Teacher-Student Learning Paradigm for Tri-training: An Efficient Method for Unlabeled Data Exploitation

Yash Bhalgat
Qualcomm AI Research
yashbhalgat95@gmail.com

Zhe Liu, Pritam Gundecha, Jalal Mahmud, Amita Misra
IBM-Research, Almaden
{liuzh, psgundec, jumahmud, amita.misral}@us.ibm.com

Abstract

Given that labeled data is expensive to obtain in real-world scenarios, many semi-supervised algorithms have explored the task of exploitation of unlabeled data. Traditional tri-training algorithm and tri-training with disagreement have shown promise in tasks where labeled data is limited. In this work, we introduce a new paradigm for tri-training, mimicking the real world teacher-student learning process. We show that the adaptive teacher-student thresholds used in the proposed method provide more control over the learning process with higher label quality. We perform evaluation on SemEval sentiment analysis task and provide comprehensive comparisons over experimental settings containing varied labeled versus unlabeled data rates. Experimental results show that our method outperforms other strong semi-supervised baselines, while requiring less number of labeled training samples.

1 Introduction

Machine learning algorithms often require large amount of labeled data for training. As collecting labeled examples can be expensive, semi-supervised learning has been proposed (Zhu, 2006). Among the existing semi-supervised approaches, self-training (Triguero et al., 2015), co-training (Blum and Mitchell, 1998), and tri-training (Zhou and Li, 2005) are the most notable ones. However, they suffer from one major issue of the gradually increased level of noise during the iterative labeling process. This problem can be attributed to two factors: (1) static labeling threshold, and (2) inappropriate stopping criteria.

Many self-labeled algorithms iteratively enlarge labeled training set with unlabeled instances whose prediction confidence is larger than a static labeling threshold. Static labeling threshold produces a good classification performance only when the proportion of correctly labeled instances remains above a constant level. However, given the continuously added noisy labels during the semi-supervised process (Triguero et al., 2015), it is unlikely that any fixed assignment of the threshold will produce optimal classifications.

Besides, deciding when to stop the iterative instance labeling process is also critical for the self-labeled techniques. Existing stopping criteria include: setting a threshold on the number of labels that the algorithm is willing to generate, or stopping the labeling process when little to no accuracy increase occurs in an iteration. Stopping criteria is still an open issue, as too conservative or too liberal stopping criteria may produce many mislabeled examples to the self-labeled process.

To solve the two challenges, we propose a new tri-training-based method, called tri-training with teacher-student paradigm. Specifically, in each iteration, a double-teacher-single-student teaching relation is established based on predefined teacher and student thresholds, where teachers teach the student with generated proxy labels on the unlabelled data. Along the teaching process, the teacher-student relationship is continuously adjusted with adaptive teacher and student thresholds. The teacher-student relationship terminates on either running out of teachable instances or when reaching a graduation point, where the student threshold equals the teacher threshold.

We evaluate the tri-training with teacher-student paradigm approach on the sentiment analysis task of SemEval-2016 over various labeled-unlabeled data ratios. The proposed method outperforms many strong baselines in terms of gaining better prediction performances while consuming less number of unlabeled examples.
2 Method

Assume we are given a set of unlabeled samples \( U \) as well as a set of labeled samples \( L \), where \( L \ll U \). The proposed method starts by training three independent base classifiers \( m_i, m_j, m_k \) on bootstrapped sample subsets \( S_i, S_j, S_k \) respectively taken from \( L \). The aim of the bootstrap sampling is to increase the diversity of base classifiers trained through the labeled set. Next, for every sample \( x \) in \( U \), each of the trained models \( m_i, m_j, m_k \) predicts a label \( c_i, c_j, c_k \) with corresponding prediction probability \( p_i(c_i|x), p_j(c_j|x), p_k(c_k|x) \).

2.1 Teacher-Student Assignment

Instead of assigning \( x \) a majority voted label, as implemented in the original tri-training (Zhou and Li, 2005), here we model the learning task from a teacher-student perspective. In each iteration of our proposed approach, two classifiers (\( m_j \) and \( m_k \)) are ascertained to be teachers if their prediction probabilities \( p_j(c_j|x) \) and \( p_k(c_k|x) \) are both larger than the teacher threshold \( \tau_t \). The other classifier \( m_i \) is then treated as student if its prediction probability is less than the student threshold \( \tau_s \). An unlabeled sample \( x \) in \( U \) will only be assigned a label after it is identified as teachable. Teachable examples are defined according to the function SelectTeachableSamples, as shown in Algorithm 2. The required criteria are as follows: Firstly, the predicted labels \( c_j \) and \( c_k \) from the two teachers \( m_j \) and \( m_k \) must agree with each other. Second, both teachers’ prediction confidences \( p_j \) and \( p_k \) must exceed \( \tau_t \) and at the same time, the student’s confidence \( p_i \) must be less than \( \tau_s \). This setting of using two teachers ensures that bias in any of these models doesn’t affect the quality of the information taught to the student. It’s similar to the real-life teacher-student learning process, where only qualified teachers can teach students things that they are the most comfortable with. Here, it is important to note that the teacher-student roles are rotated in each iteration, \( i \in \{1, 2, 3\}, (j, k \neq i) \), allowing each classifier to learn from the other classifiers’ experiences, as \( m_i \) is further trained with the original labeled set \( L \) along with the identified teachable samples \( L_i \).

2.2 Adaptive Thresholds

Another novel aspect that we adopt from real-world teaching scenarios is the proposed method is the continuously adjusted teacher-student relationship. To be more specific, as a student learns from the teachers, it would become more confident of its prior knowledge taught by the teachers. In that sense, the student threshold \( \tau_s \) increases monotonically in every iteration. On the other hand, as student progresses through the learning process, the teachers are supposed to teach them more advanced cases, i.e. cases where the teachers are less confident about. This is captured in our approach by monotonically decreasing the teacher threshold \( \tau_t \). For this work, we chose a linear adaptive rate for the adaptive process as shown in line 10 and 11 of Algorithm 1.

2.3 Stopping Criteria

Existing self-labeled techniques often stop when no sample can be labeled, or no performance improvement occurs in an iteration. The original tri-training paper introduces an error constraint that checks if a peak performance has been reached. However, the error measurement is conducted only on the labeled dataset, hence assuming that the labeled set distribution is representative of the unlabeled set distribution. Tri-training may also lead to a limited number of co-labeling examples for training and

---

Algorithm 1 Teacher Student Tri-training

**Require:** \( L \) - set of labeled samples, \( U \) - set of unlabeled samples, \( m_{i,j,k} \) - teacher-student models, \( \tau_s \) - teacher threshold, \( \tau_t \) - student threshold, \( \lambda_t \), \( \lambda_s \) - teacher-student adaptive rates

1: for \( i \in \{1,3\} \) do
2: \( S_i \leftarrow \text{bootstrap sample}(L) \)
3: \( m_i \leftarrow \text{train model}(S_i) \)
4: end for
5: while \( \tau_t \leq \tau_s \) do
6: for \( i \in \{1,3\} \) do
7: \( L_i \leftarrow \text{SelectTeachableSamples}(U, m_{i,j,k}, \tau_t, \tau_s) \)
8: \( m_i \leftarrow \text{train model}(L \cup L_i) \)
9: end for
10: \( \tau_t \leftarrow \tau_t - \lambda_t \)
11: \( \tau_s \leftarrow \tau_s + \lambda_s \)
12: end while
13: apply majority vote over \( m_{i,j,k} \)

Algorithm 2 Select Teachable Samples

**Require:** \( U \) - set of unlabeled samples, \( \tau_s \) - teacher threshold, \( \tau_t \) - student threshold, \( m_i \) - student model, \( m_{i,j,k} \) - teacher models

1: \( \pi \leftarrow \emptyset \)
2: for all \( x \in U \) do
3: if \( c_j = c_k \) then
4: \( tcf = \min(p_j(c_j|x), p_k(c_k|x)) \)
5: \( scf = p_i(c_i|x) \)
6: if \( tcf > \tau_t \) & \( scf < \tau_s \) then
7: \( \pi \leftarrow \pi \cup \{(x, c_j(x))\} \)
8: end if
9: end if
10: end for
11: return \( \pi \)
a premature termination while dealing with large datasets (Chou et al., 2016).

In this work, we present our stopping criterion by comparing the student’s confidence threshold with the teacher’s threshold during each training iteration. We assume that when a student reaches the same confidence level as the teachers in a particular iteration, then there is nothing to be learned for the students from the teachers. This happens in our algorithm 2, when \( \tau_s \geq \tau_t \). At this point, adding newer samples to the training set of \( m_i \) (the student) would not contribute to its learning anymore. In that sense, we called the point when \( \tau_s \geq \tau_t \) as the graduation point, so as to stop the tri-training process naturally when the constraint is reached.

3 Evaluation

3.1 Experimental Settings

Datasets. We evaluate our model on the sentiment classification dataset of SemEval-2016 Task 4 Subtask A (Nakov et al., 2016). In total, there are 6000 training sentences, including 3094 positive, 863 neutral, and 2043 negative instances. We use 2000 sentences from the dev set for validation and we have 20632 for test. To test the model’s generalizability, we subsequently examine it under different proportions of labeled data. We select 10%, 20%, 30% and 40% of the training set randomly as labeled samples \( L \) and treat the rest as unlabeled \( U \) by hiding their labels. Hidden labels are used later for quality check of the generated proxy labels.

Baselines. Since our method improves upon the foundations laid by the typical semi-supervised methods as mentioned in the related work section (e.g. tri-training and self-training), we compare with the following baselines:

1. \textit{NB STr} - Self-training with Naive Bayes as base classifier.
2. \textit{SVM STr} - Self-training with SVM as base classifier.
3. \textit{MLP STr} - Self-training with neural networks (multilayer perceptrons) as base classifier.
4. \textit{Tri} - Tri-training with SVM as base classifiers.
5. \textit{Tri-D} - Tri-training with disagreement with SVM as base classifiers (Søgaard, 2010).

Our proposed approach is tri-training with teacher-student paradigm (Tri-TS). We don’t compare with co-training here because there are no clear independent views (Zhou and Li, 2005) in the sentiment analysis task. We do not use any deep learning model as base learner in this study, as deep learning models may not perform well in the presence of limited labeled data. We did try FastText (Joulin et al., 2017) as a proof case, but even under the 40% label rate, its performance is unsatisfactory (an initial \( F_{1PN} \) of 0.346 with an improvement of +0.034 using the proposed model).

In all the baselines, we experiment with different base classifiers and their combinations, namely Naive Bayes, SVM and Neural Networks. We use a linear kernel (LinearSVC) for SVM. For the neural networks (MLP), we use 50 neurons in the hidden layer with a softmax output. We use Glove 300-dimensional word embeddings pennington2014glove. After text-cleaning and tokenization, we average the word-embeddings for the tokens present in the sentence to get the feature vectors. For both the tri-training baselines, Tri and Tri-D, we obtain the best results with SVM as base classifiers. Hence, we report these for comparison with our approach.

Note that, as mentioned in Section 2.3, for the baselines Tri and Tri-D, we use their own respective stopping criteria during evaluation, as a comparison to our newly proposed stopping criterion.

Parameter Tuning. All parameters required in both the proposed method and the baselines are fine-tuned using the validation set. A grid search is used to determine those parameter values that maximize each model’s performance. For the proposed method \( \tau_s \) is tuned \( \in [0.7, 1.0] \), \( \tau_t \in [0.6, 0.95] \). The best performed rates of \( \lambda_t \) and \( \lambda_u \) are found empirically as 0.001. For the tri-training baselines, we try to tune the error constraint as suggested in the original paper, but it generates only small number of proxy labels during the training process and terminates after very limited number of iterations. In that sense, we discard the error constraint and try the threshold based tri-training method as adopted in (Ruder et al., 2017) and (Søgaard, 2010). Best performed parameters are obtained again via evaluations on validation set.

3.2 Results

We evaluate our approach and the baselines from three different aspects: the overall model performance, the quality of generated proxy-labels, and the quantity of unlabeled data consumed. Model performances are reported using \( F_{1PN} \)-score as
Table 1: $F^P_{1N}$ comparison averaged over 5 runs for different proportions of labeled data.

Table 2: Percentage of matches between the produced proxy-labels and the ground truth averaged over 5 runs for different proportions of labeled data.
cases in the very beginning and gradually increases
the difficulty along the learning process. On the
contrary, the original tri-training grows very fast
but also plateaus earlier, hence not achieving the
full potential of using the three base classifiers.
This early plateauing is avoided in our case with
the adoption of the adaptive thresholds.

**Sensitivity Analysis.** We further perform sensi-
tivity analysis for the assessment of the initial set-
tings of $\tau_t$ and $\tau_s$ with respect to their impact on
the model performance. Specifically, we compare
the experiment results with: (1) the initial teacher
threshold $\tau_t$ set over $[0.7, 1.0]$ with initial $\tau_s$
fixed as 0.85; and (2) the initial student threshold $\tau_s$
set over $[0.6, 0.95]$ with initial $\tau_t$ fixed as 0.94. In both
settings, $\tau_t$ and $\tau_s$ are continuously updated with the
learned adaptive rates $\lambda_t$ and $\lambda_s$ after their initial as-
signment. We observe only marginal performance
losses with an average difference of $-0.015 F^P_{i1}$
over all values. This indicates that the initial value
for $\tau_t$ and $\tau_s$ would not affect the performance that
much, as long as they are adaptive.

4 Conclusion

In this paper, we propose a new teacher-student
paradigm for original tri-training with continu-
ously adaptive threshold and a natural stopping
criteria. We show that our model outperforms all
self-training and tri-training baselines in terms of
achieving higher overall performance, higher qual-
ity of generated proxy labels, while consuming
a less quantity of the unlabeled data. Although
we only validate the proposed method against the
benchmark SemEval dataset in this paper, our ul-
timate goal is to utilize it as a solution for the sce-
narios with limited labeled data and to tackle real-
world problems, where labeled data is hard to find
or expensive to attain.

**References**

Avrim Blum and Tom Mitchell. 1998. Combining la-
beled and unlabeled data with co-training. In Pro-
cedings of the eleventh annual conference on Com-
putational learning theory, pages 92–100. ACM.

Chien-Lung Chou, Chia-Hui Chang, and Ya-Yun
Huang. 2016. Boosted web named entity recogni-
tion via tri-training. *ACM Transactions on Asian
and Low-Resource Language Information Process-
ing (TALLIP)*, 16(2):10.

Armand Joulin, Edouard Grave, Piotr Bojanowski,
and Tomas Mikolov. 2017. Bag of tricks for efficient
text classification. In *Proceedings of the 15th Con-
ference of the European Chapter of the Association
for Computational Linguistics: Volume 2, Short Pa-
pers*, pages 427–431. Association for Computational
Linguistics, April.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio
Sebastiani, and Veselin Stoyanov. 2016. Semeval-
2016 task 4: Sentiment analysis in twitter. In *Pro-
cedings of the 10th international workshop on se-
metric evaluation (semeval-2016)*, pages 1–18.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin.
2017. Knowledge adaptation: Teaching to adapt.
arXiv preprint arXiv:1702.02052.

Anders Søgaard. 2010. Simple semi-supervised train-
ing of part-of-speech taggers. In *Proceedings of the
ACL 2010 Conference Short Papers*, pages 205–208.
Association for Computational Linguistics.

Isaac Triguero, Salvador Garcia, and Francisco Herr-
era. 2015. Self-labeled techniques for semi-
supervised learning: taxonomy, software and em-
pirical study. *Knowledge and Information Systems,*
42(2):245–284.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Ex-
ploring unlabeled data using three classifiers. *IEEE
Transactions on knowledge and Data Engineering*,
17(11):1529–1541.

Xiaojin Zhu. 2006. Semi-supervised learning lit-
erature survey. *Computer Science, University of
Wisconsin-Madison*, 2(3):4.