven't yet experimented beyond person-location relations, but we believe that Gated Instruction is generalizable, particularly where there are clear criteria to be taught. We believe that Gated Instruction can greatly improve training data for other NLP tasks beside relation extraction as well.

Second, we provide practical and easily instituted guidelines for a novel crowdsourcing protocol, Gated Instruction, as an effective method for acquiring high-quality training data. It's important to break complex annotation guidelines into small, digestible chunks and to use tests (gates) to ensure that the worker reads and understands each chunk of the instructions before work begins. Without these extra checks, many poor workers pass subsequent gold tests by accident, polluting results.

Acknowledgment

This work was supported by NSF grant IIS-1420667, ONR grant N00014-15-1-2774, DARPA contract FA8750-13-2-0019, the WRF/Cable Professorship, and a gift from Google. We are very grateful to Chris Re and Ce Zhang who generously supplied their annotated data. Likewise, Masha Pershina and Ralph Grishman allowed us an early look at their paper and gave us their data. Similarly, Gabor Angeli and Chris Manning not only provided their data, but provided many insights about the best ways to combine crowd and distantly supervised annotations. Without this cooperation it would have been impossible to have done our work. We appreciate Natalie Hawkins, who helped prepare the data used in our study. We further thank Shih-Wen Huang and Congle Zhang for their suggestions on the design of the crowdsourcing system and the experiments.

References

Gabor Angeli, Julie Tibshirani, Jean Y. Wu, and Christopher D. Manning. 2014. Combining distant and partial supervision for relation extraction. In *EMNLP*.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

M. Califf and R. Mooney. 1997. Relational learning of pattern-match rules for information extraction. In *Workshop in Natural Language Learning, Conf. Assoc. Computational Linguistics*.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*.

Peng Dai, Jeffrey M Rzeszotarski, Praveen Paritosh, and Ed H Chi. 2015. And Now for Something Completely Different : Improving Crowdsourcing Workflows with Micro-Diversions. In *CSCW*.

A.P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Matthew R. Gormley, Adam Gerber, Mary Harper, and Mark Dredze. 2010. Non-expert correction of automatically generated relation annotations. In *Proceedings of NAACL and HLT 2010*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL*. Association for Computational Linguistics.

Panagiotis G. Ipeirotis and Evgeniy Gabrilovich. 2014. Quizz: targeted crowdsourcing with a billion (potential) users. In *WWW '14: Proceedings of the 23rd International Conference on the World Wide Web*.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1148–1158, Stroudsburg, PA, USA. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01.

Christopher H. Lin, Mausam, and Daniel S. Weld. 2014. To re(label), or not to re(label). In *HCOMP*.

Christopher H. Lin, Mausam, and Daniel S. Weld. 2016. Reactive learning: Active learning with relabeling. In *AAAI*.

Andrew Mao, Yiling Chen, Eric Horvitz, Megan E Schwamb, Chris J Lintott, and Arfon M Smith. 2013. Volunteering Versus Work for Pay: Incentives and Tradeoffs in Crowdsourcing. In *HCOMP*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 68–74.

David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. 2011. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In *Human Computation Workshop*, page 11.

Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. 2014. Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of ACL*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML-2010)*, pages 148–163.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. *NAACL HLT 2013*, pages 74–84.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

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. Association for Computational Linguistics.

S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. 1995. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1314–21.

Mihai Surdeanu and Heng Ji. 2014. Overview of the English slot filling track at the TAC2014 knowledge base population evaluation.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*, pages 455–465. Association for Computational Linguistics.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS'09*.

F. Wu and D. Weld. 2007. Autonomously semantifying Wikipedia. In *Proceedings of the ACM Sixteenth Conference on Information and Knowledge Management (CIKM-07)*, Lisbon, Portugal.

Ce Zhang, Feng Niu, Christopher Ré, and Jude Shavlik. 2012. Big data versus the crowd: Looking for relationships in all the right places. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 825–834. Association for Computational Linguistics.