CorrActive Learning: Learning from Noisy Data through Human Interaction

Ramesh Nallapati, Mihai Surdeanu and Christopher Manning
Natural Language Processing group
Department of Computer Science
Stanford University
{nmramesh,mihais,manning}@stanford.edu

Abstract

We introduce a new framework of supervised machine learning called *CorrActive Learning*, short for Corrective Active Learning. Similar to active learning, this setting involves learning through human interaction. However, unlike active learning which aims to acquire labels for unlabeled examples, corrActive learning addresses the problem where the set of training data provided to the supervised learner is noisy with respect to its labels. In this scenario, the objective is to accomplish the following two related goals simultaneously, using minimal assistance from the user : (a) clean up the noisy labeled data and (b) improve the performance of the supervised learner.

As a solution, we present a simple algorithm that learns initially from the noisy labeled data, and proceeds to correct the labeling errors in the data iteratively by presenting to the user only those examples that are most likely to be mislabeled, and simultaneously learning from the corrected examples.

Our preliminary experiments involving a human suggest that the new corrActive learner significantly improves the performance of a supervised classifier learned on noisy data. In addition, our synthetic experiments show that the corrActive learner is able to learn much faster than a learner that chooses examples using random sampling, when the labeling error rate is low to moderate ($<$ 25%). At all error rates, the corrActive learner is able to identify the mislabeled examples much better than the random sampler.

1 Introduction

Supervised learning is one of the most successful machine learning approaches used widely in problems such as information extraction, pattern classification and data mining. In this scenario, the user provides the system with a *training set* of several examples $\mathcal{X} = \{x_1, \cdots, x_n\}$, where each x_i is explicitly labeled with its true-category $y_i \in \mathcal{Y}$. Using these as the input, the system then learns a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps any new example $x \notin \mathcal{X}$, drawn from the same distribution as the training set $\mathcal{X}$ to its correct label y.

A supervised learner depends on good quality labeled data to learn the classification function. However, employing domain experts to annotate data with labels is in general not only time consuming but also financially expensive, besides being potentially non-scalable if a large number of annotated examples are needed.

In the recent past, researchers are moving towards procuring annotations from a number of non-experts as a cheaper and scalable method for labeled data acquisition. Broadly, there are two different ways to obtain non-expert annotation: (a) by framing annotation tasks as fun online games, and enticing users to annotate for free [von Ahn *et al.*, 2006] and (b) by posting the task online and seeking non-expert annotations using a pay-per-example setting as pioneered by the Amazon Mechanical Turk system[1].

Both methods help reduce costs significantly, but the downside is that the acquired label data is typically noisy. This may result in a degradation of the supervised learner performance compared to one trained on expert-annotated data. Despite this fact, we believe noisy label acquisition is here to stay, due to its attractive cost benefits. Hence, it is very important to solve the problem of classifier degradation in a noisy training scenario. As a solution, we propose a new *corrActive learning* setting that corrects the labeling noise and thereby improves classifier performance using minimal user assistance.

The rest of the paper is organized as follows. In section 2, we present the past work done on modeling noisy data and also work done in the related area of active learning. Section 3 describes the simple model we used in this work. We describe both the experiments performed with humans as well as synthetic experiments in section 4. We wrap up the discussion in section 6 with a few pointers towards future work.

2 Comparison with Past Work

In this section, we first present past research that addresses the problem of noisy data. Then we present past work done in the field of active learning, and compare and contrast that work from ours.

[1] See www.mturk.com.

2.1 Learning from Noisy Data

The problem of learning from noisy data is not new. Researchers addressed this problem even in the early days of Machine Learning when the focus was on learning symbolic rules for classification. For example, [Schlimmer and Jr., 1986] described a program called STAGGER that incrementally creates new concepts, and adds weights to thse concepts given noisy instances. [Angluin and Laird, 1988] presented a polynomial time algorithm that identifies concepts in the form of k-CNF formulas when the labeling error rate is less than half.

Another noisy data setting that reseachers addressed in the past is one where multiple non-expert annotations per example are available. For instance, [Dawid and Skene, 1979] introduce an EM algorithm to simultaneously estimate annotator biases and latent label classes. However, [Albert and Dodd, 2004] review several related models and argue that they have various shortcomings and stress on the importance of having a gold standard labeled data. In response to this observation, [Snow *et al.*, 2008] used a statistical classifier to predict the true label for each example, given multiple annotations. This approach assumes the availability of a small pool of gold-standard expert-annotated data, on which the classifier trains.

In contrast, our work addresses the setting where there are not necessarily multiple annotators per example, and there is no prior gold-standard data. This is indeed a very likely scenario in the modern Mechanical Turk era. Clearly, previous approaches fail in this situation.

In our approach, we first learn a classifier from the noisy training data, and then iteratively present only potentially mislabeled examples to the user while also learning from the user's corrections. In this setting, we believe that the user does not even need to be a domain expert in most cases. Since our framework forces the user's attention on only potentially mislabeled examples, the user is less likely to make errors due to oversight, hence a non-expert would also be able to do a reasonable job. Even if a domain expert needs to be employed in this setting, costs can still be kept at a bare minimum by an efficient algorithm that accurately identifies the mislabeled examples.

We will also discuss past work on active learning in the next subsection and contrast how our proposed approach differs from standard active learning.

2.2 Active Learning

Active learning for classification was introduced by [Lewis and Gale, 1994]. In this setting, the learner has access to a set of unlabeled examples in addition to the labeled training data. The learner initially learns the classification function from the labeled data and then iteratively requests label from the user for an example sampled from the unlabeled pool. The goal of active learning research is to sample these unlabeled examples in such a way that minimizes the number of requests and maximizes the classifier performance. A representative paper in this line of research is that of [Tong and Koller, 2002], which presents both theoretical and experimental analysis of active learning using SVMs. The work in active learning that comes closest to ours is that of [Balcan *et al.*, 2006], which addresses the issue of label noise in an active learning framework.

1. **Input:** Training Data $\mathcal{X}$ with noisy labels $\mathcal{Y}$, n, number of examples to be presented to the user per iteration.
2. Train a base-classifier on $\mathcal{X}$.
3. Set of examples presented to the user $\mathcal{U} = \{\}$.
4. Compute $\mathcal{M}$, an ordered set of potentially mislabeled examples with confidence scores such that $\mathcal{M} \bigcap \mathcal{U} = \{\}$.
5. while $\mathcal{M} \neq \{\}$ and $\mathcal{U} \neq \mathcal{X}$:
   (a) present $\mathcal{M}_n$, the top n examples from $\mathcal{M}$ to the user for label correction.
   (b) $\mathcal{U} = \mathcal{U} \bigcup \mathcal{M}_n$
   (c) if labels are corrected:
      i. update the labels $\mathcal{Y}$.
      ii. retrain the classifier.
   (d) Recompute $\mathcal{M}$.
6. **Output:** Training data with updated labels $\mathcal{Y}$ and new classifier trained on the updated labels.

Table 1: CorrActive Learning Algorithm.

Both active learning and the new corrActive learning frameworks are based on interactive feedback from the user. However, they are complementary in nature: while the former focuses on acquiring new labeled data from the user, the latter focuses on correcting the existing noisy labeled data with user's assistance.

3 CorrActive Learning Algorithm

The corrActive learning setting is presented in algorithmic form in Table 1. The algorithm iteratively presents potentially mislabeled examples to the user and relearns the classifier based on the updated labels.

The most important aspect of the corrActive learning algorithm is step (4) in the table, which involves estimating mislabeled examples. In this work, we use the cost of misclassification as an estimate of mislabeling score. For a binary classification problem, the misclassification cost for an example x with label y is defined as $1 - P(y|x)$. For the conditional probability $P(y|x)$, we use the estimate given by the logistic regression model.[2] To limit the number of examples in the set $\mathcal{M}$, we consider only the misclassified examples (as seen by the current classifier on the current label set) as candidates for mislabeled examples.

The intuition behind using this metric is the following: if the classifier learns a reasonably good decision boundary, then it may be safe to assume that it considers the mislabeled examples as outliers and misclassifies them. This assumption may hold well when the labeling noise is small. However, there is a possibility of the classifier overfitting the noisy data,

[2]The mislabeling score, presented here is based on a probabilistic model, but it can be easily defined for a non-probabilistic model such as an SVM as well

particularly in high noise situations. In such cases, the classifier may not be able to detect mislabeled examples accurately. Indeed, our experimental results discussed in section 5 demonstrate this to be the case.

4 Experiments and Results

4.1 Data

We perform all our experiments on the binary classification task of tagging legal docket entries as positives or negatives for *Claim Construction Orders (CCO)*. To elaborate, a legal docket is a list of brief notes usually written by the court clerk, stating what action was taken on a given day regarding a particular case. Each entry in the docket is usually a short, natural language text snippet. Claim Construction Order is an important action taken by the court in patent litigation lawsuits where the judge issues an order that interprets the claims of litigated patent(s). This is considered an important milestone in a patent lawsuit that potentially tilts the balance of the case, if the court's interpretation of patent claims is in favor of one of the litigating parties.

We collected a total of 7,042 docket entries across thousands of cases all over the U.S. federal circuit courts that are related to claim construction orders. All of them are hand-tagged positive or negative by non-experts, so it is expected that there are several mislabeled examples. Out of the 7,042 docket-entries, 1,207 were labeled as positive.

We present a few positive and negative examples for CCOs in Table 2. It is clear from the table that it is not straight forward to distinguish true CCOs from related events. There are several actions surrounding claim construction orders such as proposals, orders for reconsideration, services, etc. that can confuse a classifier, as well as a distracted annotator.

In this work, we used a standard logistic regression with Gaussian prior as the classifier. For preprocessing, we did case-folding and removed stopwords, but did not do any stemming. As features, we used unigrams, bigrams and biterms[3]. We did not do any automatic feature selection, but we discarded all features that occurred in less than 5 examples.

4.2 Real-life Experiments

In our first experiment, we ran the corrActive learner system using a legal domain expert as our human supervisor. As described in section 3, the system first trained on the initial set of labels. Next in each user interaction, it presented four most misclassified examples to the human for potential label correction. If the label(s) of any of the examples has been corrected by the user, the system retrained itself and the cycle repeats until either there are no more documents to be presented, or the classifier finds no more misclassified examples.

In all, the system ran for about 40 iterations and presented about 160 documents to the user. The user corrected the labels of about 100 of those examples. Figure 1 plots the 10-fold cross-validation F1-performance[4] of the classifier on our data as a function of number of iterations. It is clear from the plot that the performance steadily improves as the user corrects more and more labels.

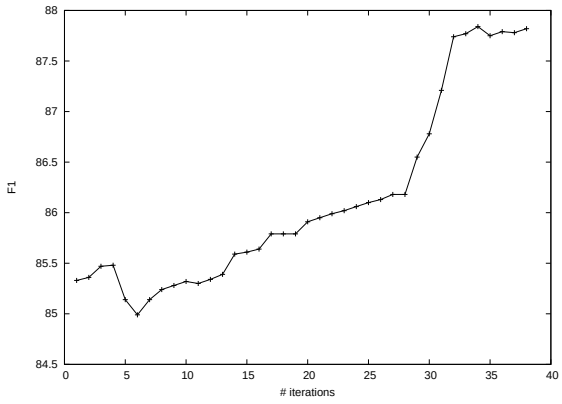

Figure 1: Cross-validation F1 performance of the corrActive learner.

We note that the results are only preliminary, since they are performed by just one user, and only in one sitting. We need to perform more extensive and repeated experiments to arrive at strong conclusions. Nevertheless, we believe these results suggest the utility of the corrActive learner system.

4.3 Synthetic experiments

The experiments above were done on a dataset where the initial performance of the classifier was already very high (about 85%). Also, the number of labeling errors corrected by the user was very small (about 1.5% of the total data). It was not clear if the performance of the corrActive learner would receive a similar boost if there were more errors in the data.

To analyze the performance of the system in the presence of higher labeling errors and lower initial classifier performance, we performed synthetic experiments. We assume that the labeled data corrected after the real active label correction system is perfect[5].

We then corrupt the labels of the data randomly with increasing levels of noise. Since we know the true labels, we were able to simulate the active label correction system by replacing the human with an autmoatic oracle that has the true labels.

For the experiments reported below, we split the data randomly in a 70-30 ratio into training and test sets respectively. As a baseline, we used a setting where the learner samples examples randomly for label correction. This setting, helps us

[3]We call any unordered-pair of words that occur in the same document, a *biterm*. We can afford to use biterms without exploding the feature space because our documents are short snippets of text.

[4]F1 is the harmonic mean of precision and recall, and is a standard evaluation measure in text classification. Please see [Lewis *et al.*, 2004] for more information.

[5]Note that this is not necessarily true – a perfect classification does not necessarily mean perfect labeling. Nevertheless, it is a good approximation in this case.

Positive Examples

1. Order Construing Claims by Judge M. J. Lorenz: The disputed terms are interepreted as set forth in this order. (dkt clerk) (Entered: 12/02/2005)
2. ORDER: the contested patent claim language in the '903 and '008 patents shall be construed consistently with the Memorandum Opinion (D.I. 180) (signed by Judge Mary P. Thynge) copies to: cnsl. (ntl) (Entered: 08/20/2002)
3. MEMORANDUM AND ORDER RE: Patent Claim Construction. Signed by Judge Marvin J. Garbis on 8/16/05. (mcb, Deputy Clerk) (Entered: 08/16/2005)
4. The interpretation of Claim 1 of US Patent #5,282,613 is entered in accordance with dfts' markman claim construction order. See written Order. (cc: all counsel) (KM, ilcd) (Entered: 04/03/2003)
5. The court accepts the magistrate judge's recommendation that independent claim 11 is invalid pursuant to 35 U.S.C. 112, and declares claim 11 invalid. Finally, the court rejects the magistrate judge's use of the words "transmission data" on page four of the First Report as a typographical error, and replaces it with "transmission data." All other claim terms are to be given their ordinary and customary meaning. (See Order for specifics) (Signed by Judge Sam A Lindsay on 9/10/07) (skt) Modified on 9/11/2007 (jyg). (Entered: 09/11/2007)

Negative Examples

1. CERTIFICATE OF SERVICE Preliminary Proposed Construction of Claim Terms by Shaw Industries, Inc., .(Zidar, Bernard) (Entered: 04/11/2006)
2. PROOF OF SERVICE of Order re: Claim of Interpretation filed 6/20/02 [158-1] to counsel Steven Paganetti for dfts Paboojian (rab) (Entered: 07/10/2002)
3. ORDER DENYING STIPULATED PROPOSAL FOR PROCEDURE TO RESOLVE DISPUTED CLAIM INTERPRETATIONS: This matter is before the Court on Plaintiff and Defendants' stipulated proposal for a procedure to resolve disputed claim interpretations before trial. (Entered: 06/25/2007)
4. ORDER denying defendants' motions for reconsideration of : (1) claim construction order and finding of infringement of claim 8 of the '725 patent' (ys,) (Entered: 01/02/2003)
5. ORDER granting 100 Motion for Leave to File Excess Pages for Defendants' Joint Answering Brief on Claim Construction. Signed by Judge Leonard Davis on 1/15/08. (mjc) (Entered: 01/15/2008)

Table 2: Representative examples for Claim Construction Orders.

in estimating the utility of using misclassification cost as the metric for sampling examples. In each interactive iteration, we provided exactly 10 documents to the oracle for label correction.

We measured the performance of the system on the following metrics:

- **Classification F1:** We report the F1 performance of the classifier on the test set as a function of number of interactive iterations.
- **Mislabel retrieval F1:** This metric captures the overlap between the examples presented by corrActive leaner for label correction, and the truly mislabeled examples. More specifically, imagining the task of the corrActive learner as one of retrieving mislabeled examples, we computed the recall, precision, and their harmonic mean F1 of the mislabeled examples at the end of each interactive session. We call this metric Mislabel retrieval F1. Higher mislabel retrieval F1 is more desirable, because it implies higher overlap between retrieved and truly mislabeled examples.

5 Results and Discussion

The experimental results, presented in Figure 2, show the following trends.

1. CorrActive learner improves the performance of the base classifier by correcting labeling errors and thereby reducing the noise in the data (row 1 in Figure 2).
2. CorrActive learner is always better than a random sampler in detecting mislabeled examples as shown by the mislabel retrieval plots (row 2 in Figure 2).
3. At low error rates, the corrActive learner converges very quickly. For example, at 20% error rate, when there are approximately 5000 examples in the training set (and 1000 mislabeled examples), the corrActive learner displays only about 820 examples for labeling correction before converging and boosting its classification performance by 9.7%. The precision of the corrActive learner in retrieving mislabeling examples is high ($\sim$ 80%). This indicates the effectiveness of this technique compared to the overhead of manually inspecting the examples for potential labeling errors.
4. The mislabel retrieval effectiveness of the corrActive learner drops as the labeling noise is higher. This is due to overfitting as discussed in section 3.
5. At the initial stages of user interaction, corrActive learner is not as good as the random sampler. However, after a few iterations, the corrActive learner accelerates its learning rate, and surpasses the random sampler, particularly when the labeling error rate is low ($< 25\%$).

   The last observation is a very interesting and somewhat unexpected phenomenon and deserves more attention. The reason for this behavior remains to be fully understood, but we hypothesize as follows: when there is random noise in the data, the classifier performance is de-

graded because it is not able to learn the correct decision boundary. Hence, it makes sense for the classifier to focus on examples near the decision boundary, to correct itself. When the density of examples near the decision boundary is high, random sampling may be able to retrieve these examples and potentially help the classifier learn a better boundary. CorrActive learner, on the other hand, always focuses on retrieving the most mislabeled examples at all times. During the initial stages, this may result in retrieval of the easy mislabeled examples, which are far from the boundary. These examples, although useful, do not inform where the decision boundary is. Therefore, the corrActive learner fails to learn as well as a random sampler in the initial stages. However, once the corrActive learner gets past the easy examples, it learns much faster than random sampling. This suggests using random sampling at first and switching to corrActive learning later, but we have not explored it in this paper.

6 Conclusions and Future Work

In this paper, we introduced the problem of corrActive learning and presented a simple approach to address this problem. Our preliminary experiments involving a human, as well as synthetic experiments using varying noise levels strongly suggest the utility of this approach.

We believe this work opens the door for several interesting directions for future work. Some of the important open questions are:

- Given a choice between correcting the label of a mislabeled example (corrActive learning), and labeling an unlabeled example (active learning), which is more advantageous in terms of classifier performance?
- What is the best way to combine active learning and corrActive in the same framework?
- Are there better approaches to estimating mislabeled examples than misclassification cost? In particular, can we combine random sampling and corrActive learning to improve the performance of either technique?

We hope to answer these questions as part of our future work.

Acknowledgments

We thank the anonymous reviewers for their very useful suggestions and feedback.

References

[Albert and Dodd, 2004] Paul S. Albert and Lori E. Dodd. A cautionary note on the robustness of latent class models for estimating diagnostic error without a gold standard. *Biometrics*, 60:427–435, 2004.

[Angluin and Laird, 1988] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2, 1988.

[Balcan *et al.*, 2006] M. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *International Conference on Machine Learning*, 2006.

[Dawid and Skene, 1979] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28:20–28, 1979.

[Lewis and Gale, 1994] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, page 312, 1994.

[Lewis *et al.*, 2004] D. D. Lewis, Y. Yang, T. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[Schlimmer and Jr., 1986] Jeffrey C. Schlimmer and Richard H. Granger Jr. Incremental learning from noisy data. *machine Learning*, 1, 1986.

[Snow *et al.*, 2008] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP*, 2008.

[Tong and Koller, 2002] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2002.

[von Ahn *et al.*, 2006] Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: A game for collecting commonsense knowledge. In *ACM Conference on Human Factors in Computing Systems*, 2006.

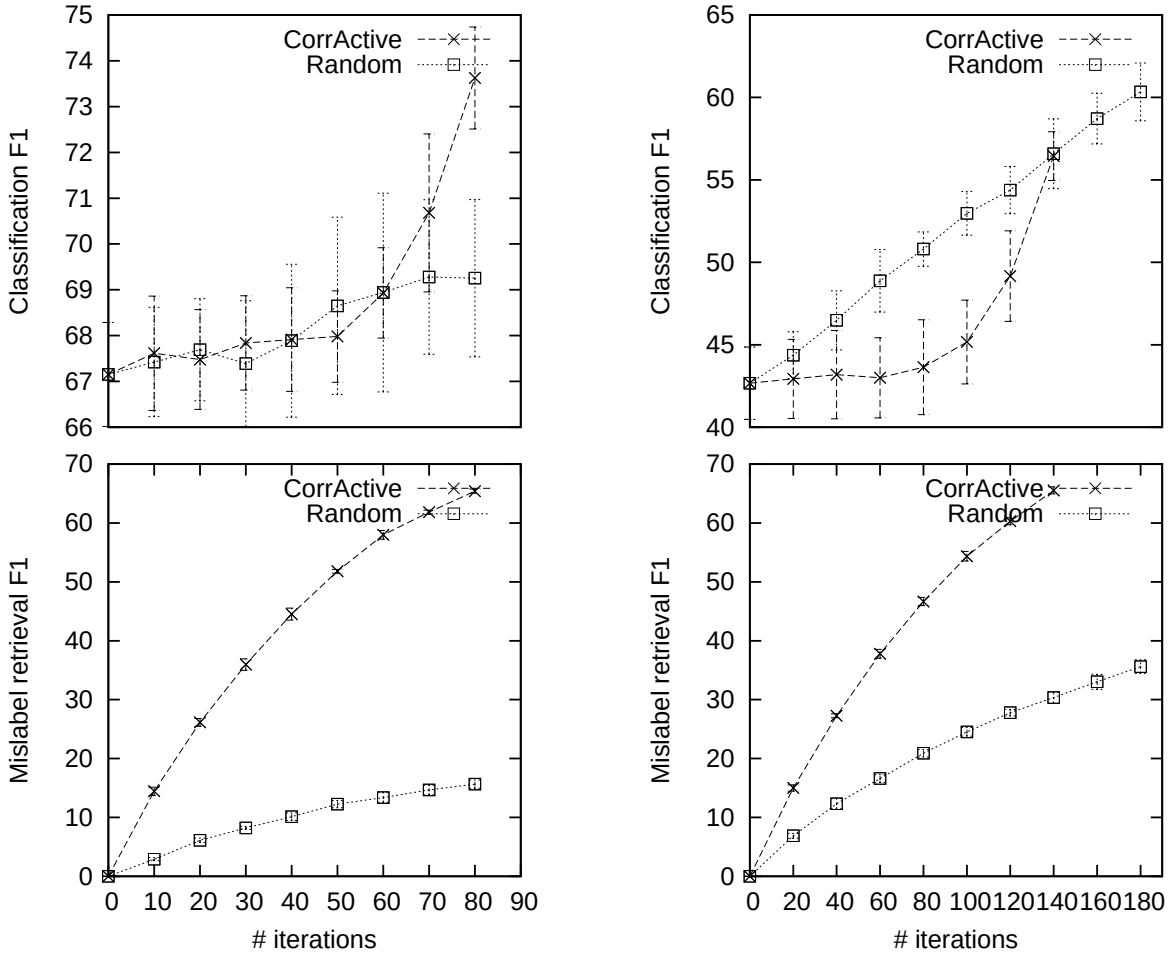

Figure 2: Experimental results: top row shows results on classification F1 on the test set and the bottom row displays results on mislabel retrieval F1 on training set, both as a function of number of iterations. Each iteration consists of a user interaction session in which 10 documents are shown to the user for label correction. Each data point in the figure is a result of averaging across 5 runs and the error bars are two standard deviations wide. Left column is for error rate of 20% while the right column is for 40% error rate. As is evident from top left box, corrActive learning is not better than random sampling at first, but quickly recovers and surpasses random sampling at low error rates. At high error rates (top right box), corrActive learner converges before surpassing random sampler. However, as shown in the bottom row, corrActive learning is always better than random sampling in identifying mislabeled documents.