Active Deep Learning to Tune Down the Noise in Labels

Karan Samel
Astound
111 Independence Dr.
Menlo Park, CA 94024
karan@astound.ai

Xu Miao
Astound
111 Independence Dr.
Menlo Park, CA 94024
xu@astound.ai

ABSTRACT

The great success of *supervised learning* has initiated a paradigm shift from building a deterministic software system to a probabilistic artificial intelligent system throughout the industry. The historical records in *enterprise* domains can potentially bootstrap the traditional business into the modern data-driven approach almost everywhere. The introduction of the *Deep Neural Networks* (DNNs) significantly reduces the efforts of feature engineering so that *supervised learning* becomes even more automated. The last bottleneck is to ensure the data quality, particularly the label quality, because the performance of *supervised learning* is bounded by the errors present in labels. In this paper, we present a new *Active Deep Denoising* (ADD) approach that first builds a DNN noise model, and then adopts an *active learning* algorithm to identify the optimal denoising function. We prove that under the low noise condition, we only need to query the oracle with $\log n$ examples where n is the total number in the data. We apply ADD on one *enterprise* application and show that it can effectively reduce $\frac{1}{3}$ of the prediction error with only 0.1% of examples verified by the oracle.

CCS CONCEPTS

• **Information systems → Data cleaning**; • **Computing methodologies →** *Neural networks*; *Cluster analysis*;

KEYWORDS

Active Learning, Denoising, Deep Neural Networks, Classification

ACM Reference format:
Karan Samel and Xu Miao. 2018. Active Deep Learning to Tune Down the Noise in Labels. In *Proceedings of The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom, August 19–23, 2018 (KDD '18),* 10 pages.
https://doi.org/10.1145/3219819.3219914

1 INTRODUCTION

During the past decades, *supervised learning* has achieved a great success across many fields, e.g., natural language processing, computer vision, and information retrieval, with the emergence of *Big data* and *Cloud computing*. For any given task, we collect data, label data, build a model, evaluate the model, revise and repeat. This loop, as shown in Figure 1, has become the new programming paradigm to solve many real-world problems, and shifted the major focus from building a high-quality software system to building a high-quality dataset, e.g., *ImageNet* [9] for object recognition and *SQuAD* [22] for machine comprehension. However, if the collected data contains noisy and conflicting labels, the performance of the *supervised learning* is upper bounded. For example, as a popular crowd-sourcing choice, data are labeled through *Mechanical Turk* (MT) where annotators do not necessarily have domain-specific knowledge, hence bringing human errors into the dataset. Even with the majority vote mechanism, the quality of the labels is mostly not guaranteed. This is more severe in *enterprise* domains where labels come from different persons at different locations and times who follow different rules. Recent research [18] studied the effect of low-quality training data on clinical reports, and demonstrated that the difficulty of acquiring high-quality data actually bottlenecks the wide adoption of the data-driven approach in the health domain. Indeed, how can we obtain super-human accuracy from noisy inaccurate data? In this paper, we address this challenge from the *active learning* perspective.

Active learning [3, 11, 15, 19, 26, 31, 35] was proposed to effectively utilize unlabeled data with a costly oracle. When the prior distribution over the hypothesis space is known, algorithms like *Query by Commitee* [11], i.e., pick the most uncertain example to ask, can effectively query the oracle with a guaranteed generalization bound. When the prior distribution is unknown, this approach can bring significant biases to the samples, and ends with statistically inconsistent models. Research [5, 6] shows that with certain search heuristics, e.g., exploitation vs. exploration, we can avoid this bias and obtain statistical consistency. However, the label complexity remains linear with respect to *supervised learning* because we might still need to query a significant number of examples to obtain the desired accuracy. Theoretical studies [2, 7, 13] have shown that algorithms like *Agnostic Active* (A^2) can be exponentially effective under the low-noise condition, but are bound by a large VC-dimension factor. It is practically impossible to combine A^2 with a family of models like *Deep Neural Networks* in the *unsupervised learning* domain as we have to search through a huge hypothesis space to obtain a confident estimation.

Fortunately, in our setting, all data are labeled, although with a certain number of errors. These noisy labels allow us to build a model to address the noise systematically, which we use in restricting the hypothesis space. In this paper, we present a new *Deep Neural Network*-based *Active Learning* algorithm to correct noisy labels. We first build a DNN noise model to form a denoising function family. Then, we adopt the A^2 algorithm to learn the optimal

KDD '18, August 19–23, 2018, London, United Kingdom
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219914

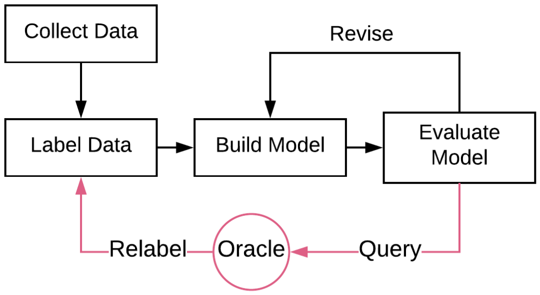


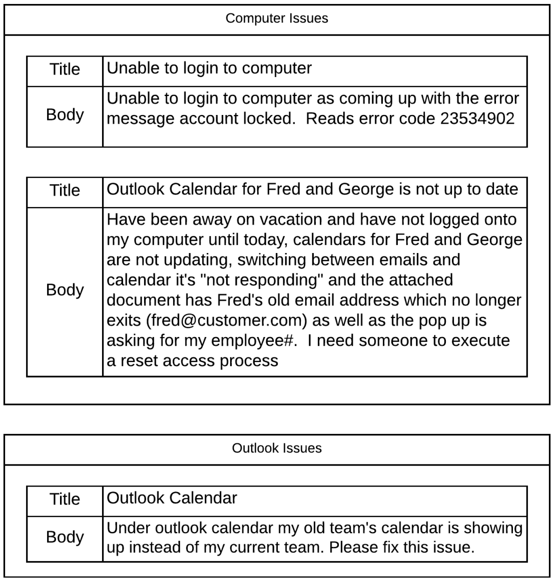
Figure 1: Data quality is the key to a data driven paradigm.

denoising function from the oracle. We prove an exponential reduction of the label complexity under the low-noise condition because the constructed denoising function has the VC-dimension of 1. In addition, an empirical comparison among different choices of noise models demonstrates that the proposed algorithm can effectively tune down the noise with fewer queries to the expensive oracle.

2 PROBLEM STATEMENT AND RELATED WORK

Noisy labels are ubiquitous for all real-world tasks. We study the problem on *enterprise* domains which are traditionally built as a system of records. These historical data facilitate bootstrapping the process from a deterministic *software* system to a data-driven *AI* system. For example, an *IT service desk* provides assistance to employees to troubleshoot their IT-related issues. The system creates an *incident* to engage with a user, and human agents classify the incidents into different categories and route them to different IT teams for resolution. See examples in Figure 2. The accuracy of the classification is crucial for incidents being resolved with low latency because the mis-routing can cause significant delay within the system. The machine learning task is to learn to categorize from the data recorded by human agents. The challenge is that these historical data contain significant errors in labels that bottlenecks the performance of the *supervised learning* approach. First of all, human agents make mistakes due to lack of training or operational errors. Secondly, different agents might have different understandings of the system, and they conflict with each other in many cases. Finally, the system routing has been changed over time, which renders the problem *non-stationary*. All of these factors result in noisy labels. Experts exist in the system who can help clean up the labels, but they are very expensive. Having them go through the entire historical record is not economic. The cost is even higher if the AI service is provided from a third party. In fact, effectively utilizing expert oracles to bootstrap an *AI* system has become a common challenge for *enterprise* AI applications.

Actively denoising labeled data is not a new topic [10, 12, 14, 20, 21, 24, 27, 29]. The problem was studied under the *inductive logic programming* framework in the early days of Machine Learning. For example, a polynomial time algorithm was proposed in [1] that identifies concepts in the form of k-CNF formulas if the labeling error rate is less than half. Research [8] introduced an *EM* algorithm to simultaneously estimate annotator biases and latent label classes if multiple annotators labeled the same example. Other work [27] applied a statistical classifier to predict the true label for each example given multiple annotations for each example. This early work focuses on the case where multiple annotators are available for each data point, which is quite restricted and unlikely to occur. Especially in many *enterprise* domains, it is almost impossible to collect multiple annotations for each data point. *CorrActive* [20] proposed an algorithm to estimate the confidence of mislabeling according to a probability model and iteratively query the oracle with the most likely mislabeled data. This algorithm does not require multiple annotators for every data point and demonstrates the effectiveness on one real application. However, it did not provide any theoretical or empirical study about how to best estimate the confidence of mislabeling.



Computer Issues

Title	Unable to login to computer
Body	Unable to login to computer as coming up with the error message account locked. Reads error code 23534902

Title	Outlook Calendar for Fred and George is not up to date
Body	Have been away on vacation and have not logged onto my computer until today, calendars for Fred and George are not updating, switching between emails and calendar it's "not responding" and the attached document has Fred's old email address which no longer exits (fred@customer.com) as well as the pop up is asking for my employee#. I need someone to execute a reset access process

Outlook Issues

Title	Outlook Calendar
Body	Under outlook calendar my old team's calendar is showing up instead of my current team. Please fix this issue.

Figure 2: Here are incidents assigned to either computer issues or Outlook issues. However, the second incident in computer issues should belong to Outlook issues but was mislabeled.

Support Vector Machines based approaches have been widely studied too [10, 24, 29, 31, 33, 37]. The main observation is that mislabeled data are likely to be support vectors, and the margin is directly related to the uncertainty of the data. In spite of its success in the past, we do not base our work on SVMs for several reasons. First of all, the recent trend of *Deep Neural Networks* has significantly reduced the feature engineering efforts during modeling, and DNN has become the de facto approach for dealing with text, audio and image data. Secondly, research study in [4, 7] has shown that the uncertainty-driven approach is biased if significant noise is present. The algorithm spends a lot of time on querying the same uncertain area repeatedly without an escaping mechanism. Lastly,

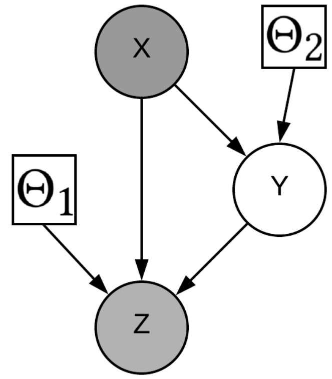


Figure 3: Probabilistic Graphical Model For Denoising. X **is the input data, i.e., caller, agent and description.** Y **is the hidden true label.** Z **is the given noisy label.** Θ_1 **and** Θ_2 **are model parameters.**

SVMs are good to model the noise from the input signals, but not to utilize the output label information.

A clustering-based approach was proposed in [6] to address the inconsistency issue for the uncertainty-driven active learning approach. The example to query is picked according to maximum information gain, i.e., how much information we can gain with respect to the distribution over both data and hypothesis. After the true label is acquired, the neighborhood's label is flipped collectively, e.g., by a majority vote mechanism. In this paper, we follow a similar approach: estimate the neighborhood through a DNN and propose the examples from highly likely mislabeled clusters. DNN provides a low-dimensional dense vector embedding to represent the semantic meaning of the original data [16, 28, 30, 36]. This representation models well on the joint manifold of both input and output labels, and creates a more accurate estimation of mislabeling clusters.

We study the approach from both Bayesian and Non-Bayesian perspectives. *Bayesian Deep Learning* [32] combines *Probabilistic Graphical Models* [17, 34] and DNNs to model noises and uncertainty within the Deep Learning framework. Algorithms like *Evidence Lower Bound Optimization* (ELBO) [23] allow fast approximate optimization to avoid intractability of the normalization function in PGMs. The *non-Bayesian* perspective further relaxes the conditional independence among data points, i.e., clusters share similar noise distributions. The uncertainty estimation is more robust with a local neighborhood smoothing than just a point estimation. Our empirical study also shows that combining the two perspectives together provides the best results.

3 ACTIVE DEEP DENOISING (ADD)

Deep Neural Networks have demonstrated a great advantage representing unstructured data, e.g., text and images. Unlike the unsupervised dimensionality reduction approach that does not take the output space into consideration, DNNs learn a metric space towards the maximal discrimination even with the presence of significant noise. This potentially helps the active learning process identify similar mislabeled patterns and correct them collectively.

3.1 Modeling noise

A high-quality noise model provides the foundation to engage an active denoising process. The probabilistic graphical model shown in Figure 3 describes the noise model where X is the input data, Y is the hidden true label, and Z is the given noisy label. We train and monitor a model that predicts the original labels, given the inputs and our estimated true label, in order to verify that our label changes result in systematic improvement. With improvement, the model is able to learn the underlying noise in the data. From the Bayesian perspective, we maximize the likelihood of the two models by marginalizing the hidden true label Y. In Equation 1, we adopt the *Evidence Lower Bound* optimization approach to approximate the marginalization.

$$\begin{aligned}
& \text{maximize}_{\Theta_1,\Theta_2} \\
& \mathbb{E}[\log \sum_{Y} P(Z, Y|X; \Theta_1, \Theta_2)] \qquad (1) \\
= \; & \mathbb{E}[\log \sum_{Y} P(Z|Y, X; \Theta_1) P(Y|X; \Theta_2)] \\
\geq \; & \mathbb{E}_{X,Z} \left[\mathbb{E}_{Y|X,Z}[\log P(Z|Y, X; \Theta_1) P(Y|X; \Theta_2)] + \mathbb{H}_{Y|X,Z} \right] \\
\geq \; & \frac{1}{|S|} \sum_{\mathbf{x}, z \in S} \sum_{y} P(y|\mathbf{x}, z)(\log P(z|y, \mathbf{x}; \Theta_1) + \log P(y|\mathbf{x}; \Theta_2)) \\
& - \frac{1}{|S|} \sum_{\mathbf{x}, z \in S} \sum_{y} P(y|\mathbf{x}, z) \log P(y|\mathbf{x}, z) - \frac{\gamma}{2}(\|\Theta_1\| + \|\Theta_2\|)
\end{aligned}$$

Both $P(z|y, \mathbf{x}; \Theta_1)$ and $P(y|\mathbf{x}; \Theta_2)$ are DNN models as depicted in Figure 4. $P(y|\mathbf{x}, z)$ is the posterior estimation of the true label given the noisy label and input data. The relaxed objective is to minimize the expected errors for both predicting noisy labels and true labels with regularizations over the posterior and the parameters. Posterior estimation by the Bayesian rule is easier to compute, but also prone to bias when data is sparse. The non-Bayesian perspective relaxes the posterior estimation to a parametrized model $P(y|\mathbf{x}, z; \Theta_3)$, e.g., with k-nearest-neighbor (kNN) smoothing.

ELBO allows the optimization of Equation 1 directly if only the Bayesian approach is taken. For non-Bayesian smoothing, jointly optimizing three models over a large input and label space becomes infeasible. Instead we propose to optimize each model individually as a coordinate descent approach, until the overall loss converges. The algorithm is presented in Algorithm 1.

The estimation of the posterior determines potential label changes as the denoising process iterates. If the true labels conform to the input attributes of similar features, $P(y = z|\mathbf{x}, y; \Theta_3) \approx 1$. Otherwise, the probability decreases. Determining how to estimate these probabilities is crucial, thus multiple methods for estimating Θ_3 are proposed and compared.

The first method is a Bayesian posterior calculated by simply inferring the hidden label from Θ_1 and Θ_2. This gives us a smooth probability distribution over a single data point $\{\mathbf{x_i}, z_i\}$, but does not take into account similar examples with varying z.

Bayesian point estimation tends to be biased in regions with sparse data. Taking a neighborhood of $\mathbf{x}$, we can take into account

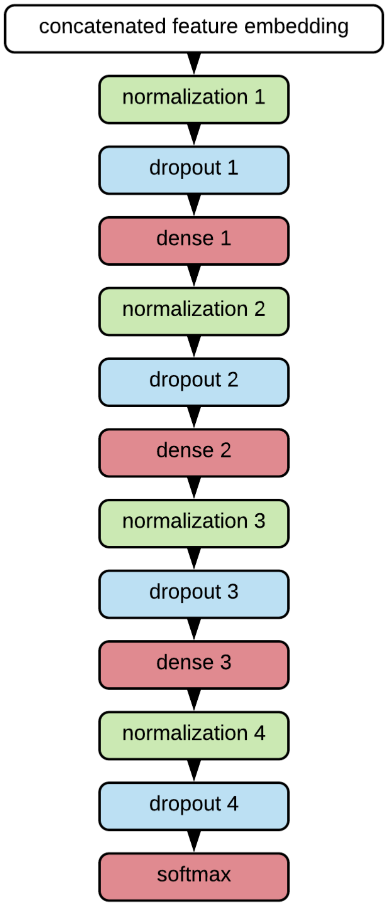


Figure 4: Deep Neural Network structure for Θ_1 and Θ_2. Our concatenated description feature vectors are fed through multiple iterations of dropout, dense layers, normalization layers, and a final softmax layer.

Algorithm 1: Modeling Noise

```
input  : $S = \{\mathbf{x}, z\}$
output : $\Theta_1, \Theta_2, \Theta_3$
begin
    Let $y = z$, $P(y|\mathbf{x}, z; \Theta_3) = 1$ if $y = z$; 0 otherwise;
    repeat
        $\Theta_2 = \arg\max_{\Theta_2} \sum_{\mathbf{x},z} \sum_y P(y|\mathbf{x}, z; \Theta_3) \log P(y|\mathbf{x}; \Theta_2) - \frac{\gamma}{2}||\Theta_2||$;
        $P(y|\mathbf{x}, z; \Theta_3) = \text{Posterior}(S, \Theta_1, \Theta_2)$;
        $\Theta_1 = \arg\max_{\Theta_1} \sum_{\mathbf{x},z} \sum_y P(y|\mathbf{x}, z; \Theta_3) \log P(z|\mathbf{x}, y; \Theta_1) - \frac{\gamma}{2}||\Theta_1||$;
    until Equation 1 converges;
end
```

Algorithm 2: Posterior by simple Bayesian rule

```
input  : $S = \{\mathbf{x}, z\}, \Theta_1, \Theta_2$
output : $P(y|\mathbf{x}, z; \Theta_3)$
begin
    $P(y|\mathbf{x}, z; \Theta_3) = \frac{P(z|\mathbf{x}, y; \Theta_1) P(y|\mathbf{x}; \Theta_2)}{\sum_y P(z|\mathbf{x}, y; \Theta_1) P(y:\mathbf{x}; \Theta_2)}$
end
```

variations to smooth out the posterior estimation. The most common surrounding label for similar inputs also indicates the true label, regardless of what given label was assigned to that point. This majority-vote mechanism is the kNN classifier where the neighborhood is based on the manifold learned from noisy data by the DNNs.

Algorithm 3: Posterior by neighborhood majority

```
input  : $S = \{\mathbf{x}, z\}, \Theta_2$, k
output : $P(y|\mathbf{x}, z; \Theta_3)$
begin
    $\mathbf{x_h}$ =the last hidden layer representation from $\Theta_2$ for all $\mathbf{x} \in S$;
    Let $\zeta$ return kNN from $\mathbf{x_h}$ given the last hidden layer representation from $\Theta_2$ with input $\mathbf{x}$;
    foreach $\mathbf{x_i}, z_i \in \{\mathbf{x}, z\}$ do
        $\{\mathbf{x_s}\} = \zeta(\mathbf{x_i})$;
        $\mathbf{w} = \sum_{\mathbf{x_{sj}} \in \{\mathbf{x_s}\}} \text{softmax}(\mathbf{x_{sj}}; \Theta_2)$;
        $\hat{y} = \arg\max w$;
        $P(y_i|\mathbf{x_i}, z_i; \Theta_3) = \mathbb{I}_{z_i = \hat{y}}$;
    end
end
```

The neighborhood is taken from a dense layer from our Θ_2 model during inference. We cluster the data based on the last dense layer as they represent the high level features in a low-dimensional space. The low dimensional manifold learned from the noisy data provides a higher-quality clustering than any unsupervised approach commonly used in other active learning algorithms. Note that this is one of the key reasons that ADD succeeds.

While the majority-vote algorithm yields the consensus of the neighborhood, the posterior distribution is effectively one hot encoded, which does not provide us a range of values to use when computing Algorithm 1. We explore a combined approach that uses the Bayesian estimation to smooth out the neighborhood.

3.2 Actively denoising labels

After the noise model is built, we select samples that our models suggest and verify these changes with the oracle. The verification process is expensive, and we are allowed K examples to get the high quality verifications on. If K has be linear on the sample size $|S|$, the oracle is required to relabel most of the data to achieve the optimal accuracy. If $K = O(\log(|S|)$, the active denoising saves the expensive cost exponentially. We show that the exponential speedup of the label complexity is feasible under the low-noise

```
Algorithm 4: Posterior by neighborhood smoothing
  input  : $S = \{\mathbf{x}, z\}, \Theta_2$, k
  output : $P(y|\mathbf{x}, z; \Theta_3)$
begin
    $\mathbf{x_h}$ =the last hidden layer representation from $\Theta_2$ for all $\mathbf{x} \in S$;
    Let $\zeta$ return kNN from $\mathbf{x_h}$ given the last hidden layer representation from $\Theta_2$ with input $\mathbf{x}$;
    foreach $\mathbf{x_i}, z_i \in \{\mathbf{x}, z\}$ do
        $\{\mathbf{x_s}\} = \zeta(\mathbf{x_i})$;
        $P(y_i|\mathbf{x_i}, z_i; \Theta_3) = \sum_{\mathbf{x_{sj}} \in \{\mathbf{x_s}\}} \text{softmax}(\mathbf{x_{sj}}; \Theta_2)$;
    end
end
```

condition as we construct a threshold based denoising function that has the VC-dimension of 1.

Since we were looking for examples that could generalize to other examples for relabeling, we looked at the clustering methods that are used to flip the labels during modeling in Section 2. When constructing Θ_3 to determine the true label, the estimations were made for every single point using a learned hidden representation. This representation is used to create the vectors that are clustered to determine similarly-structured examples. This led our true label model Θ_2 to achieve a much higher accuracy than any base model Θ_1. In selection, we have the knowledge of Θ_2, but need to verify those changes with an oracle. We use the metric in Equation 2 to indicate the score of mislabeling, and select samples accordingly.

$$f(\mathbf{x}, z) = \max_{y \neq z} \log P(y|\mathbf{x}, z; \Theta_3) - \log P(z|\mathbf{x}; \Theta_1, \Theta_2) \tag{2}$$

This metric also allows us to define the denoising function parametrized by the threshold h as shown in Equation 3 where $\hat{y}$ is the label that attains the maximum value of $f(\mathbf{x}, z)$.

$$\hat{y}(\mathbf{x}, z; h) = \hat{y} \text{ if} f(\mathbf{x}, z) > h \text{ otherwise } z \tag{3}$$

Proposition 3.1 shows that the true noise level can be upper bounded through the VC bounds where VC-dimension of the denoising function Equation 3 is 1. The active learning process is to effectively query the oracle to approximate the true entropy with fewer samples. We just need to sort the data points according to the scores and verify the point from the oracle that all data points above are mislabels. We adopt the *Agnostic Active Learning* approach to estimate the target threshold.

Proposition 3.1. *Let ξ represent the true noise level in the labels where $\bar{y}$ represents the unknown true labels. In addition, Δ is the set of data points with verified true labels, and $|\Delta| \ll |S|$. η is a pre-defined confidence parameter that $\eta < 1/2$.*

$$\xi = -\frac{1}{|S|} \sum_{\mathbf{x}, z, \bar{y} \in S} P(\bar{y}|\mathbf{x}, z; \Theta_3) f(\mathbf{x}, z) \tag{4}$$

$$\xi \leq -\frac{1}{|\Delta|} \sum_{\mathbf{x}, z, \bar{y} \in \Delta} P(\bar{y}|\mathbf{x}, z; \Theta_3) f(\mathbf{x}, z) + \sqrt{\frac{\log 1/\eta}{2|\Delta|}} \tag{5}$$

$$\xi \geq -\frac{1}{|\Delta|} \sum_{\mathbf{x}, z, \bar{y} \in \Delta} P(\bar{y}|\mathbf{x}, z; \Theta_3) f(\mathbf{x}, z) - \sqrt{\frac{\log 1/\eta}{2|\Delta|}} \tag{6}$$

The active denoising algorithm is defined in Algorithm 5. Note that the disagreement function only needs to check the disagreements on the boundary points, i.e., $H = [h_1, h_2)$. Corollary 3.2, as a straight application of A^2 (Agnostic Active) algorithm's result, proves that under the low-noise condition the label complexity is on the log-scale, otherwise it is on the quadratic scale.

```
Algorithm 5: Active Denoising
  input  : $S = \{\mathbf{x}, z\}, \Theta_1, \Theta_2, \Theta_3$, Oracle $O$ and small error $\epsilon$
  output : $\hat{h}$
begin
    Let $H = (-\infty, +\infty)$, $\Delta = \{\}$;
    Let $U(h, \Delta)$ denote Equation 5;
    Let $L(h, \Delta)$ denote Equation 6;
    repeat
        Sample $2\Delta + 1$ from $S$ where $\hat{y}(\mathbf{x}, z; h)$ disagrees on different $h \in H$;
        Query the oracle $O$ for $\bar{y}$;
        Add these samples $\{\mathbf{x}, z, \bar{y}\}$ to $\Delta$;
        $\hat{h} = \arg\min_{h \in H} U(h, \Delta)$;
        $H = \{h \in H; L(h, \Delta) \leq U(\hat{h}, \Delta)\} = [h_1, h_2)$;
    until $|\hat{y}(\mathbf{x}, z; h_1) \neq \hat{y}(\mathbf{x}, z; h_2)|$ decreases by less than half or $U(\hat{h}, \Delta) \leq \epsilon$;
end
```

Corollary 3.2. *If $\xi \leq \frac{\epsilon}{16}$, with a high probability $1 - \eta$, Algorithm 5 makes $O(\ln(\frac{1}{\epsilon} + \frac{1}{\eta}))$ calls to the oracle. Otherwise, it makes $O(\frac{\xi^2 \ln \frac{1}{\eta}}{\epsilon^2})$ calls.*

Algorithm 5 is not restricted to the metric function defined in Equation 2. It can be generalized to other metric functions as well. In fact, in our experimentation, we smooth out the estimation by grouping data points together, and measure the aggregate noise level. This additional aggregation allows the noise level estimation to be more stable due to the lower variance. Therefore, fewer iterations are required to achieve the optimal denoising threshold.

4 EXPERIMENTS AND ANALYSIS

For our experiments we used one of our incident categorization datasets with 150k incidents available for training, validation, and testing. In total the dataset contains 403 categorical labels that need to be classified. The distribution of these categorical labels are highly skewed, which is mitigated by sample weights during model training (Figure 5).

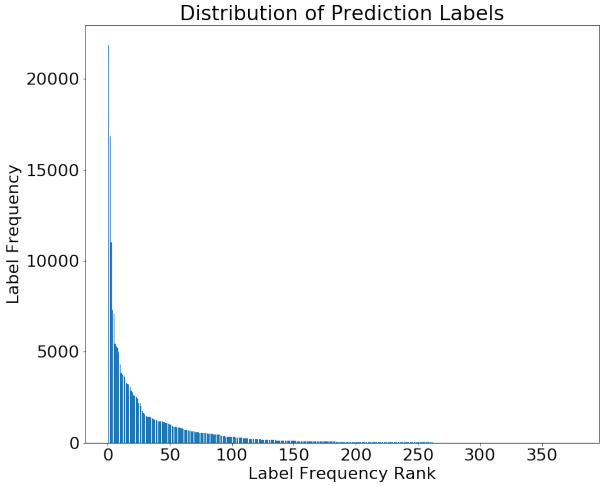


Figure 5: The label distribution used for prediction.

We focus on modeling the accuracy gains, given a similar DNN model architecture, over the accuracies of the DNN models themselves. The DNN structure, as described in Figure 4, contains 3 sets of dropout, dense, and normalization layers before the final softmax prediction layer. This structure was chosen as it was already parameterized and performing well for incident inference, so we wanted to augment the performance further.

4.1 Denoising comparisons

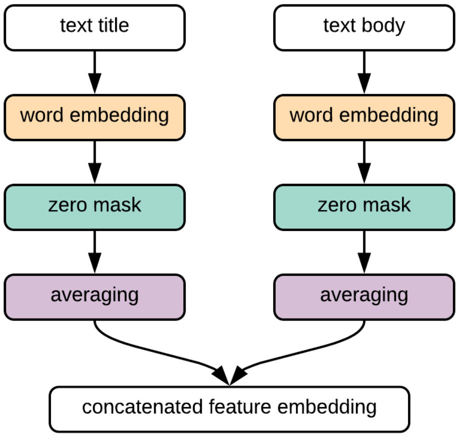


Figure 6: Average embedding representation. The input descriptions are taken in and embedded as vectors. Then the non-zero vectors are removed, and an average vector is computed before the features are concatenated.

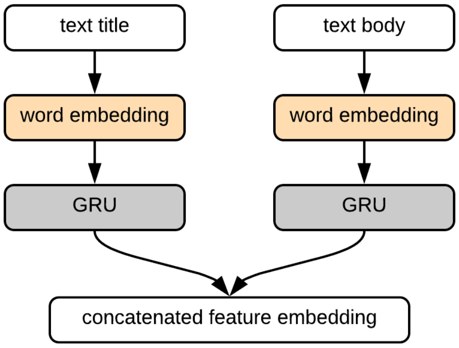


Figure 7: An embedding using Gated Recurrent Unit (GRU) layers. Like the average embedding, the inputs are taken and converted into word vectors. These word vectors are fed into the GRU layers, which output temporal representations of word embeddings provided by the inputs.

We first demonstrate the denoising accuracy gains by comparing the Bayesian and neighborhood Θ_3 modeling methods as described in Algorithm [2,3,4]. We also test two different approaches for embedding our feature vectors to input into our DNN. We try both an average word vector embedding (Figure 6), and a learned recurrent word vector representation (Figure 7). Our convergence criterion is estimated as follows:

$$|(\Theta_1 \text{ accuracy}+\Theta_2 \text{ accuracy})_{t-1}-(\Theta_1 \text{ accuracy}+\Theta_2 \text{ accuracy})_t| \leq \delta$$

Here t is the iteration number for the repeat loop described in Algorithm 1. In our tests we set the $\delta = 0.01$ for convergence test.

Once the models are computed, we adapted the denoising Algorithm 5 for the industry setting. One issue is that many incidents sent for IT Agent verification could be similar when successively sampling under the denoising algorithm. To address this, we sample a single incident for annotation from the errors grouped by (z, y) pairs, reducing the probability of sending an agent a similar incident. Another logistic issue is that the IT agent is less willing to wait for multiple iterations of denoising. We approach this problem with a batch adaptation of denoising, where samples from multiple pair groups are queried ahead of time. These methods are laid out in Algorithm 6, where we typically set h such that $|A| = 100$. Feedback for the annotations is then requested from the agent.

When asking for feedback we provide an explanation as to why the model would make each label prediction (Figure 8). This allows agents to understand what the model is looking at when the model is making a label prediction, based on an approximation of the network features [25]. It also serves as a debugging process if the model is under suspicion of overfitting, given the generalizability of the features returned.

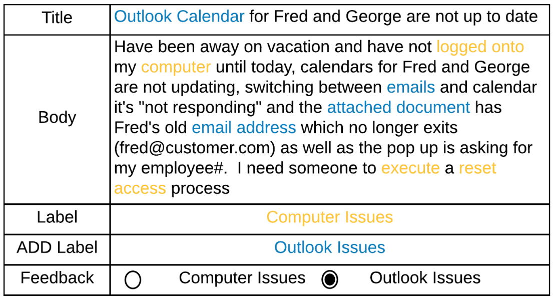
For each verified error pair, all $\hat{y}$'s of the error pair's corresponding error pair group are set to $\bar{y}$. These verified denoised values are saved as the last step in the full denoising process (Figure 9).

Table 1: Modeling Accuracies and statistics across different combinations

Embedding	Method	Base Accuracy	Denoised Accuracy	Accuracy Gain	Entropy Reduction	% Labels Changed
Average Vector	Bayesian	0.7572	0.7877	0.0305	0.0127	13.2531
	Neighbor-Majority	0.7572	0.8543	0.0971	0.0981	13.1244
	Neighbor-Smooth	0.7562	0.8570	**0.1008**	0.1097	13.1501
GRU	Bayesian	0.7288	0.7676	0.0388	-0.0169	8.9279
	Neighbor-Majority	0.7253	0.7795	0.0542	0.1093	9.0586
	Neighbor-Smooth	0.7283	0.7825	0.0542	0.1226	9.0174

Algorithm 6: Annotation Selection

```
input  : $S = \{\mathbf{x}, z\}, h > 0, \Theta_3$
output : $A$ (data to re-annotate) such that $A \in S$
begin
    $G = \{z, \hat{y} \mid \mathbf{x}, z \in S, \hat{y} = \arg\max P(\mathbf{x}, z; \Theta_3)\}$;
    $E = \{\{(z, y, \mathbf{x}) \mid \mathbf{x}, z' \in S$ where $z' = z$ and $\hat{y} = y\}$
      $(z, y) \in G$ where $y \neq z\}$;
    $A = \{\{\mathbf{x}, z, y \mid$ random sample $\mathbf{x}, z, y \in e$ where
      $\sum_{\{\mathbf{x}, z\} \in e} \sum P(\mathbf{x}, z; \Theta_3) > h\} e \in E\}$;
end
```



Title	Outlook Calendar for Fred and George are not up to date
Body	Have been away on vacation and have not logged onto my computer until today, calendars for Fred and George are not updating, switching between emails and calendar it's "not responding" and the attached document has Fred's old email address which no longer exits (fred@customer.com) as well as the pop up is asking for my employee#. I need someone to execute a reset access process
Label	Computer Issues
ADD Label	Outlook Issues
Feedback	○ Computer Issues ● Outlook Issues

Figure 8: Feedback layout presented to the oracle. The original label and the proposed ADD label are provided. The color corresponding to each label are overlaid on the original text, indicating which n-grams are influential in the corresponding label predictions.

Afterwards, $\hat{y}$ can be utilized to train a final, agent-backed model. The final model, $\Theta_{denoised}$, is trained on $\{\mathbf{x}, \hat{y}\}$ and compared with Θ_{base}, which is trained on $\{\mathbf{x}, y\}$. We repeated this for every combination of average/GRU embedding and Θ_3 modeling methodology.

For each run we keep track of a couple of metrics. The key metric measure is $\Theta_{denoised} - \Theta_{base}$, in order to view our final impact of our verified label changes on the newly-trained model over the previous best. We calculate the entropy over the predictions for all the data and keep the mean entropy reduction over all mean label entropies. This is the entropy reduction calculated from Θ_{base} to $\Theta_{denoised}$. This metric shows us if our modifications to the labels increased the overall confidence to predict each category, due to the decrease of noisy labels. For a sanity check we also keep track of the percentage of labels flipped while the process is carried out.

We analyze the effect of accuracy increase given the number of feedbacks requested of the oracles. We compare this to a baseline test of randomly sampling examples and making the label changes to the data directly (Figure 11). This clearly demonstrates the effectiveness of the algorithm while a random sampling approach represents the label complexity for the traditional *supervised learning*.

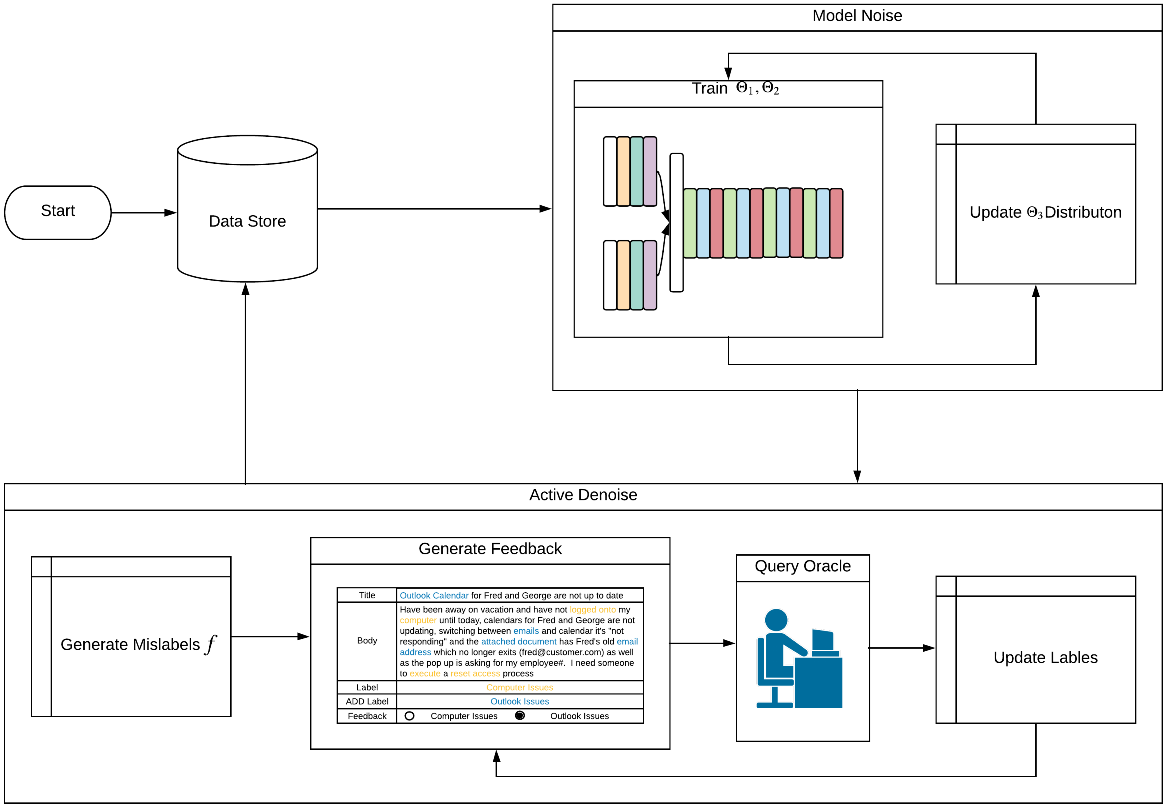
4.2 Denoising results

From the results in Table 1, the average embedding using the smooth kNN approach gives the best results in our data. Intuitively, this makes sense, as the smooth estimation generalizes better to the test data and future variation. We were expecting the GRU performance to be on par or exceeding that of the average embedding, but the accuracy improvements in the test were halved. Looking at our data, this result is not surprising, as the text bodies can often be largely segmented and discontinuous pieces of text.

The entropy reduction scores give interesting results. The entropy drop for our best accuracy improvement (Table 1), average embedding with smooth kNN, is on par the GRU kNN majority approach but doubled the accuracy gain. We proceeded to explore the loss reductions per category basis, looking at the most frequent categories. For the top 7 recurring categories (Figure 10), there is an average entropy reduction of 0.09 and 0.05 for the average and the GRU methods respectively. With the only difference being the embedding methodology, we infer that the patterns learned by the GRU generalized over many category labels. Observing the text body of incidents show us that they are structured very similarly for different categories, for example: "I am not able to login to x.", where x can be any concept ranging from payroll, email, environments, etc. We suspect that our GRU is able to capture this general structure, but we could not achieve a robust enough (without overfitting) model to distinguish the nuances with "x", each of which may belong to different category labels. In comparison, the averaging embedding model is able to better focus on these discrepancies.

Figure 10 also shows that the most common category had an insignificant entropy reduction, while the next most common labels were more significant. The most common label ended up being the most general label, computer issues. Many specific incidents are assigned to this category despite other more nuanced labels being present, such as email, browser, that closely match these incidents. Thus consistency within the general category does not improve, as no consensus regarding what is supposed be within that general computer label group is provided. Therefore, the posterior

Figure 9: A full view of the end-to-end ADD process.



Title	Outlook Calendar for Fred and George are not up to date
Body	Have been away on vacation and have not logged onto my computer until today, calendars for Fred and George are not updating, switching between emails and calendar it's "not responding" and the attached document has Fred's old email address which no longer exits (fred@customer.com) as well as the pop up is asking for my employee#. I need someone to execute a reset access process
Label	Computer Issues
ADD Label	Outlook Issues
Feedback	○ Computer Issues ● Outlook Issues

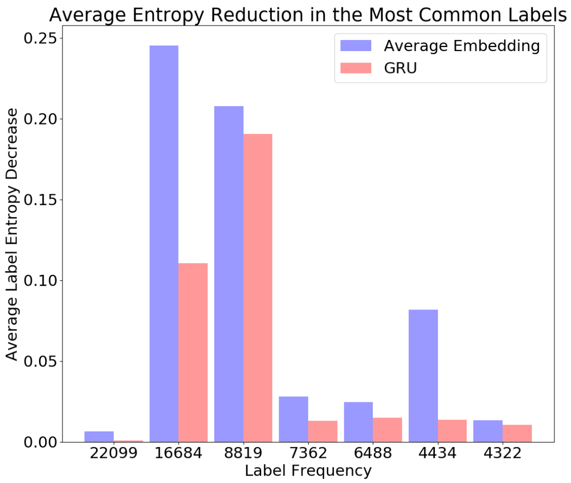


Figure 10: Assessing the reduction of mean entropy per label after generating $\Theta_{denoised}$ vs. Θ_{base}. This compares how the average embedding and GRU representations compared when it came to reducing model loss in the most frequent categories.

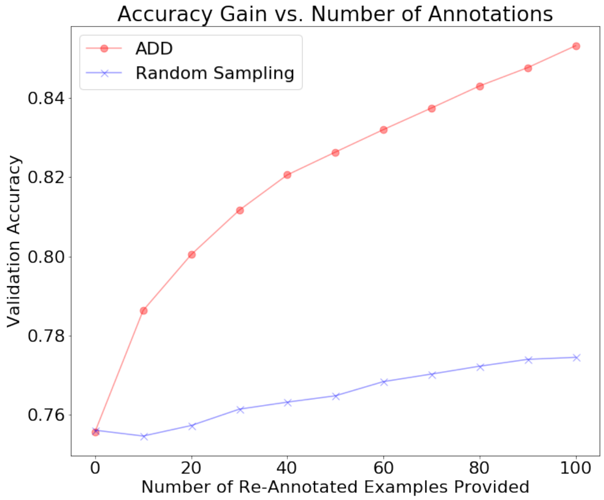


Figure 11: The accuracy vs. the number of labels queried

for that computer label decreased while the probabilities for the more specific categories increased. Based on the shift between these labels, the percentages of label changes fall within the range of what we expected (Table 1), which is in the range of 10-20%, based on observational random sampling.

Figure 11 shows how the accuracy improves during the first few iterations for our ADD method. This is partially due to the fact that confirmation from the oracle within our h thresholds leads to more corrections as it covers larger error pair groups. As the labels become denoised, the cardinality of the error pair groups shrinks, providing fewer corrections than before. In comparison, randomly-corrected samples increase the accuracy linearly over time. Most importantly, a DNN noise model effectively represents both input and output space in a smooth manifold so that the neighborhood clustering effectively groups similar errors and allow the denoising function to correct them collectively.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose a DNN based label denoising approach. We first model the noise with a Bayesian DNN. Given the DNN, we develop Bayesian and non-Bayesian techniques to calculate the posterior label probability for every incident. Then, we query the oracle with an active denoising algorithm, which selects examples that provide label verification for similarly-structured data points. Once these mislabels of similar examples are corrected, the DNN is trained on these more-confident labels and the process can be repeated. We apply this approach on the IT incident categorization dataset where the experiment results demonstrate significant accuracy improvements by just querying a small portion of mislabeled data. We compare our results among Bayesian, non-Bayesian (neighborhood majority), and combined approach (neighborhood smooth). The combined approach demonstrates the best accuracy improvement. To our surprise, GRU-based DNN produces an inferior result than a simple Average Embedding based DNN. As ADD can be generalized to other data, we look forward to comparing its noise detection performance on public datasets as well.

ACKNOWLEDGEMENTS

We would like to thank Zhenjie Zhang, Ankit Arya, Masayo Iida, Sara Woodhull, and Naghi Prasad for their insights and feedback regarding the project.

REFERENCES

[1] Dana Angluin and Philip Laird. 1988. Learning from noisy examples. *Machine Learning* 2, 4 (1988), 343–370.

[2] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. 2009. Agnostic active learning. *J. Comput. System Sci.* 75, 1 (2009), 78–89.

[3] David Cohn, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine learning* 15, 2 (1994), 201–221.

[4] Sanjoy Dasgupta. 2005. Analysis of a greedy active learning strategy. In *Advances in neural information processing systems*. 337–344.

[5] Sanjoy Dasgupta. 2006. Coarse sample complexity bounds for active learning. In *Advances in neural information processing systems*. 235–242.

[6] Sanjoy Dasgupta and Daniel Hsu. 2008. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 208–215.

[7] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. 2008. A general agnostic active learning algorithm. In *Advances in neural information processing systems*. 353–360.

[8] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* (1979), 20–28.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[10] Rajmadhan Ekambaram. 2017. *Active Cleaning of Label Noise Using Support Vector Machines*. Ph.D. Dissertation. University of South Florida.

[11] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. 1997. Selective sampling using the query by committee algorithm. *Machine learning* 28, 2-3 (1997), 133–168.

[12] Dragan Gamberger, Nada Lavrac, and Saso Dzeroski. 2000. Noise detection and elimination in data preprocessing: experiments in medical domains. *Applied Artificial Intelligence* 14, 2 (2000), 205–223.

[13] Steve Hanneke. 2007. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*. ACM, 353–360.

[14] Dominik Henter, Armin Stahl, Markus Ebbecke, and Michael Gillmann. 2015. Classifier self-assessment: active learning and active noise correction for document classification. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 276–280.

[15] Ashish Kapoor, Eric Horvitz, and Sumit Basu. 2007. Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning.. In *IJCAI*, Vol. 7. 877–882.

[16] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[17] Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

[18] Diego Marcheggiani and Fabrizio Sebastiani. 2017. On the Effects of Low-Quality Training Data on Information Extraction from Clinical Reports. *J. Data and Information Quality* 9, 1, Article 1 (Sept. 2017), 25 pages.

[19] Andrew Kachites McCallumzy and Kamal Nigamy. 1998. Employing EM and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning (ICML)*. 359–367.

[20] Ramesh Nallapati, Mihai Surdeanu, and Christopher Manning. 2009. Corractive learning: Learning from noisy data through human interaction. In *IJCAI Workshop on Intelligence and Interaction*.

[21] Natalie Parde and Rodney Nielsen. 2017. Finding Patterns in Noisy Crowds: Regression-based Annotation Aggregation for Crowdsourced Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1907–1912.

[22] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP*.

[23] Rajesh Ranganath, Sean Gerrish, and David Blei. 2014. Black box variational inference. In *Artificial Intelligence and Statistics*. 814–822.

[24] Umaa Rebbapragada and Carla E Brodley. 2007. Class noise mitigation through instance weighting. In *European Conference on Machine Learning*. Springer, 708–715.

[25] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 1135–1144.

[26] B. Settles. 2012. *Active Learning*. Morgan & Claypool.

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

[28] Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2014. Using active learning and semantic clustering for noise reduction in distant supervision. In *4e Workshop on Automated Base Construction at NIPS2014 (AKBC-2014)*. 1–6.

[29] Jack W Stokes, Ashish Kapoor, and Debajyoti Ray. 2016. Asking for a second opinion: Re-querying of noisy multi-class labels. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2329–2333.

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[31] Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research* 2, Nov (2001), 45–66.

[32] Dustin Tran, Matthew D Hoffman, Rif A Saurous, Eugene Brevdo, Kevin Murphy, and David M Blei. 2017. Deep probabilistic programming. *arXiv preprint arXiv:1701.03757* (2017).

[33] Hamed Valizadegan and Pang-Ning Tan. 2007. Kernel based detection of mislabeled training examples. In *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 309–319.

[34] Martin J Wainwright, Michael I Jordan, et al. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1, 1–2 (2008), 1–305.

[35] Manfred K Warmuth, Gunnar Rätsch, Michael Mathieson, Jun Liao, and Christian Lemmen. 2002. Active Learning in the Drug Discovery Process. In *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.). MIT Press, 1449–1456.

[36] Shusen Zhou, Qingcai Chen, and Xiaolong Wang. 2013. Active deep learning method for semi-supervised sentiment classification. *Neurocomputing* 120 (2013), 536–546.

[37] Xingquan Zhu, Xindong Wu, and Qijun Chen. 2003. Eliminating class noise in large datasets. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 920–927.