TrustAL: Trustworthy Active Learning using Knowledge Distillation

Beong-woo Kwak\textsuperscript{1}, Youngwook Kim\textsuperscript{2}, Yu Jin Kim\textsuperscript{1}, Seung-won Hwang\textsuperscript{3}, Jinyoung Yeo\textsuperscript{1*}
\textsuperscript{1} Department of Artificial Intelligence, Yonsei University
\textsuperscript{2} Department of Computer Science, Yonsei University
\textsuperscript{3} Department of Computer Science and Engineering, Seoul National University
\{beongwoo.kwak, youngwook, yujin000731\}@yonsei.ac.kr
seungwonh@snu.ac.kr
jinyeo@yonsei.ac.kr

Abstract

Active learning can be defined as iterations of data labeling, model training, and data acquisition, until sufficient labels are acquired. A traditional view of data acquisition is that, through iterations, knowledge from human labels and models is implicitly distilled to monotonically increase the accuracy and label consistency. Under this assumption, the most recently trained model is a good surrogate for the current labeled data, from which data acquisition is requested based on uncertainty/diversity. Our contribution is debunking this myth and proposing a new objective for distillation. First, we found example forgetting, which indicates the loss of knowledge learned across iterations. Second, for this reason, the last model is no longer the best teacher. For mitigating such forgotten knowledge, we select one of its predecessor models as a teacher, by our proposed notion of “consistency”. We show that this novel distillation is distinctive in the following three aspects; First, consistency ensures to avoid forgetting labels. Second, consistency improves both uncertainty/diversity of labeled data. Lastly, consistency redeems defective labels produced by human annotators.

Introduction

Labeling data is a fundamental bottleneck in machine learning due to annotation cost and time. One practical solution is Active Learning (AL): given a limited labeling budget \( k \), which example should I ask human annotators to label? Generally, this can be done through an iterative process of labeling data, model training, and data acquisition steps, until sufficient labels are obtained. At each iteration, based on the last trained model, unlabeled yet the \( k \) most desirable examples are recognized and added to the labeled dataset to train a new model. This process continues to the next iteration for selecting next \( k \) unlabeled examples based on the newly trained model. That is, a naive belief in AL is that the last trained model can be a good reference or surrogate for the distribution of the currently labeled data, which indicates what examples are desired for the next model update.

In this work, our empirical observation dispels this myth. Although the model knowledge learned from the labels is expected to be “consistently” kept or improved across iterations, we find that knowledge learned at some time is suddenly forgotten, which indicates that the recent model is ineligible to be treated as a good reference of the labeled dataset. More specifically, we can observe such inconsistent behaviors of the trained model during inference time, where sample \( t \) predicted correctly at iteration \( t \) is predicted incorrectly at iteration \( t + \Delta t \), which is called example forgetting [Toneva et al. 2019].

Motivated by this, in this work, we argue that correct-consistency (which we call consistency for brevity) should be an essential criterion, which is the model ability to make consistent correct predictions across successive AL generations for the same input [Wang et al. 2020]. In the view of consistency, prior AL methods only focusing on data acquisition steps (Dasgupta 2011; Xu et al. 2003; Bodó, Minier, and Csató 2011; Ash et al. 2019) are still sub-optimal since the three transitions among AL steps may suffer from following problems due to inconsistency (reverse phenomenon of consistency), which we empirically analyze later:

- **From labeling to model training**: Despite successful data acquisition, the subsequent labeling efforts can be negated by forgetting the learned knowledge later, which wastes annotation cost. We argue that consistency is key to make a label-efficient AL (Figure 3).
- **From model training to data acquisition**: Inconsistent data acquisition models cannot serve as a good reference for the current data distribution, which leads to contaminating the next data acquisition step. Improving consistency may be synergetic to either uncertainty- or diversity-based acquisition strategies (Figure 5).
- **From data acquisition to labeling**: Human annotators who act as oracles are usually subject to accidental mislabeling (Bouguela et al. 2018) which degrades traditional AL methods. Learning to keep consistency enables to mitigate the confusion from the noisy labels (Figure 6).

To overcome these drawbacks and thus make all the three transitions in AL more trustworthy, we propose a label-efficient AL framework, called Trustworthy AL (TrustAL), for bridging the knowledge discrepancy between labeled data and model. In TrustAL, our key idea is to add a new step in the iterative process of AL to learn the forgotten knowledge, which is orthogonally applicable to state-of-the-art data acquisition strategies in a synergistic manner. Specifi-
cally, at each iteration, TrustAL first searches for an expert model for the forgotten knowledge among the predecessor models. Then, TrustAL transfers the model knowledge (e.g., logits) to the current model training step by leveraging the knowledge distillation technique [Hinton, Vinyals, and Dean 2015]. By optimizing the dual goals of following both human labels and machine labels of the expert predecessor, the newly trained model can relieve forgotten knowledge and thus be more consistent, keeping its correct predictions.

For the purpose of identifying which predecessor is the most desired teacher to relieve the forgotten knowledge, we further explore the teacher selection problem. To resolve this, we present two teacher selection strategies, (1) TrustAL-MC: monotonicty choice of the most recent model (i.e., a proxy of the most accurate model), and (2) TrustAL-NC: non-monotonic choice of the well-balanced model with accuracy and consistency, which we thoroughly design as analysis/evaluation measures in this paper.

Our experiments show that the TrustAL framework significantly improves performance with various data acquisition strategies while preserving the valuable knowledge from the labeled dataset. We validate the pseudo labels from the predecessor models are not just approximate/weak predictions - it can be viewed as knowledge from the previous generation, and can be used as consistency regularization for conventional AL methods solely aiming at higher accuracy.

**Preliminaries & Related Work**

**Active Learning for Classification**

Given an arbitrary classification task, assume that there is a large unlabeled dataset \( \mathcal{U} = \{x_i\}_{i=1}^n \) of \( n \) data samples. The goal of AL is to sample a subset \( \mathcal{Q} \subset \mathcal{U} \) to efficiently label so that newly training a deep neural network parameters \( \theta \) for the classifier \( f(x; \theta) \) improves test accuracy. Algorithm 1 describes the conventional procedure in AL. On each iteration \( t \), the learner uses strategy \( \mathcal{A} \) (e.g., uncertainty or diversity) to acquire \( k \) samples \( \mathcal{Q}_t \) from dataset \( \mathcal{U} \). Generally, data acquisition model \( M_t \) is used for evaluating unlabeled samples according to \( \mathcal{A} \). Then, for sample \( x_i \), the learner queries for its oracle label \( y_i \), where \( c \) is the number of classes. We denote the predicted label of trained model \( \theta_t \) for \( x_i \) by \( \hat{y}_i = \arg \max_{y \in c} f(y|x_i; \theta_t) \).

In most AL approaches, a data acquisition model at time \( t \) corresponds to the trained classification model at time \( t-1 \), i.e., \( M_t = \theta_{t-1} \). We call this monotonicty acquisition, since a naive belief would be assuming the last trained model \( \theta_{t-1} \) is effective enough to not only provide a good representation for the entire labeled data \( \mathcal{L} \) but also estimate acquisition factors (e.g., confidence) for remaining unlabeled data \( \mathcal{U} \).

**Data Acquisition Strategies in AL**

The ultimate goal of AL is to improve the classification accuracy with a fixed annotation budget [Settles 2009; Lowell, Lipton, and Wallace 2018]. Existing research efforts on pool based active learning [Lewis and Gale 1994] achieve this goal by focusing on data acquisition based on query strategy and data strategy [Ren et al. 2020]. As a query strategy, there are two general approaches to recognize the most appropriate samples [Dasgupta 2011] with monotonic acquisition: uncertainty sampling and diversity sampling. While uncertainty sampling efficiently searches the hypothesis space by finding difficult examples to label [Asghar et al. 2017; He et al. 2019] [Ranganathan et al. 2017], diversity sampling exploits heterogeneity in the feature space [Hu, Mac Namee, and Delany 2010; Bodó, Minier, and Csato 2011]. Recently, hybrid approaches are proposed [Zhdanov 2019; Ash et al. 2019]. Particularly, BADGE [Ash et al. 2019] successfully integrates both aspect by clustering hallucinated gradient vectors based on monotonic acquisition scheme.

**Data Acquisition Models in AL**

Despite the remarkable success in query strategies, recent research has concerned several limitation of AL. Yun, Kim, and Kim (2020); Wang et al. (2016) point out the difficulty of learning good representation across AL iterations since insufficient annotations may lead to the instability of training models. This indicates that the monotonic acquisition does not ensure the last trained model as a good surrogate of the currently labeled data to identify the informative samples for data acquisition. As a result, [Karamcheti et al. 2021; Farquhar, Gal, and Rainforth 2020; Prabhu, Dogan, and Singh 2019] reveal that the acquired samples are vulnerable to sampling bias, and especially, [Karamcheti et al. 2021] have presented Dataset Maps [Swayamdipta et al. 2020] of AL, which visualizes harmful outliers preferred by AL methods. Despite these facts, Lowell, Lipton, and Wallace (2018) suggest that the monotonic acquisition is still promising as another remedy using external models (e.g., SVM out of the AL iterations) for data acquisition extremely hampers accuracy of AL.

Motivated by this line of research, in this work, we explore how the limitation of monotonic acquisition can be addressed, in particular, considering consistency as a solution to mitigate the instability of AL iterations. Similar to Lowell, Lipton, and Wallace (2018) reporting unreliable performance of AL in the NLP field, we choose text classification tasks as our testbed.

---

**Algorithm 1**: Conventional AL procedure

**Input:** Initial labeled data pool \( \mathcal{L} \), unlabeled data pool \( \mathcal{U} \), number of queries per iteration (budget) \( k \), number of iterations \( T \), sampling algorithm \( \mathcal{A} \)

**Output:** Model parameters \( \theta_T \)

\( \theta_0 \leftarrow \text{Train a seed model on } \mathcal{L} \)

for \( t = 1, \ldots, T \) do
  \( M_t(x) = f(x; \theta_{t-1}) \)
  \( \mathcal{Q}_t \leftarrow \text{Apply } \mathcal{A}(x, M_t, k) \) for \( \forall x \in \mathcal{U} \)
  \( \hat{y}_t \leftarrow \text{Label queries } \mathcal{Q}_t \) by oracles
  \( \mathcal{L} \leftarrow \mathcal{L} \cup \hat{y}_t \)
  \( \mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{Q}_t \)
  \( \theta_t \leftarrow \text{Train a new model on } \mathcal{L} \)

end

return \( \theta_T \)
Accuracy-Consistency Dynamics

In this section, we analyze the training dynamics of AL in terms of consistency along with accuracy, observing (catastrophic) example forgetting event (Toneva et al. 2019) on the AL iterations: these occur when examples that have been “learnt” (i.e., correctly classified) at some time \( t \) in the optimization process are subsequently misclassified – or in other terms “forgotten” – at a time \( t+\Delta t > t \). Formally,

**Definition 1** (Forgetting and Learning Events) Given a predicted label \( \hat{y}_t \), let \( \text{acc}_t^i = \mathbb{I}_{\hat{y}_t^i = y_t} \) be a binary variable indicating whether the example is correctly classified by \( \theta_t \). Sample \( i \) undergoes a forgetting event when \( \text{acc}_t^i \) decreases between two different iterations, i.e., \( \text{acc}_t^i > \text{acc}_{t+\Delta t}^i \), where \( \Delta t > 0 \) (misclassified). Conversely, a learning event has occurred if \( \text{acc}_t^i < \text{acc}_{t+\Delta t}^i \).

While learning new knowledge is also one of the important factors for generalization ability, our focus is on measuring how well models in AL preserve the learned knowledge. For further analysis, we introduce Correct Inconsistency of a model as a measure of how inconsistent the target model is with its predecessor models for a sample. That is, correct inconsistency counts the forgetting events between the model and each of the predecessor models.

**Definition 2** (Correct Inconsistency) The degree of correct inconsistency of \( \theta_t \) for sample \( x_i \) is measured as the number of occurrences of forgetting events for sample \( x_i \) from any predecessor model \( \theta_{t-\Delta t} \), where \( 0 < \Delta t \leq t \):

\[
C_i^{(t)} = \sum_{\Delta t=1}^{t} \mathbb{I}_{(\text{acc}_{t-\Delta t}^i > \text{acc}_{t-\Delta t}^i)}
\]

As the number of predecessor models are different per AL iteration, to fairly show the degree of inconsistency, we use mean of correct inconsistency for every sample in development split, e.g., \( \text{MCI} = \frac{1}{n} \sum_{i=1}^{n} C_i^{(t)}/t \).

In Figure 1 we present both accuracy and MCI of trained models through the full progress of AL. We analyze three data acquisition strategies that are carefully chosen by considering the uncertainty-diversity dichotomy (Yuan, Lin, and Boyd-Graber 2020) along with random strategy. CONF (Wang and Shang 2014), CORESET (Sener and Savarese 2018), and BADGE (Ash et al. 2019) represent three lines of acquisition strategies in AL: uncertainty, diversity, and their hybrid. Across all dataset and acquisition strategies, accuracy and MCI follow the anti-correlated relationship. For convenience of analysis, we divide the training progress into two phases based on the transition of tendency in terms of accuracy: stable and saturated phases.

In the stable phase, more data leads to more accurate model. Validation accuracy increases on 0-50% of TREC and 0-40% of SST-2, while MCI decreases, where newly labeled samples improve generalization of the trained models. In this phase, AL strategies seek to achieve label efficiency, reaching higher accuracy with a given amount of labeled samples or reversely achieving the same accuracy with less amount of labeled samples. What is notable here is that dramatic improvement of accuracy mostly involves the rapid drop in MCI. These analytical results provide a guide towards the idealistic property of AL methods, which is preserving existing knowledge and simultaneously learning new knowledge. Thus, we hypothesize that relieving forgetting events may contribute to faster and better (i.e., higher accuracy) convergence of AL.

In the saturated phase, the monotonic trends observed in stable phase do not hold. Validation accuracy converges or decreases with the rapid increase of MCI, suggesting that the generalization performance of model deteriorates as even more labeled samples are fed into the trained models.
That is, more data does not always lead to more accurate models, which indicates labeling efforts may be negated in this phase. Although such an extremely undesirable situation in AL is barely addressed by stopping AL iterations in prior work (Ishibashi and Hino 2020), an idealistic AL framework would avoid this phase so that the models can be learned in a more label-efficient manner.

TrustAL: Trustworthy Active Learning

Based on the prior findings on training dynamics of AL procedures, we aim to relieve forgotten knowledge to train better acquisition models that serve as a good surrogate for labeled data. A naive way to obtain more generalized models is simply using enough labeled data. However, this approach is not always applicable since budget is limited in AL. Another line of work is using multiple equivalent models (e.g., ensemble) based on complementary nature across different generations. However, this approach is also not always affordable since querying on the huge pool of unlabeled data using multiple models is computationally too expensive.

We now present TrustAL (Trustworthy AL) that enables the training of consistent acquisition model that serves as a good reference for labeled dataset in smart and resource-efficient way. TrustAL utilizes additional machine generated labels for the purpose of mitigating the forgotten knowledge. Especially, among predecessor models, TrustAL identifies a proper expert model that can efficiently contribute to mitigating forgotten samples, which is a novel way of tackling the possible knowledge loss during AL procedure.

Distillation-based Consistency Regularization

Knowledge distillation (Hinton, Vinyals, and Dean 2015) is originally proposed to transfer knowledge from one model (i.e., teacher model) to another (i.e., student model) to compress the size of model. Inspired by recent approaches to transfer knowledge between equivalent models (Furlanello et al. 2018), we propose using the inferior (e.g., less accurate) predecessor model as a teacher model to mitigate example forgetting of the student model (i.e., last trained, superior model) by learning from pseudo labels (i.e., logits). This distillation method can be interpreted as a type of consistency regularizer to alleviate forgotten knowledge.

Formally, Algorithm 2 describes the overall procedure of the TrustAL framework on the AL iterations. When given a labeled data pool \( \mathcal{L} = \{x_i, y_i\}_{i=1}^n \) at the \( t \)-th iteration, let \( L_{CE} \) be a typical cross-entropy loss with oracle labeled examples, i.e., \( \sum_{(x_i, y_i) \in \mathcal{L}} \text{CrossEntropy}(y_i, f(x_i; \theta_t)) \), and let \( L_{KL} \) be a knowledge distillation loss with the pseudo labels of a predecessor model from \( t-\Delta t \), i.e., \( \sum_{(x_i, y_i) \in \mathcal{L}} KL-Divergence(f(x_i; \theta_{t-\Delta t}), f(x_i; \theta_t)) \). On the top of an arbitrary data acquisition method (e.g., CORESET and BADGE), model parameter \( \theta_t \) produced by TrustAL is obtained by the following optimization:

\[
\theta_t = \text{argmin}_{\theta_t} L_{CE}(\theta_t) + \alpha \cdot L_{KL}(\theta_{t-\Delta t}; \theta_t) \tag{1}
\]

where \( \alpha \) is a preference weight. We empirically analyze the effect of varying \( \alpha \) in Appendix D.
sition but also for teacher selection, synchronizing both, i.e., always $\theta_{t-\Delta t} = \theta_{t-1} = M_t$. This allows to iteratively transfer the learned knowledge generation by generation.

**Non-monotonic Consistency (TrustAL-NC)** *Correct Inconsistency* (Definition 2) can be a strong signal to indicate which sample is forgettable for the current acquisition model. Using such sample level inconsistency, we aim at cate which sample is forgettable for the current acquisition (Definition 2) can be a strong signal to indicate inconsistency

Specifically, given a development dataset $D_{dev}$ with $m$ samples, let $C^t$ be a vector of correct inconsistency values of $M_t$ ($= \theta_{t-1}$) at the $t$-th iteration for all $m$ samples, i.e., $\langle c_1(t-1), ..., c_m(t-1) \rangle \in \mathbb{R}^m$. For the purpose of using this vector as importance weights for samples, we normalize $C^t$ into $\tilde{C}^t$ where $\sum_{i} \tilde{C}^t_i = 1$ by a softmax function. We note that the sample $x_i$ with high importance weight $\tilde{C}^t_i$ means easily forgettable sample for $M_t$. Based on such consistency-aware sample importance, we define a function $g(\theta_{t-\Delta t}, M_t)$ of measuring how reliably a predecessor model $\theta_{t-\Delta t}$ can be a synergetic teacher with the data acquisition of $M_t$, by an weighted accuracy as:

$$g(\theta_{t-\Delta t}, M_t) = \tilde{C}^t \top (acc_1^{t-\Delta t}, ..., acc_m^{t-\Delta t}) / m \quad (2)$$

High $g(\theta_{t-\Delta t}, M_t)$ implies that the teacher model $\theta_{t-\Delta t}$ tends to have the knowledge of forgettable examples for the current data acquisition model $M_t$, and vice versa. Based on this, we can select a predecessor having the maximum value, as a teacher model to teach a new model $\theta_t$:

$$\text{argmax}_{1 < \Delta t \leq t} g(\theta_{t-\Delta t}, M_t) \quad (3)$$

### Development Set Strategies

One of the plausible tools to estimate the learning status of AL generations is development set as it is often used for validation process. In fact, TrustAL-NC catches forgetting signals as a by-product of the validation process. The experiment on the robustness of TrustAL-NC on the size of development set shows marginal performance decrease even when halving development set size, which resolve concerns about keeping development set in label-scarce situation. Full empirical results are presented in Appendix D.

### Experiments

#### Experimental Setup

**Dataset** We use three text classification datasets, TREC [Roth et al. 2002], Movie review [Pang and Lee 2005] and SST-2 [Socher et al. 2013], which are widely used in AL (Lowell, Lipton, and Wallace 2018; Siddhant and Lipton 2018; Yuan, Lin, and Boyd-Graber 2020) and statistically diverse. The data statistics are presented in Appendix A for more details.

**Baselines** As TrustAL is orthogonally applicable to any data acquisition strategy, for the purpose of better analysis, we use the following three acquisition methods as baselines.

- **CONF** [Wang and Shang 2014]: An uncertainty-based method that selects samples with least confidence.
- **CORESET** [Sener and Savarese 2018]: A diversity-based method that selects coreset of remaining samples.
- **BADGE** [Ash et al. 2019]: A hybrid method that selects samples considering both uncertainty and diversity.

More details on the baselines are discussed in Appendix B.
baselines. To facilitate the comparison of label efficiency of TrustAL and a baseline, we draw the horizontal reference line where the baseline starts to show convergence in Figure 3 (c). As a result, we find that TrustAL-MC and TrustAL-NC require only 40% and 30% of the training data pool, respectively, while baselines require 50% of total training data to reach the same level of accuracy. This result suggests that keeping consistency of model in AL is an essential criterion, and TrustAL successfully satisfies the ultimate goal of AL, i.e., improving the label efficiency.

Further, TrustAL-NC performs comparably to ensemble based distillation method (Fukuda et al. 2017) which aims to distill the ensemble (i.e. averaged) probability distribution of multiple models. This indicates TrustAL-NC selects a teacher model that can effectively relieve forgotten knowledge, even without using all predecessor models. Figure 4 visualizes the behavior of teacher selection by TrustAL. The figure shows that TrustAL-MC selects the most recent model as its definition and TrustAL-NC chooses the teacher model based on the consistency guidance. While preferring the more generalized teacher models from the end of the stable learning stages (16-20) rather than earlier stages, TrustAL-NC also selects earlier generation that might be inferior but professional in terms of forgotten knowledge. That is, TrustAL-NC can select complementary models for forgotten knowledge in an automatic manner.

Data Acquisition Quality (RQ2) Having tested for the overall accuracy and MCI of TrustAL, we evaluate the quality of data acquisition results when using TrustAL. To discuss how TrustAL affects data acquisition, we analyze TrustAL-NC based on the two distinctive strategies on data acquisition: uncertainty and diversity. Note that, we choose to compare acquisition quality of stable phase only since the label efficiency of saturated phase is negative for traditional
Table 1: Correct consistency of TrustAL-NC with (A) CONF (B) CORESET and (C) BADGE.

| Noise Ratio | TREC | Movie review | SST-2 |
|-------------|------|--------------|-------|
| A 7%        | 0.726| 0.637        | 0.686 |
|            | 0.770| 0.669        | 0.727 |
| TrustAL-MC  | 0.743| 0.654        | 0.705 |
| TrustAL-NC  | 0.774| 0.676        | 0.730 |
| B 15%       | 0.727| 0.627        | 0.697 |
|            | 0.777| 0.658        | 0.724 |
| TrustAL-MC  | 0.753| 0.654        | 0.707 |
| TrustAL-NC  | 0.785| 0.665        | 0.724 |
| C 7%        | 0.735| 0.636        | 0.681 |
|            | 0.773| 0.668        | 0.722 |
| TrustAL-MC  | 0.748| 0.653        | 0.711 |
| TrustAL-NC  | 0.780| 0.670        | 0.729 |

(a) Noise ratio : 7%
(b) Noise ratio : 15%

Figure 6: Robustness analysis varying the ratio of noise. Accuracy and MCI are shown in pair for each noise ratio.

Conclusion

In this paper, we debunk the monotonicity assumption which is a common belief in conventional AL methods by empirical observation of example forgetting. For that, we present TrustAL, an effective and robust framework that uses the predecessor model as an expert model for knowledge distillation to compensate the loss of knowledge between data and model. Especially, our framework can be orthogonally applicable to existing data acquisition in a highly efficient way. Further, we present multi-pronged analysis for our method through extensive experiments.
Acknowledgements
This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2020-0-01361, Artificial Intelligence Graduate School Program (Yonsei University)). And this research was partly supported by the MSIT(Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program(2021-11-1603) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). Jinyoung Yeo is a corresponding author.

References
Asghar, N.; Poupart, P.; Jiang, X.; and Li, H. 2017. Deep Active Learning for Dialogue Generation. In Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017), 78–83.
Ash, J.; and Adams, R. P. 2020. On Warm-Starting Neural Network Training. Advances in Neural Information Processing Systems, 33.
Ash, J. T.; Zhang, C.; Krishnamurthy, A.; Langford, J.; and Agarwal, A. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. ICLR.
Bodó, Z.; Minier, Z.; and Csató, L. 2011. Active learning with clustering. In JMLR.
Bouguelia, M.-R.; Nowaczyk, S.; Santosh, K.; and Verikas, A. 2018. Agreeing to disagree: Active learning with noisy labels without crowdsourcing. International Journal of Machine Learning and Cybernetics, 9(8): 1307–1319.
Dasgupta, S. 2011. Two faces of active learning. Theoretical computer science, 412(19): 1767–1781.
Farquhar, S.; Gal, Y.; and Rainforth, T. 2020. On Statistical Bias In Active Learning: How and When to Fix It. In International Conference on Learning Representations.
Fukuda, T.; Suzuki, M.; Kurata, G.; Thomas, S.; Cui, J.; and Ramabhadran, B. 2017. Efficient Knowledge Distillation from an Ensemble of Teachers. In INTERSPEECH.
Furlanello, T.; Lipton, Z.; Tschanen, M.; Itti, L.; and Anandkumar, A. 2018. Born again neural networks. In International Conference on Machine Learning, 1607–1616. PMLR.
He, T.; Jin, X.; Ding, G.; Yi, L.; and Yan, C. 2019. Towards better uncertainty sampling: Active learning with multiple views for deep convolutional neural network. In 2019 IEEE International Conference on Multimedia and Expo (ICME), 1360–1365. IEEE.
Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.
Hochreiter, S.; and Schmidhuber, J. 1997. LSTM can solve hard long time lag problems. NIPS.
Hu, R.; Mac Namee, B.; and Delany, S. J. 2010. Off to a good start: Using clustering to select the initial training set in active learning. In Twenty-Third International FLAIRS Conference.

Ishibashi, H.; and Hino, H. 2020. Stopping criterion for active learning based on deterministic generalization bounds. In International Conference on Artificial Intelligence and Statistics, 386–397. PMLR.
Karamcheti, S.; Krishna, R.; Fei-Fei, L.; and Manning, C. 2021. Mind Your Outliers! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).
Lewis, D. D.; and Gale, W. A. 1994. A sequential algorithm for training text classifiers. In SIGIR '94, 3–12. Springer.
Liu, Q.; Zhu, Y.; Liu, Z.; Zhang, Y.; and Wu, S. 2021. Deep Active Learning for Text Classification with Diverse Interpretations. arXiv preprint arXiv:2108.10687.
Lowell, D.; Lipton, Z. C.; and Wallace, B. C. 2018. Practical obstacles to deploying active learning. EMNLP-IJCNLP.
Pang, B.; and Lee, L. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. ACL.
Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3967–3976.
Prabhu, A.; Dognin, C.; and Singh, M. 2019. Sampling Bias in Deep Active Classification: An Empirical Study. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 4058–4068.
Ranganathan, H.; Venkateswara, H.; Chakraborty, S.; and Panchanathan, S. 2017. Deep active learning for image classification. In 2017 IEEE International Conference on Image Processing (ICIP), 3934–3938. IEEE.
Ren, P.; Xiao, Y.; Chang, X.; Huang, P.-Y.; Li, Z.; Chen, X.; and Wang, X. 2020. A Survey of Deep Active Learning. arXiv preprint arXiv:2009.00236.
Roth, D.; Cumby, C.; Li, X.; Morie, P.; Nagarajan, R.; Rizzolo, N.; Small, K.; and Yih, W.-t. 2002. Question-answering via enhanced understanding of questions. In TREC. Citeseer.
Sener, O.; and Savarese, S. 2018. Active learning for convolutional neural networks: A core-set approach. ICLR.
Settles, B. 2009. Active learning literature survey.
Shannon, C. E. 2001. A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review, 5(1): 3–55.
Siddhanth, A.; and Lipton, Z. C. 2018. Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2904–2909.
Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models
for semantic compositionality over a sentiment treebank. In *EMNLP*.

Swayamdipta, S.; Schwartz, R.; Lourie, N.; Wang, Y.; Hajishirzi, H.; Smith, N. A.; and Choi, Y. 2020. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9275–9293.

Toneva, M.; Sordoni, A.; Combes, R. T. d.; Trischler, A.; Bengio, Y.; and Gordon, G. J. 2019. An empirical study of example forgetting during deep neural network learning. *ICLR*.

Wang, D.; and Shang, Y. 2014. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, 112–119. IEEE.

Wang, K.; Zhang, D.; Li, Y.; Zhang, R.; and Lin, L. 2016. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12): 2591–2600.

Wang, L.; Ghosh, D.; Gonzalez Diaz, M.; Farahat, A.; Alam, M.; Gupta, C.; Chen, J.; and Marathe, M. 2020. Wisdom of the Ensemble: Improving Consistency of Deep Learning Models. *Advances in Neural Information Processing Systems*, 33.

Xu, Z.; Yu, K.; Tresp, V.; Xu, X.; and Wang, J. 2003. Representative sampling for text classification using support vector machines. In *ECIR*, 393–407. Springer.

Yuan, L.; Tay, F. E.; Li, G.; Wang, T.; and Feng, J. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3903–3911.

Yuan, M.; Lin, H.-T.; and Boyd-Graber, J. 2020. Cold-start active learning through self-supervised language modeling. *EMNLP*.

Yun, J.; Kim, B.; and Kim, J. 2020. Weight Decay Scheduling and Knowledge Distillation for Active Learning. In *European Conference on Computer Vision*, 431–447. Springer.

Zhdanov, F. 2019. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*.

Zhou, Y.; Renduchintala, A.; Li, X.; Wang, S.; Mehdad, Y.; and Ghoshal, A. 2021. Towards Understanding the Behaviors of Optimal Deep Active Learning Algorithms. In *International Conference on Artificial Intelligence and Statistics*, 1486–1494. PMLR.
Appendix A: Dataset

We use the following three text classification datasets. The statistics for each dataset is shown in Table 2. We split the official training split into 90% and 10% as training and validation sets respectively.

TREC (Roth et al. 2002): This dataset is for question classification task with 6 categories related to the subject of the question.

Movie Review (Pang and Lee 2005): This polarity dataset is for the sentimental classification task (positive/negative) using movie review snippets on a five star scale.

SST-2 (Socher et al. 2013): Stanford Sentiment Treebank dataset is for sentimental classification task using a binary class (positive/negative).

| Dataset        | Train | Dev  | Test | #Classes |
|----------------|-------|------|------|----------|
| TREC           | 4,952 | 500  | 500  | 6        |
| Movie review   | 8,526 | 1,068| 1,068| 2        |
| SST-2          | 6,920 | 872  | 1,821| 2        |

Table 2: Text classification dataset statistics

Appendix B: Baselines

We apply the following three data acquisition methods to TrustAL framework, while considering their stand-alone as baselines.

CONF (Wang and Shang 2014): This strategy samples with the smallest of the predicted class logit, for k classes, \( \max_{i=1}^{k} f(x; \theta)_i \). Specifically, the sample is classified as uncertain if the probability of the most probable label for a sample is small.

CORESET (Sener and Savarese 2018): This approach attempts to select samples by constructing a core subset. The embedding of each sample is computed by the second end layer and the samples at each round are selected using a greedy furthest-first traversal conditioned on all labeled samples.

BADGE (Ash et al. 2019): The approach samples point by using clustering in the hallucinated gradient space represented approximately based on pseudo labels. This method automatically balances the prediction uncertainty of the model and the diversity of the samples in a batch.

Appendix C: Training Details

In the entire process of model and conducting experiments, we used machine with Intel CoffeeLake i5-9400F CPU, 64GB of RAM and RTX 2080ti GPU. For the model architecture, we use single layer Bi-LSTM with 200 hidden units with dropout 0.5. The hyperparameters for LSTMs on each task are listed in Table 3.

We conduct the experiments in section with five random seeds. We represent words initially with GloVe vectors (?) with 300-dimensional glove embeddings pretrained on 6B tokens. We use a batch size set to be 50 which is common setting in AL assuming low budget. The active learning process begins with a random acquisition of 2% from the dataset. We train an initial model on this data. Then, we iteratively apply baseline acquisition function to sample an additional 2% of examples and train a new model based on this data. For each round, we evaluate models for every epoch on development set and select the model with the highest validation accuracy to report the test accuracy.

| hyperparameter | TREC | Movie review | SST-2 |
|----------------|------|--------------|-------|
| Optimizer      | Adam | Adam         | Adam  |
| Learning rate  | 0.001| 0.001        | 0.001 |
| Training epochs| 10   | 10           | 10    |
| \( \alpha \) for \( L_{KL} \) | 0.75 | 0.75         | 1     |
| Batch size     | 50   | 50           | 50    |

Table 3: The hyperparameters of the experiment

Appendix D: Robustness Analysis

We evaluate the robustness of TrustAL by varying the hyperparameters that can affect the performance. The experiment is conducted on TREC with three AL algorithms we used across this paper. The performance is the average performance of each stage. The relative performance difference from each stand-alone baseline is also presented in parenthesis where “+” indicates performance gain and “−” indicates performance degradation. The experimental results of TrustAL-NC are presented in Table (5-7) and TrustAL-MC are presented in Table (8-10).

The Size of the Budget \( k \) The number of candidates for model selection in TrustAL-NC depends on the size of the budget \( k \). We compare the performance of TrustAL-NC from low-budget to high-budget on three datasets. As a result, TrustAL-NC outperforms baseline regardless of the size of the budget \( k \) in most cases.

The Size of the Validation Set We conduct an experiment to validate whether TrustAL-NC is robust on the size of development set. The experiment on TREC dataset shows that TrustAL-NC shows robust performance even with half-sized development set, resulting only 0.008 point decrease in accuracy on average over 3 AL strategies. We believe that these results relieve some concern about infeasibility of keeping development set in data scarce situations.

Preferential Weight for \( L_{KD} \) To analyze the effect of KD, we report the performance varying preference weights from 0.3 to 20. As a result, larger preference weight \( \alpha \) mostly results in lower MCI. However, excessive preference weight seems result lower accuracy.

Appendix E: TrustAL-NC using BERT

To validate TrustAL-NC on different architecture, we report the performance of BERT(?) We use Hugging Face’s im-
|       | Stable acc. | Saturated acc. | Stable MCI | Saturated MCI |
|-------|-------------|----------------|------------|---------------|
| **A** | baseline    | 0.759          | 159.6      | 0.844         | 70.0          |
|       | TrustAL-NC  | 0.764          | 154.7      | 0.848         | 65.6          |
| **B** | baseline    | 0.757          | 136.9      | 0.846         | 64.9          |
|       | TrustAL-NC  | 0.758          | 118.0      | 0.840         | 60.0          |
| **C** | baseline    | 0.769          | 123.9      | 0.851         | 60.0          |
|       | TrustAL-NC  | 0.769          | 114.2      | 0.844         | 59.0          |

Table 4: Average performance of TrustAL-NC using BERT with (A) CONF (B) CORESET and (C) BADGE on SST-2.

Implementation of the BERT-small for efficiency. We use the AdamW optimizer with learning rate of 2e-5 with batch size 32, a max number of tokens of 128, epoch 3, adam epsilon 1e-8, gradients clipping to 5, distillation preference weight rate 1.5. TrustAL-NC using BERT outperforms the baseline regardless of the AL methods as it is shown in Table 4.

Appendix F: Experimental Results

Additional experimental results on Movie Review and SST-2 are presented in Figure 7 and Figure 8.

Appendix G: Time Complexity of TrustAL-NC

Here, we compare TrustAL-NC and query-by-committee in terms of time complexity to clarify the difference between two confusing concepts. Both utilize multiple models. However we claim TrustAL is (a) orthogonally applicable to any AL strategies including query-by-committee and (b) more efficient than query-by-committee inference/training. Regarding the inference time, TrustAL uses a single model to evaluate each sample in 𝑈. TrustAL only requires extra inference time to obtain pseudo labels which can be optimized by simple caching trick to store the predictions of training samples and make predictions only on non-overlapping training examples which are newly acquired. We found this technique to be especially efficient considering the overlapping teacher selection tendency presented in Figure 4. Also, regarding training time, TrustAL only trains a student model and keeps the model fixed after a training step. Query-by-committee usually trains multiple committee models after each acquisition step which is computationally expensive.

\[^3\]https://github.com/huggingface/transformers
Figure 7: Accuracy versus the ratio of labeled samples (a-c) and MCI versus the ratio of labeled samples (d-f) on Movie review test dataset.

Figure 8: Accuracy versus the ratio of labeled samples (a-c) and MCI versus the ratio of labeled samples (d-f) on SST-2 test dataset.
### Table 5: Performance of TrustAL-NC with CONF on TREC for robustness analysis. The hyperparameter of the experiment on RQ1 is marked with *.

|                  | Stable acc. | CI acc. | Saturated acc. | CI acc. |
|------------------|-------------|---------|----------------|---------|
| **Ratio of $D_{dev}$** |             |         |                |         |
| 0.5              | 0.792(+0.022) | 25.02(-5.80) | 0.898(+0.042) | 13.90(-11.12) |
| 1.0*             | 0.788(+0.011) | 17.94(-5.83) | 0.896(+0.062) | 13.64(-14.43) |
| 2.0              | 0.784(+0.018) | 22.08(-4.49) | 0.876(+0.044) | 17.30(-10.76) |
| 0.02*            | 0.788(+0.011) | 17.94(-5.83) | 0.896(+0.062) | 13.64(-14.43) |
| **Size of budget $k$** |             |         |                |         |
| 0.04             | 0.802(+0.016) | 15.99(-3.38) | 0.906(+0.041) | 11.02(-11.52) |
| 0.1              | 0.825(-0.002) | 8.17(-7.48) | 0.896(+0.019) | 11.06(-8.11) |
| **$\alpha$ for $L_{KL}$** |             |         |                |         |
| 0.3              | 0.786(+0.009) | 23.86(+0.09) | 0.891(+0.058) | 16.57(-11.50) |
| 0.75*            | 0.788(+0.011) | 17.94(-5.83) | 0.896(+0.062) | 13.64(-14.43) |
| 1.5              | 0.802(+0.026) | 16.94(-6.83) | 0.895(+0.062) | 11.56(-16.50) |
| 10               | 0.810(+0.033) | 10.62(-13.15) | 0.892(+0.059) | 9.44(-18.62) |
| 20               | 0.793(+0.016) | 10.47(-13.30) | 0.874(+0.041) | 8.57(-19.49) |

### Table 6: Performance of TrustAL-NC with CORESET on TREC for robustness analysis. The hyperparameter of the experiment on RQ1 is marked with *.

|                  | Stable acc. | CI acc. | Saturated acc. | CI acc. |
|------------------|-------------|---------|----------------|---------|
| **Ratio of $D_{dev}$** |             |         |                |         |
| 0.5              | 0.785(+0.012) | 25.08(-2.98) | 0.908(+0.062) | 11.18(-16.03) |
| 1.0*             | 0.808(+0.035) | 18.06(-7.24) | 0.884(+0.038) | 14.89(-13.34) |
| 2.0              | 0.782(+0.019) | 23.50(-3.83) | 0.885(+0.061) | 15.48(-18.35) |
| 0.02*            | 0.808(+0.035) | 18.06(-7.24) | 0.884(+0.038) | 14.89(-13.34) |
| **Size of budget $k$** |             |         |                |         |
| 0.04             | 0.798(+0.014) | 13.21(-6.65) | 0.896(+0.026) | 11.72(-9.76) |
| 0.1              | 0.823(+0.005) | 10.85(-4.38) | 0.895(+0.018) | 10.21(-6.53) |
| **$\alpha$ for $L_{KL}$** |             |         |                |         |
| 0.3              | 0.793(+0.019) | 18.51(-6.79) | 0.895(+0.048) | 13.96(-14.27) |
| 0.75*            | 0.808(+0.035) | 18.06(-7.24) | 0.884(+0.038) | 14.89(-13.34) |
| 1.5              | 0.811(+0.037) | 14.33(-10.96) | 0.895(+0.049) | 10.74(-17.50) |
| 10               | 0.819(+0.046) | 12.54(-12.75) | 0.899(+0.052) | 8.84(-19.39) |
| 20               | 0.803(+0.029) | 9.94(-15.36) | 0.891(+0.044) | 8.21(-20.02) |

### Table 7: Performance of TrustAL-NC with BADGE on TREC for robustness analysis. The hyperparameter of the experiment on RQ1 is marked with *.

|                  | Stable Learning Stage accuracy | MCI | Saturated Learning Stage accuracy | MCI |
|------------------|-------------------------------|-----|----------------------------------|-----|
| **Ratio of $D_{dev}$** |                   |       |                                   |     |
| 0.5              | 0.792(+0.037) | 22.47(-9.53) | 0.897(+0.063) | 13.40(-14.90) |
| 1.0*             | 0.796(+0.017) | 17.13(-4.61) | 0.896(+0.048) | 14.08(-11.49) |
| 2.0              | 0.791(+0.011) | 19.12(-5.13) | 0.899(+0.053) | 15.96(-14.32) |
| 0.02*            | 0.796(+0.017) | 17.13(-4.61) | 0.896(+0.048) | 14.08(-11.49) |
| **Size of budget $k$** |                   |       |                                   |     |
| 0.04             | 0.787(-0.002) | 16.78(-1.19) | 0.911(+0.040) | 10.84(-10.77) |
| 0.1              | 0.816(+0.010) | 10.98(-6.23) | 0.897(-0.004) | 13.21(-2.00) |
| 0.3              | 0.794(+0.015) | 19.28(-2.47) | 0.884(+0.036) | 17.53(-8.04) |
| **$\alpha$ for $L_{KL}$** |                   |       |                                   |     |
| 0.75*            | 0.796(+0.017) | 17.13(-4.61) | 0.896(+0.048) | 14.08(-11.49) |
| 1.5              | 0.792(+0.013) | 16.79(-4.96) | 0.904(+0.056) | 10.73(-14.84) |
| 10               | 0.797(+0.018) | 13.81(-7.94) | 0.891(+0.043) | 10.66(-14.91) |
| 20               | 0.788(+0.010) | 11.51(-10.23) | 0.887(+0.039) | 10.33(-15.24) |
### Table 8: Performance of TrustAL-MC with CONF on TREC for robustness analysis. The hyperparameter of the experiment on RQ1 is marked with *. Ratio of $\frac{D_{dev}}{D}$ is skipped for TrustAL-MC.

| Size of budget $k$ | Stable acc. | Stable CI | Saturated acc. | Saturated CI |
|-------------------|-------------|-----------|----------------|--------------|
| 0.02*             | 0.774(-0.003) | 23.55(-0.22) | 0.861(+0.028) | 20.20(-7.86) |
| 0.04              | 0.797(+0.011) | 19.98(+0.62) | 0.883(+0.018) | 16.51(-6.03) |
| 0.1               | 0.820(-0.007) | 9.06(-6.58)  | 0.900(+0.024) | 8.47(-10.69) |
| $\alpha$ for $KL$ | 0.3         | 0.760(-0.016) | 26.52(+2.75)  | 0.847(+0.014) | 23.10(-4.97) |
| 0.75*             | 0.774(-0.003) | 23.55(-0.22) | 0.861(+0.028) | 20.20(-7.86) |
| 1.5               | 0.786(+0.009) | 20.42(-3.35) | 0.869(+0.035) | 19.49(-8.57) |
| 10                | 0.799(+0.022) | 16.45(-7.32) | 0.879(+0.046) | 15.41(-12.65) |
| 20                | 0.803(+0.026) | 14.56(-9.21) | 0.882(+0.048) | 12.89(-15.17) |

### Table 9: Performance of TrustAL-MC with CORESET on TREC for robustness analysis. The hyperparameter of the experiment on RQ1 is marked with *. Ratio of $\frac{D_{dev}}{D}$ is skipped for TrustAL-MC.

| Size of budget $k$ | Stable acc. | Stable CI | Saturated acc. | Saturated CI |
|-------------------|-------------|-----------|----------------|--------------|
| 0.02*             | 0.791(+0.018) | 21.45(-3.84) | 0.853(+0.007) | 22.92(-5.31) |
| 0.04              | 0.794(+0.010) | 17.07(-2.79) | 0.875(+0.005) | 19.11(-2.37) |
| 0.1               | 0.820(+0.002) | 12.23(-3.0)  | 0.900(+0.022) | 9.96(-6.78)  |
| $\alpha$ for $KL$ | 0.3         | 0.776(+0.002) | 21.62(-3.67) | 0.839(-0.007) | 27.78(-0.45) |
| 0.75*             | 0.791(+0.018) | 21.45(-3.84) | 0.853(+0.007) | 22.92(-5.31) |
| 1.5               | 0.795(+0.021) | 20.08(-5.21) | 0.852(+0.005) | 22.97(-5.26) |
| 10                | 0.806(+0.033) | 16.82(-8.48) | 0.875(+0.029) | 18.08(-10.15) |
| 20                | 0.811(+0.037) | 13.66(-11.64)| 0.890(+0.043) | 13.01(-15.22) |

### Table 10: Performance of TrustAL-MC with BADGE on TREC for robustness analysis. The hyperparameter of the experiment on RQ1 is marked with *. Ratio of $\frac{D_{dev}}{D}$ is skipped for TrustAL-MC.

| Size of budget $k$ | Stable acc. | Stable CI | Saturated acc. | Saturated CI |
|-------------------|-------------|-----------|----------------|--------------|
| 0.02*             | 0.779(+0.001) | 23.68(+1.93) | 0.862(+0.014) | 20.02(-5.55) |
| 0.04              | 0.787(-0.001) | 17.76(-0.21) | 0.89(+0.019)  | 16.06(-5.55) |
| 0.1               | 0.808(+0.002) | 17.04(-0.17) | 0.901(+0.000) | 11.30(-3.91) |
| $\alpha$ for $KL$ | 0.3         | 0.774(-0.004) | 24.82(+3.07)  | 0.855(+0.007) | 22.40(-3.17) |
| 0.75*             | 0.779(+0.001) | 23.68(+1.93) | 0.862(+0.014) | 20.02(-5.55) |
| 1.5               | 0.787(+0.008) | 18.84(-2.90) | 0.874(+0.026) | 21.27(-4.30) |
| 10                | 0.818(+0.040) | 16.01(-5.74) | 0.871(+0.023) | 22.27(-3.29) |
| 20                | 0.789(+0.011) | 15.84(-5.91) | 0.878(+0.030) | 13.26(-12.31) |