Auto-Annotation Quality Prediction for Semi-Supervised Learning with Ensembles

Dror Simon
Technion, Israel
dror.simon@cs.technion.ac.il

Miriam Farber
Amazon.com
mffarber@amazon.com

Roman Goldenberg
Amazon.com
romang@amazon.com

Abstract

Auto-annotation by an ensemble of models is an efficient method of learning on unlabeled data. However, wrong or inaccurate annotations generated by the ensemble may lead to performance degradation of the trained model. We propose filtering the auto-labeled data using a trained model that predicts the quality of the annotation from the degree of consensus between ensemble models. Using semantic segmentation as an example, we demonstrate the advantage of the proposed auto-annotation filtering over training on data contaminated with inaccurate labels. We show that the performance of a state-of-the-art model can be achieved by training it with only a fraction (30%) of the original manually labeled samples, and replacing the rest with auto-annotated, quality filtered labels.

1. Introduction

Semi-supervised learning, i.e. using the combination of a smaller set of labeled data and a larger set of unlabeled data, is becoming increasingly important with the growing capacity of trained models and their tasks complexity. Ensembles of models have been successfully used for automatic annotation of unlabeled data [1, 24]. In this setting, an ensemble is first formed by multiple instances of a target model, each trained on labeled data. The ensemble, which is said to be more accurate than a single model, then labels the unlabeled data, which is later used in a training procedure. Unfortunately, this self-labeling approach raises one issue that is often not properly addressed: the annotation quality. Specifically, using wrong or inaccurate annotations for training may negatively affect the target model.

In this work, we propose a method for predicting the quality of the annotations generated by an ensemble. The approach uses a model trained to assess the quality of the generated annotation from the degree of consensus between the models within the ensemble. Then, we propose to refine the auto-labeled data set by discarding samples with low predicted annotation quality. We show that training on a refined and reduced set is advantageous over using a lager set, contaminated with inaccurate labels. Using semantic segmentation as an example, we demonstrate that the proposed method achieves the same accuracy as a state-of-the-art model, while using only a fraction (30%) of the labels in the training set.

The main contributions of this paper are: (a) a method for automatic filtering of auto-annotations generated by ensembles, using a trained model that predicts annotation quality, and (b) a novel auto-annotation quality control scheme for semantic segmentation, which filters bad labels at the pixel level, yielding a refined partial image labeling.

2. Related work

Since it was demonstrated that ensembles of models can boost the accuracy [7, 4] of a single model, they have been used extensively to achieve state-of-the-art performance in various tasks [19, 16, 9]. Using ensembles of models for self-training was proposed in [1], where knowledge distillation was performed on unlabeled data by an ensemble trained on a smaller set of labeled data. The idea of self-labeling goes back to 1965 [18], and since then was a subject of research in semi-supervised learning [26]. The benefits of using ensembles for semi-supervised learning are advocated in [25]. Here we propose a model distillation regime where the quality of the labels generated by an ensemble is estimated by an additional, second-level model. This is closely related to the stacked generalization [22] meta-learning technique [5]. Unlike the method in [8] that uses soft labels for knowledge distillation, we train a network to filter out unreliable labels.

We demonstrate the effectiveness of the proposed technique on the example of semantic segmentation [6]. Prior work on not fully supervised semantic segmentation include weakly and partially supervised techniques [10, 14, 21], self-supervision [23] and ensemble knowledge transfer [13]. To the best of our knowledge, this is the first work that performs semi-supervised training of semantic segmentation model on auto-annotated unlabeled data, generated by ensembles with quality filtering.

3. Annotation quality prediction

Let $f : \mathbb{X} \rightarrow \mathbb{Y}$ be the target model to be trained. We use an ensemble of models $e = (f_1, \ldots, f_k), f_j : \mathbb{X} \rightarrow \mathbb{Y}$, to automatically label the unlabeled data for training the target model f. Models in e are trained on a labeled data set $\mathbb{S} = \{(x^{(i)}, \bar{y}^{(i)})\}$, $x^{(i)} \in \mathbb{X}$, $\bar{y}^{(i)} \in \mathbb{Y}$. We use the ensemble to generate labels for the large unlabeled set $\mathbb{U} = \{x^{(i)}\}$ in the following way:

- Run $e : \mathbb{X} \rightarrow \mathbb{Y}^k$ on $\mathbb{U}$ to generate vectors of labels $\mathbb{L} = \{(y_1^{(i)}, \ldots, y_k^{(i)})\}$.
- Apply a fusing function [16] $g : \mathbb{Y}^k \rightarrow \mathbb{Y}$ to combine the ensemble labels into a single label: $\widehat{\mathbb{L}} = \{\hat{y}^{(i)}\} = g(\mathbb{L}) = (g \circ e)(\mathbb{U})$.

Finally, we train the target model on the generated labeled set $\mathbb{T} = \{\mathbb{U}, \widehat{\mathbb{L}}\} = \{(x^{(i)}, \hat{y}^{(i)})\}$. Generally, some of the automatically generated labels in $\widehat{\mathbb{L}}$ are expected to be wrong. Therefore, a supposedly better approach would be to remove the corresponding data samples from $\mathbb{T}$.

In order to do that, we propose to train a function q, that predicts the quality of the labels generated by an ensemble, based on the degree of consensus within it. The function $q : \mathbb{Y}^k \rightarrow \{0, 1\}$ receives the ensemble output $(y_1, \ldots, y_k)$ and generates a quality score in $\{0, 1\}$. This is similar to Wolpert's ensemble stacking approach [22], but instead of merging ensemble outputs to generate a fused labeling, the q function predicts the labeling quality for a fixed fusor g. The function q is trained using a labeled data set $\mathbb{Q} = \{(x^{(i)}, \bar{y}^{(i)})\}$. For a data sample $(x, \bar{y}) \in \mathbb{Q}$, the input for the q model is the ensemble output $e(x)$ and the ground truth is the indicator function $\mathbb{1}_{(g \circ e)(x) = \bar{y}}$. We then use q to filter the auto-annotated set $\mathbb{T}$ by discarding data samples with low predicted annotation quality to yield a refined labeled data set $\mathbb{T}^* = \{(x, \hat{y}) \in \mathbb{T} \mid q(e(x)) = 1\}$.

4. Semi-supervised semantic segmentation with auto-annotation quality prediction

To demonstrate the proposed approach, we implement a semi-supervised training of a semantic segmentation model using auto-annotation with ensemble of models and quality filtering. We chose the task of semantic segmentation since creating manual annotations for this task is extremely labor intensive, resulting in a relatively limited amount of such annotations. Therefore, highly accurate auto-annotations could be especially useful for this task.

To populate the ensemble we use both multiple models and data augmentation, following the data distillation method presented in [15]. First, we train the same model multiple (three) times using different parameters initialization and training samples reshuffling. In addition, each model is fed six augmented versions of the input image: two horizontal flips $\times$ three scales (x0.5, x1.0, and x1.5).

In the experiments described in the next section we merge ensemble results into a single label using a simple softmax averaging [17, 20, 11, 12]. We refer to a collection of such labels as unfiltered auto annotations. The fusing function g is defined as $g(\sigma_1, \ldots, \sigma_k) = \mathrm{argmax}_{i \in [1, \ldots, C]} \sum_{j=1}^{k} \sigma_j$, where C is the number of classes in the semantic segmentation model, and $\sigma_j \in \mathbb{R}^C$ is a softmax class probability vector generated by the j-th model of an ensemble of size k.

Interestingly, since semantic segmentation models can be trained on a partially labeled image, we do not necessarily need to accept or discard the image labeling as a whole. Instead, we can do it selectively, by making a decision per pixel. Pixels with unreliable labeling are marked as a special "ignore" class in the labels mask and do not contribute to the gradient back-propagation during the training.

We implement q using a convolutional neural network (CNN) that receives $k \times C$ input channels - $(\sigma_1, \ldots, \sigma_k)$ and outputs a quality mask in $\{0, 1\}$. It has 4 hidden conv-ReLU layers. The first layer has 40 output channels while the rest of them have 20.

5. Experiments

In this section, we demonstrate the strength of the proposed quality prediction approach. We show that a model trained on a combination of manually labeled data and quality filtered auto-annotations achieves better performance than its unfiltered counterpart. In fact, we show that quality filtering allows replacing a significant amount of manually-annotated images by auto-annotated images, without any degradation in accuracy. The experiments are performed using the DeepLab model [2], a state of the art network for semantic segmentation. The training protocols follow [2].

We evaluate our approach on the pixel-level semantic labeling task of Cityscapes data set [3]. This data set has 19 semantic labels (and additional void labels that are not used for evaluation), and consists of 5000 images, which are split into training, validation and test sets of sizes 2975, 500, and 1525 respectively. We report *mIoU* (mean *IoU*) scores and *IoU* scores of each one of the 19 classes following the Cityscapes definitions. Our experiments involve several data splits. In all these splits, all the 19 classes are represented: **Cityscapes full**, **Cityscapes small**, and **Cityscapes tiny** training sets consisting of 100%, 30% and 15% of the labeled training images, respectively. In **Cityscapes extra set**, we sample 3600 images from the 20000 coarsely annotated Cityscapes data set. We do not use any available labels from this data set, but only the images themselves.

5.1. Training with and without quality prediction

We start by showing that a model trained on manually labeled data + quality filtered auto-annotations outperforms

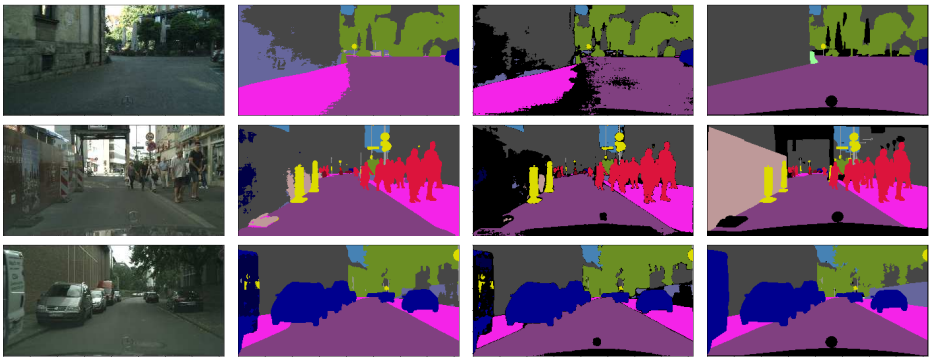

Figure 1. Columns - left to right: (1) Original image, (2) unfiltered auto-annotations, (3) quality filtered auto-annotations, (4) ground truth annotation. Rows: three different examples. Black pixels in the fourth columns represent void classes that are not counted as part of the 19 classes and do not contribute to the *mIoU* score. Black pixels in the third column represent pixels that are masked out by the quality filter.

a model trained on manually labeled data + unfiltered auto-annotations by 0.8% in terms of *mIoU*.

First, we trained a DeepLab network on the Cityscapes tiny training set 3 times. In our experiments, the best network in this setting obtained *mIoU* of 74.0% on the validation set. Then, we produced unfiltered auto-annotations for the Cityscapes extra set by fusing the ensemble's results. The ensemble consists of the 3 trained models together with 6 augmentations associated with each model. We proceed by training two additional models. The first is a DeepLab model trained on a data set that includes the Cityscapes tiny training set and the unfiltered auto-annotations described above. The network's *mIoU* on validation set is 75.5%. Next, we refine the auto-annotations by applying the quality filter, described in section 3. This quality filter is also trained on the Cityscapes tiny training set. Finally, we train a model on a data set which consists of the Cityscapes tiny training set and the quality filtered auto-annotations. This leads to an additional improvement of 0.8% in *mIoU*, reaching 76.3%. Thus, we obtain an overall improvement of 2.3% in *mIoU* on the validation set (from 74.0% to 76.3%).

We chose the tiny training set for this experiment, as in this scenario the auto-annotated data forms large portion of the training data. In fact, the auto-annotated data forms roughly 90% of the training data used for the training procedures depicted in the paragraph above. This mimics the real world scenario in which available unlabeled data sets are much larger than their manually-labeled counterparts. Our approach shows that such data sets can be utilized effectively to improve the performance of a trained network. Performing a class-wise IoU examination of the described experiment (Table 1) leads to two conclusions:

Conclusion #1: a model trained on manually labeled data + quality filtered auto-annotations outperforms a model trained on manually labeled data + unfiltered auto annotations on each one of the 19 classes. Indeed, in Table 1, the values in the fifth column surpass the values in the fourth one. This demonstrates the benefit of the quality filter. The largest gain is demonstrated for the rarest classes (lower rows), with 4% gain to the rarest class (train).

Conclusion #2: the proposed approach performs significantly better on underrepresented classes, compared to a model trained on manually labeled data only, adding up to ∼20% to the IoU in such classes. This conclusion is obtained by comparing the third and the fifth columns in Table 1. Specifically, rows (classes) in which the IoU of the model trained on manual labels + quality filtered auto-annotations surpasses the IoU of the model trained on manual annotations only are highlighted in green. Those that perform the same are highlighted in yellow. One can see that in 14 out of 19 classes, the performance of the former model is at least as good as the latter. In 11 out of these classes, the performance improved. The most significant improvement is obtained in under-represented classes like train (19.7%), bus (7.8%), truck (9.6%), and fence (4.9%).

We now experiment with manually-labeled training sets of various sizes. Specifically, we show that adding quality filtered auto-annotated data improves the model's performance in all cases, with the largest improvement achieved for the smallest manually-labeled training set. Following the same procedure described earlier, we train a model first on manually labeled data only, and then on manually labeled data + quality filtered auto-annotated data. We repeat the experiment 3 times by reducing the manually labeled training set from 100%, to 30% and, finally, to 15%. These results on are summarized in Table 2. Specifically, the accuracy of the network trained on the Cityscapes small training set+auto-annotated data is 77%, which is identical to the accuracy of the network trained on Cityscapes full training set. This demonstrates that using our approach with only 30% of the available manual annotations, leads to the same performance as training with the entire manually annotated training set (saving 70% of manual annotations).

5.2. Performance of the quality filter

In this section, we shed some light on the performance of the quality filter itself. Columns 6-7 in Table 1 report the precision rates of the filtered and unfiltered auto-annotations respectively, computed on the ground truth labels. These columns show that indeed the quality filter improves the precision of the annotations for all classes, leading to the reported improvement in IoU (columns 4-5).

Class	Ground truth occurrences	IoU per class			Precision		Retention
		Manual	Manual + unfiltered	Manual + filtered	filtered	unfiltered	
building	500	91.7%	91.6%	91.7%	97.0%	95.7%	97.5%
sidewalk	499	81.8%	82.4%	83.1%	94.5%	89.9%	86.5%
pole	499	63.3%	61.2%	62.2%	90.8%	81.3%	70.6%
road	498	97.7%	97.7%	97.8%	99.7%	99.1%	97.9%
vegetation	493	92.1%	92.0%	92.1%	96.5%	95.4%	97.5%
traffic sign	487	76.5%	76.0%	76.5%	95.3%	92.4%	89.5%
car	486	94.7%	93.7%	94.0%	98.0%	97.4%	97.7%
sky	473	94.5%	94.9%	94.9%	98.1%	97.3%	98.1%
person	453	81.0%	79.3%	79.8%	93.2%	90.2%	92.3%
fence	394	54.1%	57.8%	59.0%	87.6%	81.9%	82.9%
bicycle	392	75.0%	74.0%	74.8%	91.2%	85.4%	84.7%
traffic light	385	67.6%	65.5%	65.9%	90.5%	86.7%	86.6%
terrain	351	61.4%	63.2%	63.3%	88.0%	83.8%	89.0%
wall	339	49.1%	49.6%	49.8%	85.8%	80.4%	90.4%
rider	303	59.8%	59.7%	60.3%	84.3%	78.2%	83.7%
truck	187	70.6%	78.3%	80.2%	96.3%	90.2%	90.8%
motorcycle	178	61.4%	60.8%	63.1%	93.1%	86.3%	82.2%
bus	161	79.2%	85.8%	87.0%	92.4%	90.4%	95.8%
train	95	55.2%	70.9%	74.9%	96.2%	90.9%	86.9%

Table 1. Per-class performance on validation set following the experiment from section 5.1. The first column lists the 19 classes in the Cityscapes data set. The second column indicates the number of images in the validation set (out of 500) that contain the specified class. Classes are ordered from the most common (first row) to the rarest (last row). Columns 3 to 5 depict IoU per class. Classes in which the performance of the model trained on manual annotations + quality filtered auto-annotations surpasses the performance of the model trained on manual annotations only are highlighted in green. Those that perform the same are highlighted in yellow. Columns 6 and 7 display annotation precision (relative to the ground truth) of unfiltered and filtered auto-annotations. The last column displays filter retention rates.

# manual labels	Labeled data only	Labeled+quality filtered data
Full set (100%)	77.0%	77.6%
Small set (30%)	75.0%	77.0%
Tiny set (15%)	74.0%	76.3%

Table 2. *mIoU*s on the validation set.

Retention rates (the percentage of pixels that are retained after applying the quality filter) are indicated in the rightmost column in the table. Overall, 96.2% of the pixels are retained. These results, in combination with the results in section 5.1, show that by applying our quality filter we can improve model's performance while retaining most of the pixels (only 3.8% of the pixels are masked out).

In Figure 1, we provide a visual indication for the performance of the quality filter. This figure compares: (i) the RGB images, (ii) unfiltered auto annotations, (iii) filtered auto-annotations, and (iv) manual labels (ground truth). One can see that the quality filter identified correctly a substantial amount of errors. For example, a major part of the car's hood is masked out by the filter. The filter also correctly masked out the misclassified parts of the wall in the top and middle examples – see the purple area (second column), that is correctly filtered out (third column). In the bottom example, the misclassified area on the left car is correctly masked out by the quality filter, as well as an area on the right wall. In all three examples, additional more subtle masked out areas can be found.

6. Discussion and Conclusions

The accuracy gain due to the proposed quality filtering procedure is higher for "less experienced" teacher ensembles: the more mistakes the teacher makes, the more errors the quality filter can fix, leading to a trade-off between the requested auto-annotation quality and the amount of generated training data. While in our experiments we used a fixed quality threshold, we would like to explore the influence of the quality-quantity trade-off on the trained model accuracy.

We can further enhance the training process by iterating over the "train the teachers ensemble", "auto-annotate the unlabeled data", and "train the target model" steps. Another interesting research direction is using ensemble stacking (instead of softmax averaging) for the fusing function g and building a joint multi-task model for g and q together.

To conclude, we propose a generic method for quality prediction of automatic annotations generated by an ensemble of models. We adapt the proposed approach to semantic segmentation by doing label quality filtering at pixel level. We show that refining the auto-annotated training set by discarding data samples with low predicted label quality improves the trained model accuracy. We demonstrate that the performance of the state-of-the-art model can be achieved by training it with only a fraction (30%) of the original manually labeled data set, and replacing the rest with the auto-annotated, quality filtered labels.

References

[1] C. Bucila, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 535–541, 2006. 1

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *ECCV*, 2018. 2

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *CVPR*, 2016. 2

[4] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000. 1

[5] R. P. Duin and D. M. Tax. Experiments with classifier combining rules. In *International Workshop on Multiple Classifier Systems*, pages 16–29. Springer, 2000. 1

[6] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017. 1

[7] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, Oct 1990. 1

[8] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*, 2015. 1

[9] A. Jurek, Y. Bi, S. Wu, and C. Nugent. A survey of commonly used ensemble-based classification techniques. *The Knowledge Engineering Review*, 29(5):551–581, 2014. 1

[10] A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, volume 1, page 3, 2017. 1

[11] L. I. Kuncheva. An application of owa operators to the aggregation of multiple classification decisions. In *The ordered weighted averaging operators*, pages 330–343. Springer, 1997. 2

[12] U. Naftaly, N. Intrator, and D. Horn. Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems*, 8(3):283–296, 1997. 2

[13] I. Nigam, C. Huang, and D. Ramanan. Ensemble knowledge transfer for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1499–1508. IEEE, 2018. 1

[14] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015. 1

[15] I. Radosavovic, P. Doll'ar, R. B. Girshick, G. Gkioxari, and K. He. Data distillation: Towards omni-supervised learning. *CVPR*, 2018. 2

[16] M. Re and G. Valentini. Ensemble methods: a review. 2011. 1, 2

[17] R. Schapire. The strenght of weak learnability,(1990) machine learning 5 197-227; y. freud. *Boosting a weak learning algorithm by majority*, pages 256–285, 1995. 2

[18] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 1

[19] G. Valentini and F. Masulli. Ensembles of learning machines. In *Italian Workshop on Neural Nets*, pages 3–20. Springer, 2002. 1

[20] M. Van Breukelen, R. Duin, D. Tax, and J. Den Hartog. Combining classifiers for the recognition of handwritten digits. In *1st IAPR TC1 Workshop on Statistical Techniques in Pattern Recognition, Prague, Czech Republic*, pages 13–18, 1997. 2

[21] A. Vezhnevets and J. M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. 2010. 1

[22] D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. 1, 2

[23] X. Zhan, Z. Liu, P. Luo, X. Tang, and C. C. Loy. Mix-and-match tuning for self-supervised semantic segmentation. *arXiv preprint arXiv:1712.00661*, 2017. 1

[24] Y. Zhou and S. Goldman. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence*, pages 594–602, Nov 2004. 1

[25] Z.-H. Zhou. When semi-supervised learning meets ensemble learning. In *International Workshop on Multiple Classifier Systems*, pages 529–538. Springer, 2009. 1

[26] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4, 2006. 1