Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study

Aditya Siddhant
Carnegie Mellon University
asiddhan@cs.cmu.edu

Zachary C. Lipton
Carnegie Mellon University
zlipton@cmu.edu

Abstract

Several recent papers investigate Active Learning (AL) for mitigating the data-dependence of deep learning for natural language processing. However, the applicability of AL to real-world problems remains an open question. While in supervised learning, practitioners can try many different methods, evaluating each against a validation set before selecting a model, AL affords no such luxury. Over the course of one AL run, an agent annotates its dataset exhausting its labeling budget. Thus, given a new task, an active learner has no opportunity to compare models and acquisition functions. This paper provides a large-scale empirical study of deep active learning, addressing multiple tasks and, for each, multiple datasets, multiple models, and a full suite of acquisition functions. We find that across all settings, *Bayesian active learning by disagreement*, using uncertainty estimates provided either by Dropout or Bayes-by-Backprop significantly improves over i.i.d. baselines and usually outperforms classic uncertainty sampling.

1 Introduction

While over the past several years, deep learning has pushed the state of the art on numerous tasks, its extreme data-dependence presents a formidable obstacle under restricted annotation budgets. Active Learning (AL) presents one promising approach to reduce deep learning's data requirements (Cohn et al., 1996). Strategically selecting points to annotate over alternating rounds of labeling and learning, an active learner is hoped to outperform budget-matched i.i.d. labeling. Typical *acquisition functions* select examples for which the current predictor is most uncertain. However, how precisely to quantify uncertainty, especially for neural networks, remains an open question.

Classical approaches interpret either the entropy or the negative argmax of the predictive (e.g. softmax) distribution as the model's uncertainty, yielding the *maximum entropy* and *least confidence* heuristics, respectively. These approaches account for aleatoric but not epistemic uncertainty (Kendall and Gal, 2017). Several recent Bayesian formulations of deep learning provide alternative techniques for extracting uncertainty estimates from deep networks, including a *dropout*-based approach (Gal and Ghahramani, 2016b), previously employed in Deep Active Learning (DAL) for image classification (Gal et al., 2017) and named entity recognition (Shen et al., 2018), and Bayes-by-Backprop (Blundell et al., 2015). To our knowledge, our paper is the first to apply Bayes-by-Backprop in the context of DAL.

While the results in recent papers hint at DAL's potential, its suitability in practice has yet to be proven. That's because papers often address just a single task, just a single model, and sometimes just one or two datasets. However, it's not enough to look back retrospectively after a final round of experiments and declare that one acquisition function outperforms an i.i.d. baseline. To apply DAL in practice, we must be confident that the technique will work correctly—*the first time*—on a dataset that we have never seen before. Otherwise, we might exhaust the annotation budget while performing worse than an i.i.d. baseline. Once we've exhausted our resources for labeling, there's no going back. Moreover, many DAL papers suffer from implicit target leaks. The architectures and hyper-parameters are often tuned using the full dataset, before concealing the labels and *simulating* AL.

In this paper, we present a large-scale study[1], comparing various acquisition functions across multiple tasks: Sentiment Classification (SC),

[1] Code for all of our models and for running active learning experiments can be found at https://github.com/asiddhant/Active-NLP

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2904–2909
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

Named Entity Recognition (NER), and Semantic Role Labeling (SRL). For each task we consider, with multiple datasets, multiple models, and multiple acquisition functions. Moreover, in all experiments, we set hyper-parameters on warm-start data, allowing for a more honest assessment. This paper does not seek to champion any one approach but instead to ask, *is there any single method that we can reliably expect to work out-of-the-box on a new problem?*

To our surprise, we find that BALD (Houlsby et al., 2011), which measures uncertainty by the frequency over multiple Monte Carlo draws from a stochastic model with which the drawn models disagree with the plurality, proved effective across all combinations of task, dataset, and model. Moreover both variants of the approach, drawing samples according to the dropout method (Gal et al., 2017) and from a Bayes-by-Backprop network (Blundell et al., 2015), performed similarly well across most tasks, datasets, and models.

Related Work Only a few papers have addressed DAL for NLP, notably Shen et al. (2018) for NER and Zhang et al. (2017) who address text classification, proposing to select examples according to the expected magnitude of updates to word embeddings. In this paper, we do not consider the latter heuristic because we address sequence tagging tasks, where the difficulty of marginalizing over all possible labels blows up exponentially with sequence length. While both previous papers do conduct experiments on multiple datasets (2 and 3, respectively) they each consider just one task and just one model.

Gal et al. (2017) apply the dropout-based uncertainty estimates due to (Gal and Ghahramani, 2016a) together with the BALD framework due to (Houlsby et al., 2011) for image classification with convolutional neural networks. They obtain significant improvement over classic uncertainty-based acquisition functions on the MNIST dataset and for diagnosing skin cancer from lesion images (ISIC2016 task). Our work builds on theirs, both by offering a large-scale evaluation of BALD for NLP tasks and models, and by exploring BALD with another method for estimating uncertainty: the uncertainty of the weights as modeled by a Bayes-by-Backprop network.

2 Bayesian Deep Learning

While space constraints preclude an extensive discussion of the various Bayesian formulations of neural nets, we briefly summarize the methods compared in this paper, pointing out various design decisions that are important for reproducing our results.

Monte Carlo Dropout According to (Gal and Ghahramani, 2016b), the dropout regularization techniques for neural networks can be interpreted as a Bayesian approximation to Gaussian processes (Rasmussen, 2004). Here, unlike standard uses of dropout, we apply it at prediction time. Uncertainty estimates are produced by comparing the output of a trained neural network using T different stochastic passes through the neural network. The extension to CNNs is straightforward. To apply dropout to RNNs, we follow the approach due to (Gal and Ghahramani, 2016c), who extended their variational analysis to RNNs, arguing that dropout ought to be applied to the recurrent layers (and not just the synchronous connections, per previous standard practice (Zaremba et al., 2014)) by applying identical dropout masks at each sequence step.

Bayes by Backprop In this approach due to Blundell et al. (2015), instead of maintaining a point estimate for each weight, we maintain a probability distribution over the weights. A standard L-layer MLP model $P(y|x, w)$ is parametrized by weights $w = \{W_l, b_l\}_{l=1}^{L} \in \mathbb{R}^d$. Then, $\hat{y} = \phi(W_L \cdot ... \cdot +\phi(W_1 \cdot x + b_1) + .. + b_L)$ where ϕ is an activation function such as tanh or ReLU. Bayes-by-Backprop represents imposes a prior over the weights, $p(w)$ and seeks to learn the posterior distribution $p(w|D)$ given training data $D = \{x^i, y^i\}_{i=1}^{N}$. To deal with intractability, Bayes-by-Backprop approximates $p(w|D)$ by a variational distribution $q(w|\theta)$, typically choosing q to be a Gaussian with diagonal covariance and each weight sampled from $\mathcal{N}(\mu_i, \sigma_i^2)$. To enforce non-negativity, the σ_i are further parametrized via the *softplus* function $\sigma_i = log(1 + exp(\rho_i))$ giving variational parameters $\theta = \{\mu_i, \rho_i\}_{i=1}^{d}$.

Our objective in optimizing the variational parameters is to minimize the KL divergence between $q(\theta)$ and $p(w|D)$. Some simplification of the objective gives $\mathcal{L}(D, \theta) = \sum_{j=1}^{N} \left[\log q(w^j|\theta) - \log p(w^j) - \log p(D|w^j)\right]$, where w^j denotes the j-th Monte Carlo sample drawn from $q(w|\theta)$ (we use $N = 1$). In Bayes-by-Backprop, the parameters are optimized by stochastic gradient descent, using the re-parameterization trick popularized by Kingma and Welling (2014). Extending Bayes-by-Backprop to CNNs and RNNs is straightforward with the latter requiring minor adjustments for truncated back-propagation through time (Fortunato et al., 2017). Uncertainty estimates calculated via Bayes-by-Backprop have been shown to be useful for efficient exploration in reinforcement learning (Lipton et al., 2018).

3 Experimental Setup

3.1 Acquisition functions

In this work, we consider only uncertainty-based acquisition. In particular, we consider least confidence (LC) for classification and maximum length-normalized log probability (MNLP) for sequence labeling tasks (Shen et al., 2018). LC chooses that example with for which the prediction has lowest predicted probability. MNLP extends this to sequences, selecting by log probability normalized by length, removing the bias for the model to preferentially select longer sequences.

BALD We briefly articulate the details of the Bayesian Active Learning by Disagreement (BALD) approach due to Houlsby et al. (2011), upon which both our Bayesian approaches are based. We denote Monte Carlo Dropout Disagreement by DO-BALD and its Bayes-by-Backprop counterpart as BB-BALD. BALD originally selects samples that maximise the information gained about the model parameters. This boils down to choosing data points which each stochastic forward pass through the model would have the highest probability assigned to a different class (Gal et al., 2017). Our measure of uncertainty is the fraction of models, across MC samples from the network, that that disagree with most popular choice. This can be mathematically represented as

$$\arg\max_{j} \left(1 - \frac{count(mode(\tilde{y}_j^{(1)}, ..., \tilde{y}_j^{(T)}))}{T} \right)$$

Here $\tilde{y}_j^{(t)}$ represents the prediction (argmax) applied to the tth forward pass on jth sample $\tilde{y}_j^{(t)} = \texttt{argmax}(\hat{y}_j^{(t)})$. We resolve ties by choosing the least confident predictions as determined by the mean probability assigned to the consensus class. For sequences, we look at agreement on the entire sequence tag, noting that this may exhibit a bias to preferentially sample longer sentences. Because we measure the budget at each round in words (not sentences), while this constitutes a bias, it does not constitute an unfair advantage. Moreover, we note that all AL necessarily consists of biased sampling.

3.2 Training details

The active learning process begins with a random acquisition of 2% *warmstart* samples from the dataset. We train an initial model on this data. Then based on this model's uncertainty estimates, we apply our chosen acquisition function to sample an additional 2% of examples and train a new model based on this data In each round, we train from scratch to avoid badly overfitting the data collected in earlier rounds per observations by Hu et al. (2018). We continue with alternating rounds of labeling and training until we have annotated 50% of the dataset. For classification tasks, the we measure the budget in sentences while for sequence labeling, we measure the budget by the number of words because the annotator must provide one tag per word.

In each iteration, we train each model to convergence, decided based on early stopping with a patience of 1 epoch, or 25 epochs (whichever comes earlier). For datasets with fixed validation sets such as Conll 2003, instead of using the entire validation set for early stopping, we use the percentage of validation data equivalent to that in our current training pool. Our motivation here is to keep the simulation realistic. Essentially, we assume that given a large annotation budget, one will collect both a larger training set and a larger validation set. As a motivating example, it seems unreasonable that a practitioner might have only 500 training examples but 10,000 examples available for early stopping. Our reported results are averaged over 3 runs with different warmstart samples.

3.3 Sentence Classification

We use two datasets for simulation: one question classification dataset TrecQA (Roth et al., 2002) and one sentiment analysis dataset (Pang and Lee, 2005) and two architectures for training: CNNs and BiLSTMs. For implementation of the CNNs on both these datasets, we follow the setup of Kim (2014) and for BiLSTMs, we use a single layer model with 300 hidden units for both datasets. We

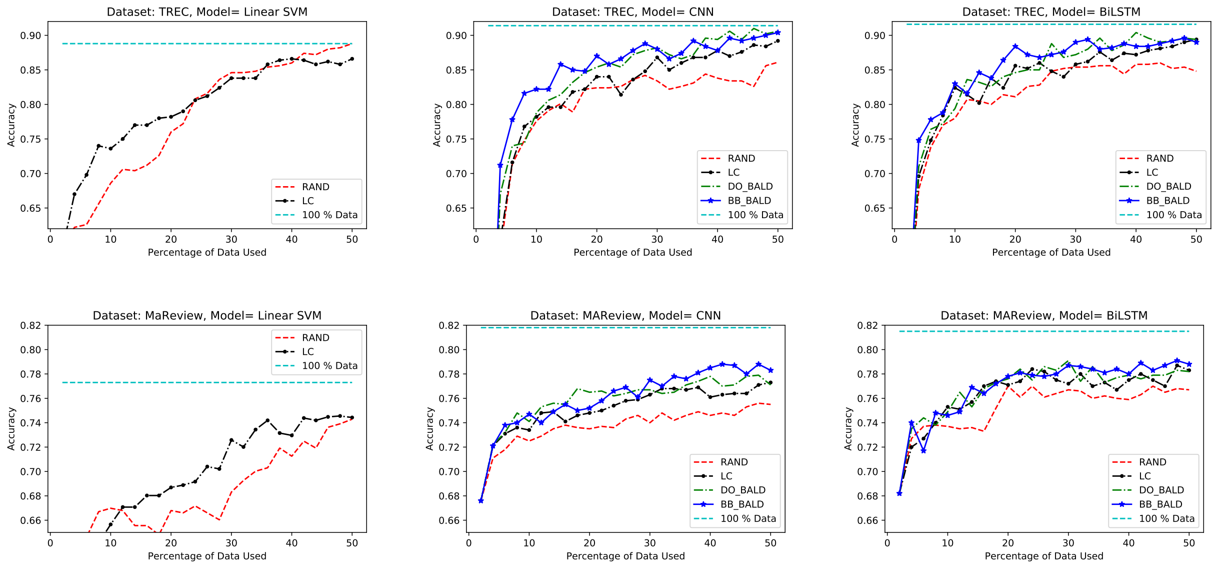

Figure 1: Performance of various models and acquisition functions for two SC datasets

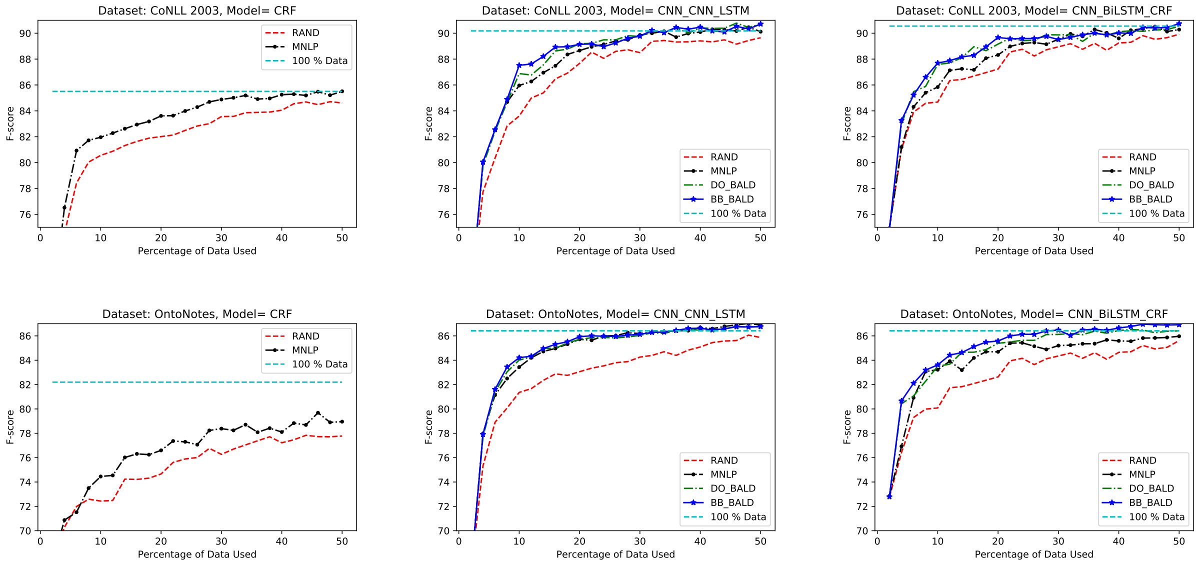

Figure 2: Performance of various models and acquisition functions for two NER datasets

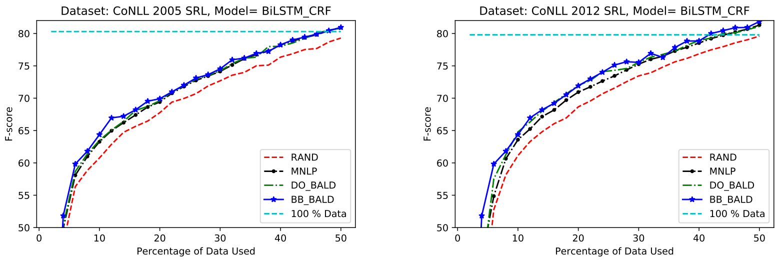

Figure 3: Performance of different acquisition functions on SRL task for two datasets

use 300-dimensional glove embeddings (Pennington et al., 2014) pretrained on 6B tokens for all 4 settings, a dropout rate of 0.5, and the Adam optimizer (Kinga and Adam, 2015) with initial learning rate 1e-3. We use a batch size set to be either 50 or the number required for at least 10 updates whichever is lower. This is done to ensure that when the training pool is small, the batch size is not too large and models get sufficient number of updates in an epoch. We also train a Unigram + Bigram + Linear SVM model with LC acquisition as a shallow AL baseline.

3.4 Named Entity Recognition

Again, we use two datasets: CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes 5.0. The two architectures used for training are CNN-BiLSTM-CRF (CNN for character-level encoding, BiLSTM for word-level encoding, and CRF for decoding) (Ma and Hovy, 2016) and CNN-CNN-LSTM (CNN for character-level encoding, CNN for word-level encoding, and LSTM for decoding) (Shen et al., 2018). We follow the exact experimental settings of these papers except that batch size is 16 for CoNLL and 80 for OntoNotes (minimum 10 updates heuristic is followed here too).

We note that our NER models consist of multiple modular components, and that we only train a subset of those units in a Bayesian fashion. In both DO-BALD and BB-BALD, we apply dropout/stochastic weights on the word-level layers, but not on the character-level encoders or decoding layers. For example, with DO-BALD, we apply recurrent dropout in the BiLSTM word-level component of CNN-BiLSTM-CRF and we apply normal dropout in the word-level (middle) CNN layer of the CNN-CNN-LSTM. For NER, as a shallow AL baseline, we have a linear chain CRF model with MNLP acquisition.

3.5 Semantic Role Labeling

We consider two datasets: CoNLL 2005 (Carreras and Màrquez, 2005) and CoNLL 2012, focusing only on an LSTM-based model this time. Our model resembles He et al. (2017), but instead of using contained A* decoding, we use a CRF decoder, noting that while this causes a 2% drop in performance (at 100% annotation), our goal is to compare acquisition functions, not achieve record-setting performance. We follow the experimental setup of the paper but use a higher dropout rate of 0.25, adjusting the batch size according to the minimum update heuristic.

3.6 Results

We plot the performance for various annotation budgets for all combinations of dataset, model, and acquisition function, for the SC, NER, and SRL tasks in Figures 1, 2, and 3, respectively. In all cases, the active learning methods perform better than random i.i.d. baseline. We note that across the board, DAL methods show significant improvement over shallow baselines. The Bayesian acquisition functions, DO-BALD and BB-BALD consistently outperform classic uncertainly sampling, although in a few cases including the setting considered by Shen et al. (2018), the improvement is only marginal. This finding underscores the importance of examining proposed AL methods on a broad set of representative tasks and with a broad set of representative models.

In general, we find that the advantages of DAL can be substantial. For example, on NER tasks, we achieve roughly 98-99% of the full-dataset performance while labeling only 20% of the samples for both CNN-BiLSTM-CRF and CNN-CNN-LSTM models. By comparison, the i.i.d. baseline requires 50% of the data to achieve comparable F score. While the reduction in the percentage of data required is not as dramatic in the classification datasets (possibly owing to their comparatively small size), the relative improvement over i.i.d. baselines remains significant.

4 Conclusion

This paper set out to investigate the practical utility of DAL for NLP. Our study consisted of over 40 experiments, each repeated for 3 times to average results and consisting of roughly 25 rounds of retraining, adding up to 3000 training runs to completion. Our goal was not to champion any one approach, but to ask if there was any consistent story at all: *can active learning be applied on a new dataset with an arbitrarily architecture, without peeking at the labels to perform hyperparameter tuning?* To our surprise, we found that across many tasks, both classic uncertainty sampling and Bayesian approaches outperform i.i.d. baselines and that DO-BALD and BB-BALD consistently perform best.

References

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural networks. In *International Conference on Machine Learning (ICML)*.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the and shared task: Semantic role labeling. In *Computational Natural Language Learning (CoNLL)*.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145.

Meire Fortunato, Charles Blundell, and Oriol Vinyals. 2017. Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798*.

Yarin Gal and Zoubin Ghahramani. 2016a. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *International Conference on Learning Representations (ICLR) Workshop Track*.

Yarin Gal and Zoubin Ghahramani. 2016b. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning ICML)*.

Yarin Gal and Zoubin Ghahramani. 2016c. A theoretically grounded application of dropout in recurrent neural networks. In *Neural information processing systems (NIPS)*.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian active learning with image data. In *International Conference on Machine Learning (ICML)*.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Association for Computational Linguistics (ACL)*.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Peiyun Hu, Zachary C Lipton, Anima Anandkumar, and Deva Ramanan. 2018. Active learning with partial feedback. *arXiv preprint arXiv:1802.07427*.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Neural Information Processing Systems (NIPS)*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Empirical Methods in Natural Language Processing (EMNLP)*.

D Kinga and J Ba Adam. 2015. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*.

Zachary C Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. 2018. BBQ-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. *Association for the Advancement of Artificial Intelligence (AAAI)*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *arXiv preprint arXiv:1603.01354*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Association for Computational Linguistics (ACL)*. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Carl Edward Rasmussen. 2004. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer.

Dan Roth, Chad M Cumby, Xin Li, Paul Morie, Ramya Nagarajan, Nick Rizzolo, Kevin Small, and Wen-tau Yih. 2002. Question-answering via enhanced understanding of questions. In *Text Retrieval Conference (TREC)*.

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. 2018. Deep active learning for named entity recognition. In *International Conference on Learning Representations (ICLR)*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Computational Natural Language Learning (CoNLL)*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Ye Zhang, Matthew Lease, and Byron C Wallace. 2017. Active discriminative text representation learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*.