A Survey on Programmatic Weak Supervision

Jieyu Zhang\textsuperscript{1,}\textsuperscript{*}, Cheng-Yu Hsieh\textsuperscript{1,}\textsuperscript{*}, Yue Yu\textsuperscript{2,}\textsuperscript{*}, Chao Zhang\textsuperscript{2} and Alexander Ratner\textsuperscript{1,3}

\textsuperscript{1}University of Washington
\textsuperscript{2}Georgia Institute of Technology
\textsuperscript{3}Snorkel AI, Inc.

{\{jieyuz2, cydhsieh, ajratner\}@cs.washington.edu}
{\{yueyu, chaozhang\}@gatech.edu}

Abstract

Labeling training data has become one of the major roadblocks to using machine learning. Among various weak supervision paradigms, programmatic weak supervision (PWS) has achieved remarkable success in easing the manual labeling bottleneck by programatically synthesizing training labels from multiple potentially noisy supervision sources. This paper presents a comprehensive survey of recent advances in PWS. In particular, we give a brief introduction of the PWS learning paradigm, and review representative approaches for each component within PWS’s learning workflow. In addition, we discuss complementary learning paradigms for tackling limited labeled data scenarios and how these related approaches can be used in conjunction with PWS. Finally, we identify several critical challenges that remain under-explored in the area to hopefully inspire future research directions in the field.

1 Introduction

During the last decade, deep learning and other representation learning approaches have achieved remarkable success, largely obviating the need for manual feature engineering and achieving new state-of-the-art scores across a broad range of data types, tasks, and domains. However, they have largely done so via complex architectures that have required massive labeled training data sets. Unfortunately, manually collecting, curating, and labeling these training sets is often prohibitively time-consuming and labor-intensive. The data-hungry nature of these models has thus led to increased demand for innovative ways of collecting cheap yet substantial labeled training data sets, and in particular, labeling them.

To tackle the label scarcity bottleneck, a variety of classical approaches have seen a resurgence of interest. For instance, active learning (AL) [Settles, 2009; Ren et al., 2021] aims to select the most informative samples to train the model with a limited labeling budget. Semi-supervised learning (SSL) [Tarvainen and Valpola, 2017; Xie et al., 2020] leverages a set of unlabeled data to improve the model’s performance. Transfer learning approaches [Pan and Yang, 2009; Wilson and Cook, 2020] pre-train a model or a set of representations on a source domain to enhance the performance on a different target domain. However, these approaches still require a set of clean labeled data to achieve satisfactory performance, thus do not fully address the label scarcity bottleneck.

To truly reduce the burdens of training data annotation, practitioners have resorted to cheaper sources of labels. One classic approach is distant supervision where external knowledge bases are leveraged to obtain noisy labels [Hoffmann et al., 2011]. There are also other options, including crowdsourced labels [Yuen et al., 2011], heuristic rules [Awasthi et al., 2020], feature annotation [Mann and McCallum, 2010], and others. A natural question is: could we combine these approaches, and an even broader range of potential weak supervision inputs, in a principled and abstracted way?

The recently-proposed programmatic weak supervision (PWS) frameworks provided affirmative answer to this question [Ratner et al., 2016; Ratner et al., 2017]. Specifically, in PWS, users encode weak supervision sources, e.g., heuristics, knowledge bases, and pre-trained models, in the form of labeling functions (LFs), which are user-defined programs that each provide labels for some subset of the data, collectively generating a large set of training labels.

The labeling functions are usually noisy with varying error rates and may generate conflicting labels on certain data points. To address these issues, researchers have developed label models [Ratner et al., 2016; Ratner et al., 2019; Fu et al., 2020; Varma et al., 2019b] which aggregate the noisy votes of labeling functions to produce training labels. Then, the training labels is in turn used to train an end model for downstream tasks. These two-stage methods mainly focus on the efficiency and effectiveness of label model, while maintaining the maximal flexibility of the end model. In addition to the two-stage methods, later researchers also explored the possibility of coupling the label model and the end model in an end-to-end manner [Ren et al., 2020; Lan et al., 2020]. We refer to these one-stage methods as joint models. An overview of weak supervision pipeline can be found in Fig.1.

In addition, these LFs often have clear dependencies among them [Ratner et al., 2016] and therefore it is crucial to specify and take into consideration the appropriate de-
PWS (Prediction with Weak Supervision) is a method for training models in situations where obtaining labeled data is difficult. The core of PWS are the labeling functions (LFs) that provide potentially noisy weak labels that fuel the entire learning pipeline. In this section, we provide an overview over the popular types of LFs, how they are generally developed, and the potential dependency structure among the LFs.

### 3 Labeling Functions

At the core of PWS are the labeling functions (LFs) that provide potentially noisy weak labels that fuel the entire learning pipeline. In this section, we provide an overview over the popular types of LFs, how they are generally developed, and the potential dependency structure among the LFs.

#### 3.1 Labeling Function Types

In PWS, users encode different weak supervision sources into LFs, each of which noisily annotates a subset of data points. While an LF can be as general as any function \( \lambda : \mathcal{X} \rightarrow \mathcal{Y} \cup \{-1\} \) that takes as input a data point and either outputs a corresponding label or abstain, we introduce the most common types of LFs used in practice.

##### 3.1.1 User-written Heuristics

In practical applications, users generally have domain knowledge about the target learning task of interest. One common type of LF is to express the domain knowledge into heuristic labeling rules that associate corresponding labels to data points. For example, in text applications, users write keyword- or regex-based LFs that assign corresponding labels to the data points that contains the keyword or matches the specified regular expression [Ratner et al., 2017; Meng et al., 2018; Awasthi et al., 2020]. In image applications, users write LFs that provide labels to the image inputs containing specific objects, or possessing some user-specified visual/spatial properties [Varma et al., 2017; Chen et al., 2019; Fu et al., 2020].

##### 3.1.2 Existing Knowledge

**Knowledge Bases.** Oftentimes, external knowledge bases can be used to provide weak supervision over the learning task of interest, commonly known as the distant supervision approach [Hoffmann et al., 2011; Liang et al., 2020]. For example, in a relation extraction task to identify mentions of spouse relationships in news articles, [Ratner et al., 2017] writes LFs that match the text inputs against the knowledge base DBPedia\(^1\) to search for known spouse relationships.

**Pre-trained Models.** Existing pre-trained models from a related task can be used as LFs to provide weak labels. For example, in a product classification task at Google, [Bach et al., 2019] leverages existing semantic topic model to identify mentions of spouse relationships in news articles. [Zhang et al., 2022] pre-trained image classification model that has a different output label space from the target classification task is used as LFs to provide indirect weak supervision for the learning task of interest.

\(^1\)https://www.dbpedia.org/
Third-party Tools. To collect weak labels cheaply, there are several existing third-party tools available that can serve as LFs. For example, for review sentiment analysis, users can simply use TextBlob\(^2\) to assign labels for each review. Take named entity recognition (NER) as another example, there are several tagging tools such as spaCy\(^3\), NLTK\(^4\), etc, and [Lison et al., 2020] adopt them as LFs for weakly-supervised NER task. Note that, the above tools are not perfect, as the weak labels generated via their outputs contains much noise.

3.1.3 Crowd-sourced Labels.

Crowd-sourcing is the classic and well-studied approach of obtaining less accurate label annotations from non-expert contributors with lower annotation cost [Dawid and Skene, 1979; Yuen et al., 2011]. In the PWS setting, each crowd-sourcing contributor can be represented as an LF that noisily annotates the data points [Ratner et al., 2017; Lan et al., 2020]. For example, in a weather sentiment classification task, each crowd-source contributor—who grades the sentiment of tweets relating to the weather into five different categories—is considered as a LF.

3.2 Labeling Function Generation

In the PWS learning paradigm, the first and foremost step is to create a set of LFs that are used to generate the weak labels for learning the subsequent models. In practice, the LFs are typically developed by subject matter experts (SMEs) who have adequate knowledge about the task of interest. When developing LFs, in addition to leveraging existing domain knowledge, SMEs usually refer to a small subset of data points sampled from the unlabeled set, called the development set, to extract further task/dataset-specific labeling heuristics that complement the pre-existing domain knowledge [Ratner et al., 2017]. This process of LF development could sometimes be challenging and time-consuming even for domain experts. For example, it often requires SMEs to explore a considerable amount of development data to generate ideas for LFs [Varma and Ré, 2018; Galhotra et al., 2021; Boecking et al., 2021]. As a result, researchers have recently aim to reduce the amount of efforts spent in designing weak supervision sources through three main directions, namely, automatic generation, interactive generation, and guided generation of LFs.

Automatic Generation. One direction to alleviate the burden in designing LFs in PWS paradigm is to automate the process of LF development. [Varma and Ré, 2018] propose a system, Snuba, that generates LFs automatically by learning weak classification models on a small set of labeled dataset. TALLOR [Li et al., 2021a] takes as input an initial set of seed LFs that are generally simpler, and automatically learn more accurate compound LFs from multiple simple labeling rules. Similarly, GLaRA [Zhao et al., 2021] learns to augment a set of seed LFs automatically by exploiting the semantic relationships between candidate and seed LFs through a graph-based model. Notably, while we refer to this line of methods as “automatic generation” approaches, they do require a minimum amount of initial supervision, either in the form of small labeled set or seed LFs.

Interactive Generation. In contrast to fully automating the generation of LFs after given a seed supervision set, interactive generation approaches cast LF development as an interactive process where users are iteratively queried for feedback used in discovering useful LFs from a large set of candidates [Galhotra et al., 2021; Boecking et al., 2021]. Specifically, in Darwin [Galhotra et al., 2021] and IWS [Boecking et al., 2021], a set of candidate LFs is first generated based on n-grams or context-free grammar information. Then, in each iteration, the user is queried to annotate whether a presented LF, proposed by the system, is useful or not (i.e., better than random accuracy). Based on the feedback provided in each iteration, the systems learn to adapt and identify a set of high-precision LFs from the candidate set, which is used
as the final set of LFs in the PWS learning pipeline. Compared to the standard active learning approaches which relies on instance-level annotations, the interactive generation approaches are shown to achieve better performance with lower annotation costs.

Guided Generation. Based on the current workflow of LF development where SMEs write LFs by looking at a small development set of data, guided-generation approaches aim to assist the users in developing LFs by intelligently curate the development set in order to efficiently guide SMEs in exploring the data and developing informative LFs that could lead to strong resultant models [Cohen-Wang et al., 2019]. The idea resembles traditional active learning [Settles, 2009] in the sense that the goal is to strategically select data points from the unlabeled set and solicit informative supervision from the users, except that the supervision is provided at the functional-level (i.e., LFs) instead of individual label level.

4 Label Model

The multiple LFs we have for a given dataset often overlap and conflict with each other. In PWS, label model is used to integrate the LFs’ output predictions into probabilistic labels, aiming to accurately recover the unobserved ground truth labels. Till now, various label models have been proposed and most of them are based on probabilistic graphical models. It is worthwhile to note that LFs developed in practice often exhibit statistical dependency among each other [Ratner et al., 2016; Cachay et al., 2021a]. Incorporating the dependency information into the label model has been shown to be critical to the model’s ability to correctly estimate the latent ground truths [Ratner et al., 2016; Bach et al., 2017; Varma et al., 2017; Cachay et al., 2021a]. However, not all label models take into account the LF dependency structure when aggregating the LFs’ votes, where some approaches simply assume conditional independence between the LFs.

In this section, we first discuss general approaches used to incorporate LF dependency in label model. Then, we introduce more in detail different existing label models, categorized by their target learning tasks, with discussion on how the LF dependency is handled in some of the approaches.

4.1 LF Dependency Structure

Earlier work on PWS rely on users to manually specify the dependency structure among the LFs [Ratner et al., 2016]. For example, users could specify two LFs to be similar; or one LF to be fixing or reinforcing another; or two LFs are exclusive. Nevertheless, as manually specifying such dependency structure is generally hard for users, researchers have recently turned to learning or inferring the dependency structure automatically without user supervision. To automatically learn the dependency structure, [Bach et al., 2017] proposes to maximize the $\ell_2$-regularized marginal pseudo-likelihood of a factor graph with high-order dependencies and select the dependencies that have non-zero weights; [Varma et al., 2019a] exploits the sparsity of label model and leverages robust PCA technique to capture the underlying dependency structure. On the other hand, instead of learning the structure from the observed labels, [Varma et al., 2017] proposes an alternative approach that infers the relations between different LFs by statically analyzing the source code of the LFs.

Having the dependency structure on hand, whether manually specified or automatically learned/inferred, a prevailing approach to incorporate the dependency information into the label model is to embed the dependency relationships into label models, which are typically graphical models, through factor functions [Ratner et al., 2016; Shin et al., 2022] or graph structure [Ratner et al., 2019; Fu et al., 2020; Varma et al., 2019b]. In the following subsections, we introduce the label models for different learning tasks in more detail, and provide an overview of these methods in Table 1.

4.2 Label Model for Classification

For classification problems, majority voting (MV) is the most straightforward approach for aggregating different LFs, as it simply uses the consensus from the multiple LFs to obtain more reliable labels without introducing any trainable parameters. Crowdsourcing models [Dawid and Skene, 1979; Dalvi et al., 2013; Raykar et al., 2010; Khetan et al., 2018] usually leverage the expectation maximization (EM) algorithm to estimate the accuracy of each worker as well as infer the latent ground truth labels, which can also be applied here when we regard each LF as a worker. Apart from these approaches, we review several label models tailored for PWS problems. These label models are all based on probabilistic graphical model and aim to maximize the probability of observing the outputs of LFs. Specifically, they share an optimization problem as following:

$$\max_{\theta} P(L; \theta) = \sum_{Y} P(L, Y; \theta) . \quad (1)$$

The key differences among existing label models are the way they parameterize the joint distribution $P_{\theta}(L, Y)$ and how the parameters are estimated. In particular, Data programming (DP) [Ratner et al., 2016] models the distribution $P(L, Y; \theta)$ as a factor graph. It is able to describe the distribution in terms of pre-defined factor functions, which reflects the dependency of any subset of random variables and are also used to encode the dependency structure of LFs. The log-likelihood is optimized by SGD where the gradient is estimated by Gibbs sampling. MeTaL [Ratner et al., 2019], instead, models the distribution via a Markov Network and recover the parameters via a matrix completion-style approach. Later on, FlyingSquid [Fu et al., 2020] is proposed to accelerate the learning process for binary classification problems. It models the distribution as a binary Ising model, where each LF is represented by two random variables, and a Triplet Method is used to recover the parameters and therefore no learning is needed. Notably, the latter two methods encode the dependency structure of LFs into the structure of the graphical model and require label prior as input.

Additionally, researchers have attempted to extend the scope of usable LFs. CAGE [Chatterjee et al., 2020] extends the existing label models to support continuous LFs. In addition, it leverages user-provided quality for LFs to increase the training stability and making it less sensitive to initialization. Moreover, NPLM [Yu et al., 2022] enables users to utilize partial LFs that output a subset of possible class labels.
and PLRM [Zhang et al., 2022] allows the usage of indirect LFs that only predict unseen but related class; both works are built on probabilistic graphical model similar to [Ratner et al., 2016] and greatly expand the scope of usable LFs in PWS.

4.3 Label Model for Sequence Tagging

Sequence tagging problems are more complex since there are dependencies among consecutive tokens. To model such properties, Hidden Markov Models (HMM) [Baum and Petrie, 1966] have been proposed, which represent true labels as latent variables and inferring them from the independently observed noisy labels through expectation-maximization algorithm [Welch, 2003]. [Lison et al., 2020] directly apply HMM for named entity recognition task and [Safranchik et al., 2020] propose Linked-HMM to incorporate unique linking rules as an adjunct supervision source additional to general weak labels on tokens. Moreover, Conditional hidden Markov model (CHMM) [Li et al., 2021b] substitutes the constant transition and emission matrices by token-wise counterpart predicted from the BERT embeddings to model the evolve of true labels in a context-dependent manner. Another characteristics for sequence tagging problems is that the supervision can be provided at different resolutions (e.g., frame, window, and scene-level for videos). To integrate them together, Dugong [Varma et al., 2019b] has been propose to assign probabilistic labels for data with graphical models. Dugong also accelerates the inference speed with SGD based optimization techniques. Finally, as shown in [Zhang et al., 2021], label models for classification task could also be applied on sequence tagging problem with certain adaptations.

4.4 Label Model for General Learning Tasks

Very recently, UWS [Shin et al., 2022] goes beyond the traditional tasks and generalizes PWS frameworks to handle more kinds of tasks including ranking, regression, and learning in hyperbolic manifolds with an efficient method-of-moments approach in the embedding space.

5 End Model

After obtaining the probabilistic labels, the end model is used to train a discriminative model on downstream tasks. Since the probabilistic training labels derived from the label model may still contain noise, [Ratner et al., 2017] suggests using a noise-aware loss as the training objective for the end model. However, one drawback for such end models is that they are usually trained only on the data covered by weak supervision, but there may exist an ignorable portion of data that are not covered by any LFs. Motivated by this, COSINE [Yu et al., 2021a] designs a better end model by leveraging the data uncovered by LFs. Specifically, it utilizes these uncovered data in a self-training manner and generates pseudo labels for each unlabel data. Apart from the above methods, other approaches designed for learning with noisy labels [Song et al., 2020] can also be utilized as end models.

6 Joint Model

The traditional pipeline for PWS usually trains the label model and end model separately, in contrast, joint model aims to train the label model and the end model in an end-to-end manner, allowing them to enhance each other mutually. In addition, the joint model usually leverages neural network as label model instead of aforementioned statistical label model; such a design choice not only facilitates the co-training of label model and end model, but also reflects the motivation of considering data feature during the training of label model, leading to a instance-dependent label model, i.e., $P(L,Y|X; \theta)$.

As opposed to statistical label models (Sec. 4) that explicitly incorporate LF dependency through the graph structure of underlying graphical models, it is observed that neural network based joint models are able to implicitly capture the dependencies among the LFs in the learning process [Cachay et al., 2021b]. However, existing joint models generally cannot incorporate pre-given dependency structure.

6.1 Joint Model for Classification

Denoise [Ren et al., 2020] and WeaSEL [Cachay et al., 2021b] first reparameterize prior probabilistic posteriors with a neural network, then assign scores for each PWS source for aggregation. After that, the posterior network and the end model are trained simultaneously to maximize the agreement between them. [Arachie and Huang, 2019; Mazzetto et al., 2021a] both formulate the weakly supervised classification problems as a constrained min-max optimization problem, and ALL [Arachie and Huang, 2019] learns a prediction model that has the highest expected accuracy with respect to an adversarial labeling of an unlabeled dataset, where this labeling must satisfy error constraints on the weak supervision sources. Differently, AMCL [Mazzetto et al., 2021b] constructs the constraints based on the expected loss within a small set of clean data.

To denoise LFs more effectively, several methods propose to use a small number of labeled data in training. ImplyLoss [Awasthi et al., 2020] jointly train a rule denoising network based on exemplars for each label, as well as a classification model with a soft implication loss. SPEAR [Maheshwari et al., 2021] extends ImplyLoss by designing additional loss functions on both labeled and unlabeled data and encourages the consistency between the two models. In addition, ASTRA [Karamanolakis et al., 2021] adopts self-training for PWS with a teacher-student framework. The student model is initialized with a small number of labeled data and generates pseudo-labels for instances not covered by LFs, while the teacher model combines LFs with the output from the student model for the final prediction.

6.2 Joint Model for Sequence Tagging

For sequence tagging problem, Consensus Network (ConNet) [Lan et al., 2020] trains BiLSTM-CRF [Ma and Hovy, 2016] with multiple CRF layers for each labeling source individually. Then, it aggregates the CRF transitions with attention scores conditioned on the quality of LFs and outputs a unified label sequence. DWS [Parker and Yu, 2021] uses a CRF layer to capture statistical dependencies among tokens, weak labels and latent true labels. Moreover, it adopts hard EM algorithm for model training: in the E-step, it finds the
most probable labels for the given sequence; in the M-step, it maximizes the probability for the corresponding labels.

7 Complementary Approaches

In this section, we briefly describe how PWS can be connected to or combined with complementary machine learning approaches that also aim to deal with the label scarcity issue.

Active Learning. Active learning (AL) attempts to handle the label scarcity issue by interactively annotating the most informative samples to achieve good performance. As complementary approach, PWS could be utilized to improve AL. For example, [Mallinar et al., 2020] expands the initial labeled set in AL by querying labels for those which are the most relevant to existing labeled ones based on LFs, and [Nashaat et al., 2018] applied PWS to generate initial noisy training labels to improve the efficiency of a later active learning process. On the other hand, AL could in turn help PWS: [Biegel et al., 2021] asks experts to provide labels for which the label model is most likely to be mistaken, and Asterisk [Nashaat et al., 2020] employs AL to enhance the label model and proposed a selection policy based on the estimate accuracy of LFs and the output of label model.

Transfer Learning. Transfer learning (TL), which adapts a trained model to the new tasks and consequently tends to require less labeled data than training from scratch, has recently attract increasing attention, especially for the great success of fine-tuning huge pretrained models with few labels. We note that TL and PWS are orthogonal to each other and could be combined together to achieve the best performance, since TL could reduce but not eliminate the demand of labeled data, which could be offered by PWS. Indeed, current state-of-the-art PWS methods usually rely on fine-tuning pretrained models with labeled products by label model [Zhang et al., 2021].

Semi-Supervised Learning. Semi-supervised Learning (SSL) aims to train the model with a small amount of labeled data with a large amount of unlabeled data. The idea of leveraging unlabeled data to improve training has also been applied to PWS methods as [Karamanolakis et al., 2021; Yu et al., 2021] use self-training to bootstrap over unlabeled data. Moreover, [Xu et al., 2021] improves SSL by leveraging the idea of PWS; specifically, they use the labeled data to generate LFs that are in turn used to annotate the unlabeled data, finally the model is trained on the whole dataset with provided or synthesized labels. To sum up, SSL and PWS are also complementary and future works include developing more advanced methods to combine clean labels and weak labels together to further boost the performance.

8 Challenges and Future Directions

Extend to More Complex Tasks. The majority of the PWS methods only support classification or sequence tagging tasks, while there are a variety of tasks that require high-level reasoning over concepts such as question answering [Rajpurkar et al., 2016], navigation [Gupta et al., 2017] and scene graph generation [Ye and Kovashka, 2021], and curating labeled data for these tasks requires even more human efforts. Moreover, in these tasks, the input data may come from multiple modalities including text, image and tables, while the current PWS methods only consider LFs with one specific modality. Hence, it is crucial while challenging to develop multi-modal PWS methods to improve the data efficiency on these tasks.

Extend the Scope of Usable LFs. Although researchers have made attempts to extend the scope of usable LFs [Zhang et al., 2022; Yu et al., 2022], there are other sources that could potentially be used as LFs, e.g., physical rules, for more complex tasks. The ultimate goal of PWS is to leverage as more existing sources as possible to minimize human efforts in the curation of training data.

Ethical and Trustworthy AI. One of the most pressing concerns in the AI community right now is ensuring that AI techniques and models are applied ethically. Within this area of focus, one of the most important and challenging topics is ensuring that the training data which informs models is ethically labeled and managed, transparent, auditable, and bias-free. PWS approaches offer a step-change opportunity in this regard, since they result in training labels generated by code which can be inspected, audited, governed, and edited to reduce bias. However, by the same token, PWS methods can also lead to more direct bias in training data sets if used and modeled improperly [Geva et al., 2019; Lucy and Bamman, 2021]. Overall, further systematic study in this area is highly critical, and has great opportunity for improving the state of data in AI from an ethics and governance perspective.

9 Conclusion

Manual annotations are always of great importance to training machine learning models, but usually expensive and time-consuming. Programmatic weak supervision (PWS) offers a promising direction to achieve large-scale annotations with minimal human efforts. In this article, we review the PWS area by introducing existing approaches for each component inside a PWS workflow. We also describe how PWS could interact with methods from related fields for better performance on downstream applications. Then, we list existing datasets and recent applications of PWS in the literature. Finally, we discuss current challenges and future directions in the PWS area, hoping to inspire future research advances in PWS.

References

[Arachie and Huang, 2019] Chidubem Arachie and Bert Huang. Adversarial label learning. In AAAI, 2019.

[Awasthi et al., 2020] Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. Learning from rules generalizing labeled exemplars. In ICLR, 2020.

[Bach et al., 2017] Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. Learning the structure of generative models without labeled data. In ICML, 2017.

[Bach et al., 2019] Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, et al. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In SIGMOD, 2019.
[Baum and Petrie, 1966] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Stat.*, 37(6), 1966.

[Biegel et al., 2021] Samantha Biegel, Rafah El-Khatib, Luiz Otavio Vilas Boas Oliveira, Max Baak, and Nanne Aben. Active weasal: Improving weak supervision with active learning. arXiv preprint arXiv:2104.14847, 2021.

[Boecking et al., 2021] Benedikt Boecking, Willie Neiswanger, Eric Xing, and Artur Dubrawski. Interactive weak supervision: Learning useful heuristics for data labeling. In *ICLR*, 2021.

[Cachay et al., 2021a] Salva R. Cachay, Benedikt Boecking, and Artur Dubrawski. Dependency structure misspecification in multi-source weak supervision models. *ICLR Workshop on Weakly Supervised Learning*, 2021.

[Cachay et al., 2021b] Salva R. Cachay, Benedikt Boecking, and Artur Dubrawski. End-to-end weak supervision. *NeurIPS*, 2021.

[Chatterjee et al., 2020] Oishik Chatterjee, Ganesh Ramakrishnan, and Sunita Sarawagi. Robust data programming with precision-guided labeling functions. In *AAAI*, 2020.

[Chen et al., 2019] Vincent S Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. In *ICCV*, 2019.

[Cohen-Wang et al., 2019] Benjamin Cohen-Wang, Steve Mussmann, Alexander Ratner, and Christopher Ré. Interactive programmatic labeling for weak supervision. *KDD DCCL Workshop*, 2019.

[Dalvi et al., 2013] Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. Aggregating crowdsourced binary ratings. In *WWW*, 2013.

[Dawid and Skene, 1979] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society*, 1979.

[Fu et al., 2020] Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. Fast and three-rious: Speeding up weak supervision with triplet methods. In *ICML*, 2020.

[Galhotra et al., 2021] Sainyam Galhotra, Behzad Golshan, and Wang-Chiew Tan. Adaptive rule discovery for labeling text data. In *SIGMOD*, 2021.

[Geva et al., 2019] Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *EMNLP*, 2019.

[Gupta et al., 2017] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *CVPR*, 2017.

[Hoffmann et al., 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, 2011.

[Karanamolakis et al., 2021] Giannis Karonamolakis, Subhabrata Mukherjee, Guoqing Zheng, and Ahmed Hassan Awadallah. Self-training with weak supervision. In *NAACL*, 2021.

[Khetan et al., 2018] Ashish Khetan, Zachary C. Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data. In *ICLR*, 2018.

[Lan et al., 2020] Ouyu Lan, Xiao Huang, Bill Yuchen Lin, He Jiang, Liyuan Liu, and Xiang Ren. Learning to contextually aggregate multi-source supervision for sequence labeling. In *ACL*, 2020.

[Li et al., 2021a] Jiacheng Li, Haibo Ding, Jingbo Shang, Julian McAuley, and Zhe Feng. Weakly supervised named entity tagging with learnable logical rules. In *ACL*, 2021.

[Li et al., 2021b] Yinghao Li, Pranav Shetty, Lucas Liu, Chao Zhang, and Le Song. Bertifying the hidden markov model for multi-source weakly supervised named entity recognition. In *ACL*, 2021.

[Liang et al., 2020] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *KDD*, 2020.

[Lison et al., 2020] Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. Named entity recognition without labelled data: A weak supervision approach. In *ACL*, 2020.

[Lucy and Bamman, 2021] Li Lucy and David Bamman. Gender and representation bias in gpt-3 generated stories. In *NAACL Workshop on Narrative Understanding*, 2021.

[Ma and Hovy, 2016] Xueze Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, 2016.

[Maheshwari et al., 2021] Ayush Maheshwari, Oishik Chatterjee, Krishnateja Killamsetty, Ganesh Ramakrishnan, and Rishabh Iyer. Semi-supervised data programming with subset selection. In *Findings of ACL*, 2021.

[Ma and Hovy, 2020] Ma and Hovy. 2016 Xueze Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, 2016.

[Mann and McCallum, 2010] Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 2010.

[Mazzetto et al., 2021a] A. Mazzetto, C. Cousins, D. Sam, S. H. Bach, and E. Upfal. Adversarial multiclass learning under weak supervision with performance guarantees. In *ICML*, 2021.

[Mazzetto et al., 2021b] A. Mazzetto, D. Sam, A. Park, E. Upfal, and S. H. Bach. Semi-supervised aggregation of dependent weak supervision sources with performance guarantees. In *AISTATS*, 2021.

[Meng et al., 2018] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised neural text classification. In *CIKM*, 2018.
[Nashaat et al., 2018] Mona Nashaat, Aindrila Ghosh, James Miller, Shaikh Quader, Chad Marston, and J. Puget. Hybridization of active learning and data programming for labeling large industrial datasets. *IEEE Big Data*, 2018.

[Nashaat et al., 2020] Mona Nashaat, Aindrila Ghosh, James Miller, and Shaikh Quader. Asterisk: Generating large training datasets with automatic active supervision. *ACM Transactions on Data Science*, 2020.

[Pan and Yang, 2009] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 2009.

[Parker and Yu, 2021] Jerrod Parker and Shi Yu. Named entity recognition through deep representation learning and weak supervision. In *Findings of ACL*, 2021.

[Rajpurkar et al., 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[Ratner et al., 2016] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *NeurIPS*, 2016.

[Ratner et al., 2017] Alexander J Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *VLDB*, 2017.

[Ratner et al., 2019] A. J. Ratner, B. Hancock, J. Dunnmon, F. Sala, S. Pandey, and C. Ré. Training complex models with multi-task weak supervision. In *AAAI*, 2019.

[Raykar et al., 2010] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, et al. Learning from crowds. *JMLR*, 2010.

[Ren et al., 2020] Wendi Ren, Yinghao Li, Haiting Su, David Kartchner, Cassie Mitchell, and Chao Zhang. Denoising multi-source weak supervision for neural text classification. In *Findings of EMNLP*, 2020.

[Ren et al., 2021] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *CSUR*, 2021.

[Safranchik et al., 2020] Esteban Safranchik, Shiyang Luo, and Stephen Bach. Weakly supervised sequence tagging from noisy rules. In *AAAI*, volume 34, 2020.

[Settles, 2009] Burr Settles. Active learning literature survey. 2009.

[Shin et al., 2022] Changho Shin, Winfred Li, Harit Vishwakarma, Nicholas Roberts, and Frederic Sala. Universalizing weak supervision. In *ICLR*, 2022.

[Song et al., 2020] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2020.

[Tarvainen and Valpola, 2017] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

[Varma and Ré, 2018] Paroma Varma and Christopher Ré. Snuba: Automating weak supervision to label training data. In *VLDB*, volume 12, 2018.

[Varma et al., 2017] P. Varma, Bryan D. He, Payal Bajaj, Nishith Khandwala, I. Banerjee, D. Rubin, and Christopher Ré. Inferring generative model structure with static analysis. *NeurIPS*, 30, 2017.

[Varma et al., 2019a] P. Varma, Frederic Sala, Ann He, Alexander J. Ratner, and C. Ré. Learning dependency structures for weak supervision models. In *ICML*, 2019.

[Varma et al., 2019b] Paroma Varma, Frederic Sala, Shiori Sagawa, Jason Fries, Daniel Fu, Saelig Khattar, Ashwini Ramamooorthy, Ke Xiao, Kayvon Fatahalian, James Priest, and Christopher Ré. Multi-resolution weak supervision for sequential data. In *NeurIPS*, volume 32, 2019.

[Welch, 2003] Lloyd R Welch. Hidden markov models and the baum-welch algorithm. *IEEE ITS Newsletter*, 2003.

[Wilson and Cook, 2020] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *TIST*, 2020.

[Xie et al., 2020] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *NeurIPS*, 2020.

[Xu et al., 2021] Yi Xu, Jiandong Ding, Lu Zhang, and Shuigeng Zhou. Dp-ssl: Towards robust semi-supervised learning with a few labeled samples. *NeurIPS*, 2021.

[Ye and Kovashka, 2021] Keren Ye and Adriana Kovashka. Linguistic structures as weak supervision for visual scene graph generation. In *CVPR*, 2021.

[Yu et al., 2021] Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *NAACL*, 2021.

[Yu et al., 2022] Peilin Yu, Tiffany Ding, and Stephen H Bach. Learning from multiple noisy partial labelers. *AISTATS*, 2022.

[Yuen et al., 2011] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. A survey of crowdsourcing systems. In *SocialCom*, IEEE, 2011.

[Zhang et al., 2021] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. WRENCH: A comprehensive benchmark for weak supervision. In *NeurIPS*, 2021.

[Zhang et al., 2022] Jieyu Zhang, Bohan Wang, Xiangchen Song, Yujing Wang, Yaming Yang, Jing Bai, and Alexander Ratner. Creating training sets via weak indirect supervision. In *ICLR*, 2022.

[Zhao et al., 2021] Xinyan Zhao, Haibo Ding, and Zhe Feng. GLaRA: Graph-based labeling rule augmentation for weakly supervised named entity recognition. In *EACL*, 2021.