It is better to Verify: Semi-Supervised Learning with a human in the loop for large-scale NLU models

Verena Weber, Enrico Piovano and **Melanie Bradford**
Amazon Alexa AI, Berlin, Germany
{wverena,piovano,neunerm}@amazon.com

Abstract

When a NLU model is updated, new utterances must be annotated to be included for training. However, manual annotation is very costly. We evaluate a semi-supervised learning workflow with a human in the loop in a production environment. The previous NLU model predicts the annotation of the new utterances, a human then reviews the predicted annotation. Only when the NLU prediction is assessed as incorrect the utterance is sent for human annotation. Experimental results show that the proposed workflow boosts the performance of the NLU model while significantly reducing the annotation volume. Specifically, in our setup, we see improvements of up to 14.16% for a recall-based metric and up to 9.57% for a F1-score based metric, while reducing the annotation volume by 97% and overall cost by 60% for each iteration.

1 Introduction

Natural Language Understanding (NLU) models are a key component of task-oriented dialog systems such as as Amazon Alexa or Google Assistant which have gained more popularity in recent years. To improve their performance and extend their functionalities, new versions of the NLU model are released to customers on a regular basis. In the classical supervised learning approach, new training data between model updates is acquired by sampling utterances from live traffic and have them annotated by humans. The main drawback is the high cost of manual annotation. We refer to this conventional workflow as *human annotation* workflow. In this paper, we propose a new workflow with the aim to reduce the annotation cost while still maintaining high quality NLU models. We refer to it as the *human verification* workflow. The proposed workflow uses the previous (current) version of the NLU model to annotate the new training data before each model update. The predicted annotation produced by the NLU model, which we refer to as NLU hypothesis or interpretation, is then reviewed by humans. If the NLU hypothesis is assessed as correct, the NLU hypothesis is used as the ground-truth annotation of the utterance during training. If the NLU hypothesis is assessed as incorrect, the utterance is sent for human annotation before being ingested for training. With the proposed workflow, only utterances for which the hypothesis of the NLU model was assessed as incorrect are annotated by humans, thereby reducing the annotation volume drastically. Since verifying is faster and cheaper than annotating, a cost reduction is achieved. We investigate the adoption of this workflow once the system has reached a certain maturity, not from the start. While these two workflows would provide the same annotation for any utterance in an ideal world, the results may differ in the real world depending on the presence of annotation or verification errors. In this paper, we would like to answer the following fundamental question: in terms of human annotation errors, human verification errors and model performance, is it better to manually verify or annotate in order to iteratively update NLU systems?

To answer this question, we investigate the impact of human annotation vs. verification in a large scale NLU system. To this end, we consider two model architectures utilized for NLU models in the current production systems, a Conditional Random Field (CRF) (Lafferty et al., 2001; Okazaki, 2007) for slot filling and a Maximum Entropy (MaxEnt) classifier (Berger et al., 1996) for intent classification as well as a transformer based BERT architecture (Devlin et al., 2018). We evaluate the proposed workflow both explicitly by measuring annotation quality as well as implicitly by comparing the resulting model performance. Our experimental results show that the *human verification* workflow boosts the model performance while reducing human annotation volumes. In addition, we show that human annotation resources are better spent on utterances selected through Active Learning (Cohn et al., 1996; Settles, 2009; Konyushkova et al., 2017).

Proceedings of the 2nd Workshop on Data Science with Human in the Loop: Language Advances, pages 8–15
June 11, 2021. ©2021 Association for Computational Linguistics

2 Related Work

Using a model to label data instead of humans is an approach that has been studied extensively since human labelling is costly while unlabelled data can be acquired easily. Under the term Semi-supervised learning (SSL) (Zhou and Belkin, 2014; Zhu, 2005) many different approaches to leverage unlabelled data emerged in the literature. SSL aims at exploiting unlabelled data based on a small set of labelled data. One approach is self-training, also referred to as self-teaching or bootstrapping (Zhu, 2005; Triguero et al., 2015). In self-training labels are generated by feeding the unlabelled data in a model trained on the the available labelled data. Typically, the predicted labels for instances with high confidence are then used to retrain the model and the procedure is repeated. For neural networks, Lee (2013) suggested pseudo-labelling which optimizes a combination of supervised and unsupervised loss instead of retraining the model on pseudo-labels. Self-training has been applied to several natural language processing tasks. To name only a few examples, Yarowsky (1995) uses self-training for word sense disambiguation, Riloff et al. (2003) to identify subjective nouns. In McClosky et al. (2006) self learning is used for parsing.

The two main drawbacks of self-training are that instances with low confidence scores cannot be labelled and that prediction errors with high confidence can reinforce itself. To mitigate the latter issue strategies to identify mis-labeled instances have been discussed. An exhaustive review is beyond the scope of this paper, we just name a few examples. Li and Zhou (2005) use local information in a neighborhood graph to identify unreliable labels, Shi et al. (2018) add a distance based uncertainty weight for each sample and propose Min-Max features for better between-class separability and within-class compactness.

In this paper we suggest to use human verification to ensure the ingested predicted labels are reliable. In addition, we rely on human annotation for those utterances that the model cannot interpret correctly. The goal is to mitigate the two afore-mentioned problems of self-training.

A so called human-in-the-loop approach has been investigated for different applications. Zhang et al. (2020) investigate a human-in-the-loop approach for image segmentation and annotation. Schulz et al. (2019) examine the use of suggestion models to support human experts with segmentation and classification of epistemic activities in diagnostic reasoning texts. Zhang and Chaudhuri (2015) suggest active learning from weak and strong labelers where these labelers can be humans with different levels of expertise in the labelling task. Shivaswamy and Joachims (2015) show that a human expert is not always needed but that user behavior is valuable feedback that can be collected more easily.

The contribution of this paper is two-fold: First, we propose a SSL approach with a human in the loop for large-scale NLU models. Second, we show this workflow boosts the performance in a production system while reducing human annotation significantly.

Active Learning (AL) (Cohn et al., 1996; Settles, 2009; Konyushkova et al., 2017) proposes to label those instances that promise the highest learning effect for the model instead of blindly labelling data. Since the proposed workflow reduces the human annotation volume, we spend some of these freed up resources on annotation of AL data.

3 Setup and Approach

In this section, we briefly discuss the NLU model, the used metrics, the concept of iterative model updates and evaluation.

3.1 NLU task

A common approach to NLU is dividing the recognition task into two subtasks. Predicting the intent and the slots of a user's utterance constitutes a way to map the utterance on a semantic space. Accordingly, our NLU model consists of two models, each performing one of these subtasks. Intent classification (IC) predicts the user's specific intent, e.g. play music or turn on a light. Slot filling (SF), finally extracts the semantic constituents from the utterance. Taking the example "Where is MCO?" from the ATIS data (Tur et al., 2010) (Do and Gaspers, 2019), should be labelled as $where-[O]\ is-[O]\ MCO-[B-airport\_code]$ by slot filling. The intent should be recognized as city. When an utterance is humanly annotated for training, the annotator performs the same operation of the NLU model by mapping the utterance to a specific intent and slots in order to be ingested for training.

3.2 Metrics

We report results considering two metrics utilized to evaluate the performance of NLU models in production systems, Semantic Error Rate (SemER) and Intent Classification Error rate (ICER). SemER takes into consideration both intent and slot classification errors, while ICER only takes intent errors into consideration. SemER is computed as follows:

$$SemER = \frac{\#(slot \ + \ intent \ errors)}{\#(slots \ + \ intents \ in \ reference)} \quad (1)$$

ICER simply is the percentage of utterances with mis-classified intent, only intent classification counts while slot errors are ignored.

$$ICER = \frac{\#(intent \ errors)}{\#(total \ utterances)} \quad (2)$$

Note that both SemER and ICER are error metrics, i.e. a metric reduction reflects an improvement. Both are one-sided metrics that do not take precision into account. Therefore, we also report F-metrics for SemER and ICER, which are referred to as F-SemER and F-ICER, respectively. They are defined as the harmonic mean of the recall-based metric and the precision. We report macro-averages over intents for all metrics.

3.3 Iterative Model Updates

NLU models need to be regularly updated to improve their capability to understand new customer requests and extend the functionalities of the virtual assistant. Therefore new models trained on recent customer data are released on a regular basis. New data is sampled from live traffic between two NLU model releases and annotated. A part of the legacy training data is then discarded and replaced by the new annotated data for two reasons: 1) practical constraints to the building time of the new release model, 2) using too old and therefore unrepresentative data could degrade model performance. As a consequence, each NLU model is trained on an almost constant number of training utterances. For example, assuming that the overall training size is constrained to 400.000 utterances, then, if in a new release 10.000 new utterances are added, the oldest 10.000 will be removed.

3.4 Maturity and workflow evaluation

When NLU models are released for the first time, only human annotated data are used for training as previous versions of the NLU model are not available. This means in theory, the two workflows can be implemented from the second release onward. This implies that during the first few releases the majority of the training data is human annotated data. However, due to the data elimination procedure described in Section 3.3, after a certain number of releases with the verification workflow the manually annotated data from the first release will be fully removed from the training set. Here we assume to be in that maturity stage, where the full training dataset is derived from either the verification or annotation workflow, and hence no mixed training set between the two workflows is considered. For evaluation of the proposed workflows, we simulate the described updates and consider a specific model update for evaluation. A schematic timeline is shown in Figure 1. As we are considering a mature NLU model, this evaluation is representative of other model updates.

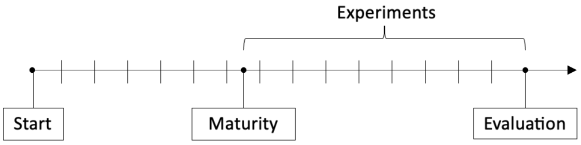

Figure 1: Schematic depiction of the NLU model updates timeline. Each dash represents a release. Results are reported for Evaluation point.

4 Proposed workflow

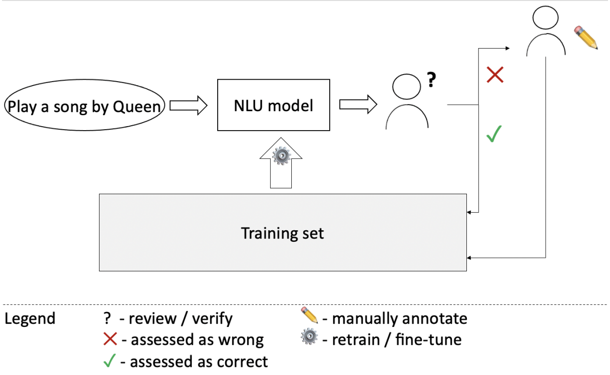

Figure 2: Schematic depiction of the proposed verification workflow. Note that the NLU model is updated periodically.

4.1 Detailed Workflow Description

This section describes the two workflows in detail. Throughout this paper we denote the *human annotation* workflow as the benchmark.

1. **Human annotation workflow** - *benchmark*: In each model update, the new training utterances are sent for manual annotation. Hence, the whole training dataset on which the NLU model is retrained (or-fine-tuned) periodically is human annotated, including the recently added utterances. The annotator only has access to the annotation guideline, but cannot see any kind of hypothesized annotation of the utterance.

2. **Human verification workflow** - *proposed*: Before each model update, the new training instances are first fed into the previous NLU model. The NLU hypothesis is then sent for human verification to assess if the NLU hypothesis is correct or not. If the annotation is evaluated as correct, the NLU hypothesis is ingested as ground-truth in the new NLU model training dataset. If the annotation is evaluated as incorrect, the utterance is sent for human annotation before being ingested. In this workflow, the evaluator has access to both the annotation guideline as well as the NLU annotation hypothesis of the utterance. Figure 2 depicts the proposed workflow. The training dataset on which the NLU model is retrained (or fine-tuned) only partially consists of human-annotated data.

With the proposed workflow, the cost is dramatically reduced as verifying is faster and cheaper than annotating. However, the question is if the verification workflow is also favorable in terms of data quality and model performance. In our experiments we therefore evaluate which of the two workflows is able to generate higher quality training data and enhance the NLU model performance. Results are discussed in Section 7.

5 Datasets

For training, we start with a dataset of unlabelled utterances representative of the user engagement with a dialog system. The dataset spans over a large number of intent and slots representative of multiple functionalities. High level statistics are listed in Table 5.

In order to have the same annotation and verification quality as in the production system, we requested the support from professional annotators. Trained and experienced annotators mimicked both workflows. For each utterance, one annotator of the team followed the *human annotation* workflow, while another followed the *human verification* workflow. For each training utterance, we also have the corresponding NLU hypothesis from the production model when the utterance was sampled. As a result two labelled training datasets were generated from one unlabelled dataset following each workflow. The overall training dataset has been built over multiple NLU releases as explained in Section 3.3.
The two training sets are then used to re-train or fine-tune each of the considered architecture. As a test set, we also consider a dataset of utterances representative of the engagement of the users with a voice assistant (see Table 5), also sampled as explained in Section 3.3. In order to have a correct and unbiased test set, test data are annotated following a different pipeline than the ones for training. For each test utterances three annotators need to produce the same annotation (100% agreement). This allows us to assume that the annotation of the test data is almost surely correct. The updated models are then evaluated on the test set to compare performance.

6 Experiments

This section describes the conducted experiments to evaluate both workflows and provides more details about how we selected utterances for annotation through AL.

6.1 Considered Model Architectures

To evaluate the proposed verification workflow, we consider two NLU architectures:

- CRF+MaxEnt classifier architecture:
  We use a Conditional Random Field (CRF) (Lafferty et al., 2001; Okazaki, 2007) for slot filling and a Maximum Entropy (MaxEnt) classifier (Berger et al., 1996) for intent classification. The new NLU model is obtained by re-training from scratch on the updated training dataset.

- BERT architecture:
  We use a transformer based BERT model (Devlin et al., 2018) that jointly solves the task of intent classification and NER. Hidden states are fed into a softmax layer to solve the two tasks. We use pre-trained mono-lingual BERT for German trained on unsupervised data from

	# utterances	# distinct intents	# distinct slots
training set	400 000	316	282
test set	100 000	316	282

Table 1: High level statistics for training and test set.

Wikipedia pages. We tokenize the input sentence, feed it to BERT, get the last layer's activations, and pass them through a final layer to make intent and NER predictions. In this case the updated NLU model is obtained by fine-tuning the initial NLU model on the new training dataset.

For both approaches we keep the set of features, hyperparameters and configuration constant for our experiments. All experiments are conducted for German. For each architecture, the models are trained by using the annotated data from the annotation vs verification workflows, respectively. For the BERT models, this step is preceeded by pretraining both models on unsupervised Wikipedia data. We then compare the performance of the resulting models.

6.2 Active Learning

We perform AL in two steps considering a corpus of millions of unlabelled utterances initially:

1. For each domain, select through a binary classifier which utterances from the unsupervised corpus are relevant to the domain.

2. Out of the candidate pool, select those with the lowest confidence score product of MaxEnt classifier (IC) and CRF (NER) and send them for annotation.

Note that a low product of IC and NER score indicates that the utterance is difficult to label for the model. We selected a total of 30.000 utterances through AL for human annotation.

7 Results

This section discusses all obtained results. We first evaluate the annotation quality for both workflows and quantify the possible cost reduction for the proposed workflow, see Sections 7.1 and 7.2. Second, we compare the performance of the NLU models when trained on data labeled through the respective workflow. Results are shown in Section 7.3.

7.1 Annotation reduction with the proposed workflow

To investigate by how much human annotation could be reduced through the proposed workflow, we calculate the percentage of utterances for which the NLU hypothesis of the previous model was assessed as correct between each update. We find that 97% of the annotation from the NLU model are assessed as correct. This means that only 3% of the utterances would be manually annotated constituting a significant reduction in annotation volume. Annotating an utterance takes about 2.5 the time of verification. Note that time is proportional to cost as we assume that human annotation specialists are paid a certain wage per hour and are able to process a certain amount of utterances depending on the task, annotation vs. verification. Let N denote the number of sampled utterances, t_A the annotation time per utterance and t_V the verification time per utterance in minutes. Then $t_A = 2.5 \cdot t_V$ or $t_V = 0.4 \cdot t_A$. The total cost for the verification workflow can then be written as:

$$total_V = N \cdot t_V + 0.03 \cdot N \cdot t_A \qquad (3)$$

Substituting $t_V = 0.4 \cdot t_A$ into 3 gives $total_V = 0.43 \cdot N \cdot t_A$. Note that $N \cdot t_A$ denotes the total cost of the annotation workflow $total_A$, so

$$total_V = 0.43 \cdot total_A. \qquad (4)$$

Thus the verification workflow leads to an overall cost reduction of almost 60 %.

7.2 Quality of evaluation vs annotation

To compare the frequency of human errors in *annotation* and *verification workflow*, we requested an assessment by specialized annotators for the annotations from each workflow for one sample of utterances. For each utterance, three specialists had to agree in their assessment. Note that we took a sample of utterances assessed as correct in the *human verification workflow* as we wanted to estimate the percentage of incorrect training data that might be ingested through the verification workflow.

Table 3 shows the human errors in the *verification workflow* relative to the *annotation workflow*.

		ICER	SemER	F-ICER	F-SemER	Annotation Reduction
1	MaxEnt+CRF	-3.85%	-2.62%	-6.81%	-3.45%	-97%
2	MaxEnt+CRF+AL	-24.58%	-20.44%	-26.04%	-17.26%	-90%
3	BERT	-14.16%	-8.77%	-9.47%	-1.80%	-97%

Table 2: Rel. difference in error metrics for *verification* vs *annotation* (baseline) workflow for all experiments.

An annotation or verification is treated as incorrect if the intent or at least one of the slots is incorrect. We can see the *verification workflow* reduces overall human errors by 66% compared to the *annotation workflow*. Note that this large human error reduction is mostly driven by fewer intent errors, which are reduced by 80% for the *verification workflow* relative to the *annotation workflow*. Overall, the frequency of verification human errors is significantly lower than the frequency of annotation human errors. This means that looking at an already annotated utterance helps to reduce the number of low-quality training data compared to annotating an utterance from scratch, where the person has no indication.

To evaluate the annotation consistency in each training dataset generated through the respective workflow, we calculate the average entropy across each dataset on token level in Table 4. Entropy will be lower the fewer interpretations we see for the same token and the more consistent the annotation is. The entropy of the training set from the verification workflow is 5% lower than for the annotation workflow.

	Rel. human error
Intent Errors	-80%
Slot Errors	-50%
Overall Errors	-66%

Table 3: Human error frequencies for *verfication* vs *annotation* on a sample of utterances.

	Avg. entropy
annotation workflow	0.5677
verification workflow	0.5378
relative	-5.3%

Table 4: Average entropy on token level for each training dataset generated through the respective workflow.

7.3 Experiment Results

Table 2 displays all the experimental results to measure the impact of the *verification workflow* vs *annotation workflow* on model performance. Specifically, we show the relative percentage change of the metric values considering the verification workflow relative to the metric values considering the annotation workflow as a baseline. As SemER and ICER are error based metrics, a "-" means an improvement of the performance for verification compared to annotation, while "+" means a degradation.

It is evident that the *verification workflow* outperforms the *annotation workflow*, often even by a substantial margin, for all experiments and metrics while drastically reducing manual annotation volume for each iteration. This is in line with the previous observation of a lower error rate and higher consistency in the training data from the verification annotation workflow, see Section 7.2. Moreover, the gain in terms of ICER is higher than SemER for all experiments, which is driven by the greater reduction of intent errors in the verification workflow. We assume the display of the NLU hypothesis influences verifiers and results in a more consistent annotation when it comes to ambiguous utterances that have multiple valid interpretations. This again leads to more consistency in the training data by reducing the number of utterances for which the model sees two different annotations.

The gains for BERT are larger than for MaxEnt+CRF, except for F-SemER. This suggests that BERT is more sensitive to contradictory training data which is why the proposed workflow yields even higher performance gains compared to the MaxEnt+CRF architecture.

Given the high reduction in annotation volume through the proposed workflow, we used some of the freed up capacities instead to have AL data annotated. We added an additional 30.000 AL utterances for the most confused intents and slots to the training dataset of each workflow. As shown in Table 2, adding comparatively few AL data boosts model performance of the verification vs annotation models by more than 20% for almost all metrics while increasing the annotation volume by less than 10%. The great relative difference in performance for verification vs annotation suggests that AL is even more beneficial for the verification workflow.

8 Conclusion

With the aim of reducing annotation costs, we test a methodology where mature NLU models are iteratively updated by ingesting labelled data via a *human verification* instead of a *human annotation* workflow. Our findings show that the proposed verification workflow not only cuts annotation costs by almost 60 %, but it also boosts the performance of the NLU system for both considered architectures. This is in line with the annotation quality evaluation we performed, where we found that the human error rate for verification is lower than the human error rate for annotation yielding more consistent training data in the former. Our findings have an important practical implication: verifying is better than annotating for mature systems. Moreover, a fraction of the annotation savings should be utilized to annotate more impactful data, for instance AL data, which generated a large performance gain in the proposed workflow with a minimal increase in annotation volume.

Acknowledgements

The authors would like to thank the Alexa DeepNLU team for providing the pre-trained BERT model for German language.
The authors would also like to thank Tobias Falke for his valuable comments on an earlier draft of this paper.

References

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Quynh Ngoc Thi Do and Judith Gaspers. 2019. Cross-lingual transfer learning for spoken language understanding. *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. 2017. Learning active learning from data. In *Advances in Neural Information Processing Systems*, pages 4225–4235.

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, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

Ming Li and Zhi-Hua Zhou. 2005. Setred: Self-training with editing. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 611–621. Springer.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs). *URL http://www. chokkan. org/software/crfsuite*.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 25–32.

Claudia Schulz, Christian M Meyer, Jan Kiesewetter, Michael Sailer, Elisabeth Bauer, Martin R Fischer, Frank Fischer, and Iryna Gurevych. 2019. Analysis of automatic annotation suggestions for hard discourse-level tasks in expert domains. *arXiv preprint arXiv:1906.02564*.

Burr Settles. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. 2018. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Pannaga Shivaswamy and Thorsten Joachims. 2015. Coactive learning. *Journal of Artificial Intelligence Research*, 53:1–40.

Isaac Triguero, Salvador García, and Francisco Herrera. 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42(2):245–284.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop*, pages 19–24. IEEE.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Chicheng Zhang and Kamalika Chaudhuri. 2015. Active learning from weak and strong labelers. *arXiv preprint arXiv:1510.02847*.

Xiaoya Zhang, Lianjie Wang, Jin Xie, and Pengfei Zhu. 2020. Human-in-the-loop image segmentation and annotation. *Science China Information Sciences*, 63(11):1–3.

Xueyuan Zhou and Mikhail Belkin. 2014. Chapter 22 - semi-supervised learning. In Paulo S.R. Diniz, Johan A.K. Suykens, Rama Chellappa, and Sergios Theodoridis, editors, *Academic Press Library in Signal Processing: Volume 1*, volume 1 of *Academic Press Library in Signal Processing*, pages 1239 – 1269. Elsevier.

Xiaojin Jerry Zhu. 2005. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.