# A Novel Workflow for Accurately and Efficiently Crowdsourcing Predicate Senses and Argument Labels

**Youxuan Jiang**[1], **Huaiyu Zhu**[2], **Jonathan K. Kummerfeld**[1], **Yunyao Li**[2], **Walter Lasecki**[1]
University of Michigan, Ann Arbor[1]
IBM Research, Almaden[2]
lyjiang@umich.edu huaiyu@us.ibm.com jkummerf@umich.edu
yunyaoli@us.ibm.com wlasecki@umich.edu

## Abstract

Resources for Semantic Role Labeling (SRL) are typically annotated by experts at great expense. Prior attempts to develop crowdsourcing methods have either had low accuracy or required substantial expert annotation. We propose a new multi-stage crowd workflow that substantially reduces expert involvement without sacrificing accuracy. In particular, we introduce a unique *filter* stage based on the key observation that crowd workers are able to almost perfectly filter out incorrect options for labels. Our three-stage workflow produces annotations with 95% accuracy for predicate labels and 93% for argument labels, which is comparable to expert agreement. Compared to prior work on crowdsourcing for SRL, we decrease expert effort by 4x, from 56% to 14% of cases. Our approach enables more scalable annotation of SRL, and could enable annotation of NLP tasks that have previously been considered too complex to effectively crowdsource.

## 1 Introduction

High quality data is crucial in NLP, but difficult to collect for complex tasks such as semantic role labeling (SRL). Annotating Propbank involved a team of annotators, each of whom took around three days to learn the annotation process (Palmer et al., 2005). For tasks such as sentiment analysis (Socher et al., 2013) and question answering (Rajpurkar et al., 2016), crowdsourcing has produced massive datasets that enabled the development of new, more sophisticated models. Recent work introduced a hybrid workflow to allow crowd workers to usefully contribute to annotation of SRL (Wang et al., 2017), but still required expert annotation in a third of cases.

This paper introduces a new hybrid SRL annotation workflow with the goal of minimizing expert annotation without sacrificing annotation accuracy. In order to develop our method, we first explored why SRL annotations are hard for crowd workers. We found that workers had difficulty identifying the correct answer because the number of options for labels in SRL can be overwhelming and workers lack the linguistic expertise to handle subtle cases. However, we also observed that (1) non-expert workers are capable of reliably identifying many of the answers that are incorrect, and (2) when given the opportunity, crowd workers can accurately identify the limits of their knowledge.

Based on these observations, we developed a three phase workflow: (1) workers filter the set of options, reducing the complexity of the task, (2) workers select an answer or say they are unsure, and (3) difficult cases that workers disagreed on or were unsure of are decided by experts. The experts choose from the complete, unfiltered set of options.

To measure the effectiveness of the approach we ran experiments at two scales. First, using 200 examples, we measured the effectiveness of each phase in the process and ran a comparison of end-to-end performance against other workflows. Second, using a larger set of 2,014 examples, we verified the end-to-end performance of our approach, showing that it achieves high accuracy while requiring experts for only 13% of cases.

Our work shows that with careful workflow design, crowd workers can effectively contribute to annotation of complex tasks such as semantic role labeling. The key ideas of crowd filtering and a mechanism for expressing uncertainty could be used in other NLP annotation tasks to enable the creation of larger, more sophisticated resources.

## 2 Related Work

A range of previous studies have explored methods of crowdsourcing SRL. Most work has focused on crowd-only workflows, with comparatively low ac-

curacy or extensive worker training (Fossati et al., 2013; Feizabadi and Padó, 2014; Chang et al., 2015; Dumitrache et al., 2019; Hahm et al., 2020). This work guided our user interface designs and our understanding of challenges in SRL annotation. For example, we apply Dumitrache et al. (2018)'s finding that cases where workers disagree are often more subtle or ambiguous. The most relevant work, Wang et al. (2017), used a classifier to assign hard examples to experts and easy examples to crowd workers. They achieved high accuracy (95%), but required experts for 34% of cases. Their classifier is complementary to the ideas we propose.

Another approach has used question-answering to annotate SRL (He et al., 2015; FitzGerald et al., 2018). This method is effective, but does not cover all roles and tends to have low recall. Recent work has improved recall, but overall accuracy remains low, with an F-score of 82 on CoNLL-2009 data (Roit et al., 2020). Another approach used an automatic process to expand existing datasets and then used the crowd to check paraphrases (Pavlick et al., 2015). While effective, this approach is limited to expanding lexical coverage using sentences from an existing resource.

Word Sense Disambiguation (WSD) is related to the predicate sense labeling task we consider. Prior work has explored crowdsourcing for WSD, but has mostly been unable to achieve high performance (Hong and Baker, 2011; Rumshisky, 2011; Kapelner et al., 2012; Venhuizen et al., 2013; Jurgens, 2013). There has been success on combining crowdsourcing with distant supervision for relation extraction (Zhang et al., 2012; Liu et al., 2016; Abad et al., 2017). Many other semantic parsing formalisms exist, such as AMR and UCCA, but we are unaware of work on crowdsourcing for them.

More generally, a range of approaches have been proposed to increase crowdsourcing quality, including worker filtering (Li and Liu, 2015), attention checks (Oppenheimer et al., 2009), and incentives (Venhuizen et al., 2013). These are all complementary to our proposed method.

## 3 Proposed Workflow

SRL can be divided into three parts: (1) identifying predicate and argument spans, (2) labeling predicate senses, and (3) labeling argument roles. We consider the latter two.[1] We describe each labeling decision as a **task**. In predicate sense classification tasks, a predicate in a sentence is given, and the goal is to identify the sense in which it is being used. In argument role classification tasks, an argument for a predicate with a known sense is given, and the goal is to identify the argument's role relative to the predicate. For example, for "John spoke .", there are five options for the sense of "speak", and between one and four options for the argument "John" depending on the sense of "speak". In this case, the correct sense is "speak.01 (speak, lecturing, talking)" for the predicate and "A0 (talker)" for the argument.

We aim to use the crowd to annotate SRL with high accuracy. This is difficult for two reasons. First, non-expert workers lack the linguistic expertise to understand some of the more complex role labels. Second, there can be an overwhelming number of label options, with subtle differences in meaning. These issues increase the cognitive load of selection, reducing the likelihood that workers will select the true label.

In a preliminary study, we measured the accuracy of asking five workers to choose a label. The crowd only outperformed a machine prediction when they were unanimous, which occurred in 1% of cases. However, we also found that workers could reliably identify the top few most likely labels, and could almost perfectly identify the most unlikely labels.

These observations led us to design a three phase workflow for predicate and role labeling:

1. **Filter:** A task is given to $n$ workers. Each worker selects the *least* likely options, selecting at least half of them. Options selected by every worker are filtered out. All other options remain available. If there are still many options we repeat the process, gradually reducing the number of options. Tasks with exactly one option remaining are assigned that option and do not go to the other phases.

2. **Select:** Tasks with two or more options remaining are given to a new set of $n$ workers, who are asked to select one of these options as the correct answer. We also provide a "not sure" option[2] to allow workers to explicitly indicate uncertainty. Tasks that (1) achieve

[1] Analysis of SRL system output indicates that label errors are the largest source of error, and automatic systems can achieve 94.5% precision and 98.5% recall on predicate detection (He et al., 2017).

[2] For argument tasks, there is one more option "none of the above", to cover situations where the automatic system assigns an argument to an incorrect predicate.

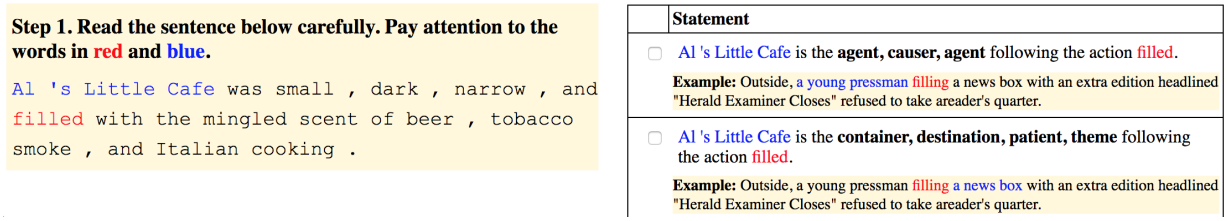


Figure 1: Part of the user interface for argument role identification in the Filter phase.

majority agreement on an answer and (2) do not have a single "not sure", are assigned the answer and do not go to the final phase.

3. **Expert:** Tasks that are not resolved in the first two phases are sent to experts. The interface presents the complete set of initial options, ranked as follows: (1) the automatic system's choice, (2) the highest voted choice in the Select phase, (3) other options chosen in the Filter phase, (4) all remaining options.

This workflow addresses the two key challenges described above. First, consider the challenge that workers lack expert knowledge. The Select phase separates out difficult cases by requiring majority agreement and no uncertainty. These difficult cases are then decided by experts with the necessary knowledge. Second, consider the challenge that there can be an overwhelming number of options. The Filter phase reduces the complexity of the task, focusing attention on likely options. This assumes that our filtering process removes unlikely options without removing the correct ones, which we verify experimentally in Section 5.1.

**Comparison Approaches** In our experiments, we compare with three other data annotation methods. *Automatic* uses the output of a statistical model (Akbik and Li, 2016), with no human input. *Review-Select* uses a two phase process. First, five workers review the system prediction. If any worker marks the prediction as incorrect, another set of workers choose an answer and we assign the most common choice. *Review-Expert* uses the same review process as the previous approach, but an expert chooses the answer rather than the crowd.

## 4 Experimental Setup

We consider experiments on two sets of data, both from the English portion of the CoNLL-2009 shared task (Hajič et al., 2009). We use one set of 200 randomly chosen tasks (drawn from the training data) to evaluate components of our approach. We use a second set of 2,014 randomly chosen tasks to evaluate our workflow end-to-end. There are 459 predicates and 1555 arguments, covering 300 sentences from the CoNLL test set. We did not include cases where there is only one frame for the predicate in Propbank as there is no decision to be made. We evaluate against the expert-annotated shared task data, with edits based on errors we found in 39 cases.

We recruited crowd workers from Amazon Mechanical Turk via LegionTools (Lasecki et al., 2014; Gordon et al., 2015), and paid them US minimum wage ($7.25/hr). In all conditions, workers received two tutorial tasks with feedback before working on ten tasks. Workers were randomly and independently assigned to tasks. $n$ is five for both the Filter phase and the Select phase.

The predicate word and argument spans are automatically identified using the Akbik and Li (2016) system. We present the workers with spans by projecting the head-word, as we expected spans to be more intuitive for workers. The sense inventory and argument types are as defined in Propbank. For argument labeling the sense of the predicate is the one produced by our workflow. If the span is incorrect, we expect workers would make a best effort to interpret the span (for example, if the span is one word too long or short they will probably still understand it correctly, especially since they see it in the context of the entire sentence). However, for evaluation, we label these cases with a special category, 'none', indicating that the span is incorrect or attached to the incorrect predicate.

To confirm the consistency of our expert annotator, we had a second expert independently perform the annotations. The Cohen's Kappa score between the two experts was 0.92 for predicates and 0.85 for arguments, near-perfect agreement (Altman, 1990).

### 4.1 Selecting the Filter Threshold

The Filter phase repeats until the number of options for a task is below a pre-defined threshold.

| | All Tasks | | Tasks with 4+ options | |
|---|---|---|---|---|
| Round | Average Number of Options | Cumulative Gold Lost | Count | Average Number of Options |
| 0 | 4.83 | 0 | 76 | 9.07 |
| 1 | 2.84 | 1 | 45 | 6.69 |
| 2 | 2.27 | 1 | 25 | 5.88 |
| 3 | 2.05 | 2 | 15 | 5.27 |
| 4 | 1.91 | 3 | 6 | 4.67 |
| 5 | 1.87 | 4 | 2 | 4.00 |
| 6 | 1.85 | 4 | 0 | – |

Table 1: Results of iterative filtering for 200 tasks. After six rounds, the gold answer has been lost in only four cases (2%), and even then it can be recovered if the task goes to the expert phase. Meanwhile, the average number of options has been dramatically reduced.

To choose the threshold, we performed an experiment in which we simulated the Filter phase and measured the accuracy of workers in the Select phase. The test involved ten predicate and ten argument tasks. We varied the number of options in each task, always keeping the true answer. We asked five workers to select the right answer and measured the accuracy of the majority choice.

With two options they were perfect, with three options they scored 0.95, and with four they scored 0.80. This confirms our preliminary observation that workers are more accurate when there are fewer options. For the rest of the experiments, we set the filter threshold to three.

## 5 Results

### 5.1 Phase Evaluation

These experiments evaluate the components of our system on a set of 200 tasks.

**Filtering effectively reduces the number of irrelevant options** Table 1 shows results over multiple rounds of filtering. As the fourth column shows, after each round there are 40% fewer tasks with 4+ options. After six rounds of filtering, all tasks have three or fewer options and only 2% of tasks have had the true answer removed. Even in those cases, if the next step (Select) does not produce an answer then the expert will be able to assign the true answer since they choose from the unfiltered set of options.

**Most tasks finish early in the workflow with high accuracy** Table 2 shows for each phase how many tasks are complete after that phase and the accuracy on those tasks. Frequently, the filter phase reduces the options down to a single correct answer. In tasks that proceed to the Select phase, we see that the number of options has been sufficiently reduced to enable high accuracy. Finally, the number of tasks that proceed to the final phase and require experts is relatively small.

| | Cumulative Finished | | Cumulative Accuracy | | This Phase Accuracy | |
|---|---|---|---|---|---|---|
| Phase | P | A | P | A | P | A |
| Filter | 38% | 13% | 0.99 | 1.00 | 0.99 | 1.00 |
| Select | 87% | 85% | 0.94 | 0.97 | 0.90 | 0.96 |
| Expert | 100% | 100% | 0.94 | 0.97 | 0.92 | 0.97 |

Table 2: Tasks finished after each phase and their accuracy for Predicates (P) and Arguments (A).

| | Accuracy | | Experts | | Crowd |
|---|---|---|---|---|---|
| Workflow | P | A | P | A | Cost |
| 200 tasks | | | | | |
| Automatic | 0.87 | 0.89 | 0 | 0 | 0 |
| Review-Select | 0.83 | 0.82 | 0 | 0 | $39 |
| Review-Expert | 0.94 | 0.97 | 55% | 58% | $30 |
| Our Workflow | 0.94 | 0.97 | 13% | 15% | $103 |
| 2,014 tasks | | | | | |
| Our Workflow | 0.95 | 0.93 | 12% | 12% | - |

Table 3: Comparison of workflows for annotation of predicates (P) and arguments (A). Our proposed workflow trades off expensive expert work for cheaper crowd work while maintaining high accuracy.

### 5.2 End-to-End Comparison

This experiment aims to compare our overall approach with other options in terms of accuracy and expert workload. Table 3 shows an end-to-end comparison of output quality between several different workflows. The final row of the table shows the results of a scaled up version of the experiment, with 2,014 tasks.

**Our approach uses substantially less expert input** If expert effort is fixed (e.g. the amount of time a research team has for annotation), then our approach allows 4x as much data to be annotated as Review-Expert. If the annotation budget is fixed, then the balance depends on the cost of experts and the speed at which they work. Assuming even low expert pay, our approach comes out ahead, as we trade expensive expert effort for cheap crowd effort (decreasing expert effort by 4x while increasing crowd effort by 3.4x).

Table 4 shows the distribution of argument labels overall and for cases that are decided by experts in our workflow. They generally follow the same

| Label | Total Count | Total Percentage | Sent to Experts Count | Sent to Experts Percentage |
|---|---|---|---|---|
| A1 | 611 | 39.3 | 65 | 33.5 |
| A0 | 378 | 24.3 | 34 | 17.5 |
| A2 | 121 | 7.8 | 15 | 7.7 |
| AM-TMP | 116 | 7.5 | 19 | 9.8 |
| AM-MOD | 68 | 4.4 | 10 | 5.2 |
| AM-MNR | 47 | 3.0 | 8 | 4.1 |
| none | 42 | 2.7 | 17 | 8.8 |
| AM-LOC | 39 | 2.5 | 9 | 4.6 |
| AM-NEG | 38 | 2.4 | 2 | 1.0 |
| AM-DIS | 37 | 2.4 | 7 | 3.6 |
| A3 | 19 | 1.2 | 3 | 1.5 |
| AM-PNC | 16 | 1.0 | 2 | 1.0 |
| AM-DIR | 13 | 0.8 | 2 | 1.0 |
| A4 | 10 | 0.6 | 1 | 0.5 |

Table 4: The distribution of labels in the end-to-end experiment overall and for cases that go to the expert. 'none' applies to cases where the predicted argument span is incorrect or attached to the incorrect predicate.

| | Gold | | | | | | |
|---|---|---|---|---|---|---|---|
| Anno | 0 | 1 | 2 | TMP | LOC | none | other |
| 0 | 369 | 16 | 2 | 1 | 1 | 5 | 1 |
| 1 | 5 | 589 | 7 | 3 | - | 12 | 2 |
| 2 | 1 | 4 | 104 | - | 2 | 4 | 2 |
| TMP | - | - | | 104 | - | 2 | 2 |
| LOC | - | 1 | 3 | - | 34 | - | 4 |
| none | - | - | | - | - | 14 | 1 |
| other | 3 | 1 | 5 | 2 | 2 | 5 | 232 / 10 |

Table 5: Confusion matrix of annotated and gold argument labels on the end-to-end data with our workflow. The other-other cell shows (matching / not matching).

trend, with core arguments (A0, A1, A2) dominating in both cases. One exception is the cases where the argument span is incorrect (none), which go to experts much more frequently. This is a positive result, as the expert may then be able to address the span error (though we did not consider this possibility in our experiments).

**Our approach maintains high accuracy** The agreement between our approach and the gold standard is comparable to expert agreement, which was 94% on predicates and 95% on arguments for Propbank before adjudication (Palmer et al., 2005). To further understand the errors, we compared them with errors made by the automatic system. We avoid 67% of the errors the automatic system makes, but do introduce errors in 1.7% of the cases it gets right. Overall, this means there is a 62.5% relative error reduction between the automatic system and our crowd workflow. Note that this is also the ideal scenario for the automatic model, as there is a close match with the training domain (also CoNLL data). Akbik and Li (2016) found precision and recall both dropped 10+ points when evaluating systems out-of-domain. As a final test, we trained an SRL system using our annotations and found no significant shift in results, which is unsurprising, given that our annotations are almost identical to the reference. Table 5 shows a confusion matrix comparing our annotations and the gold annotations. No particular type of confusion dominates the 109 argument errors.

**We identify errors in the gold standard CoNLL data** In the process of our experiments, 35 predicate tasks and 34 argument tasks had answers with unanimous agreement that did not match the CoNLL 2009 gold standard. We sent these to three experts for re-evaluation and 51% of our predicates and 62% of our arguments were actually correct. This highlights the effectiveness of this method.

## 6 Conclusion

We propose a filtering process that can simplify complex selection tasks that arise in SRL annotation. Evaluating on 2,014 examples, we find that our workflow matches gold-standard data for 95% of predicates and 93% of arguments, with expert input for only 13% of cases. More broadly, our approach expands the applicability of crowdsourcing, enabling the creation of larger, more complex, high quality resources.

## Acknowledgements

We would like to thank Laura Burdick for helpful feedback on earlier drafts of this paper and the anonymous reviewers for their helpful suggestions. This material is based in part on work supported by IBM as part of the Sapphire Project at the University of Michigan, a DARPA Young Faculty Award grant number D19AP00079, and a Bloomberg Data Science Research Grant. Any opinions, findings, conclusions or recommendations expressed above do not necessarily reflect the views of IBM.

## References

Azad Abad, Moin Nabi, and Alessandro Moschitti. 2017. Self-crowdsourcing training for relation extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 518–523.

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: Technical Papers*, pages 599–608.

Douglas G Altman. 1990. *Practical statistics for medical research*. CRC press.

Nancy Chang, Praveen Paritosh, David Huynh, and Collin Baker. 2015. Scaling semantic frame annotation. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 1–10.

Anca Dumitrache, Lora Aroyo, and Chris Welty. 2018. Capturing ambiguity in crowdsourcing frame disambiguation. In *The sixth AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, pages 12–20.

Anca Dumitrache, Lora Aroyo, and Chris Welty. 2019. A crowdsourced frame disambiguation corpus with ambiguity. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2164–2170.

Parvin Sadat Feizabadi and Sebastian Padó. 2014. Crowdsourcing annotation of non-local semantic roles. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 226–230.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-scale QA-SRL parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060.

Marco Fossati, Claudio Giuliano, and Sara Tonelli. 2013. Outsourcing FrameNet to the crowd. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 742–747.

Mitchell Gordon, Jeffrey P Bigham, and Walter S Lasecki. 2015. Legiontools: a toolkit+ ui for recruiting and routing crowds to synchronous real-time tasks. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 81–82.

Younggyun Hahm, Youngbin Noh, Ji Yoon Han, Tae Hwan Oh, Hyonsu Choe, Hansaem Kim, and Key-Sun Choi. 2020. Crowdsourcing in the development of a multilingual FrameNet: A case study of Korean FrameNet. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 236–244.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

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

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.

David Jurgens. 2013. Embracing ambiguity: A comparison of annotation methodologies for crowdsourcing word sense labels. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–562.

Adam Kapelner, Krishna Kaliannan, H. Andrew Schwartz, Lyle Ungar, and Dean Foster. 2012. New insights from coarse word sense disambiguation in the crowd. In *Proceedings of COLING 2012: Posters*, pages 539–548.

Walter S Lasecki, Mitchell Gordon, Danai Koutra, Malte F Jung, Steven P Dow, and Jeffrey P Bigham. 2014. Glance: Rapidly coding behavioral video with the crowd. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 551–562.

Hongwei Li and Qiang Liu. 2015. Cheaper and better: Selecting good workers for crowdsourcing. In *Third AAAI Conference on Human Computation and Crowdsourcing*, pages 20–21.

Angli Liu, Stephen Soderland, Jonathan Bragg, Christopher H. Lin, Xiao Ling, and Daniel S. Weld. 2016. Effective crowd annotation for relation extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 897–906.

Daniel M. Oppenheimer, Tom Meyvis, and Nicolas Davidenko. 2009. Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology*, 45(4):867 – 872.

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

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. 2015. FrameNet+: Fast paraphrastic tripling of FrameNet. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 408–413.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2020. Controlled crowdsourcing for high-quality QA-SRL annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7008–7013.

Anna Rumshisky. 2011. Crowdsourcing word sense definition. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 74–81.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Noortje J. Venhuizen, Valerio Basile, Kilian Evang, and Johan Bos. 2013. Gamification for word sense labeling. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 397–403.

Chenguang Wang, Alan Akbik, Laura Chiticariu, Yunyao Li, Fei Xia, and Anbang Xu. 2017. CROWD-IN-THE-LOOP: A hybrid approach for annotating semantic roles. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1913–1922.

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 (Volume 1: Long Papers)*, pages 825–834.