DiAd: Domain Adaptation for Learning at Scale

Ziheng Zeng
University of Illinois
Urbana-Champaign, IL
zzeng13@illinois.edu

Snigdha Chaturvedi
University of California
Santa Cruz, CA
snigdha@ucsc.edu

Suma Bhat
University of Illinois
Urbana-Champaign, IL
spbhat2@illinois.edu

Dan Roth
University of Pennsylvania
Phialdelphia, PA
danroth@seas.upenn.edu

ABSTRACT

Massive online courses occupy an important place in the educational landscape of today. We study an approach to scale predictive analytic models derived from online course discussion fora–specifically that of confusion detection–onto other courses. The primary challenge here is the lack of labeled examples in a new course and this calls for unsupervised domain adaptation (DA). As a first step in exploring DA in the education domain, we propose a simple algorithm, DiAd, which adapts a classifier trained on a course with labeled data by selectively choosing instances from a new course (with no labeled data) that are most dissimilar to the course with labeled data and on which the classifier is very confident of classification. Our algorithm is empirically validated on the confusion detection task across multiple online courses. We find that DiAd outperforms other methods on the target domain, while showing a comparable performance to a popular method that uses labeled data from the target domain.

CCS CONCEPTS

• **Information systems → Data analytics**; • **Computing methodologies → Unsupervised learning**; • **Applied computing →** *E-learning*;

KEYWORDS

Domain Adaptation, Learning at Scale, Confusion Detection

ACM Reference Format:
Ziheng Zeng, Snigdha Chaturvedi, Suma Bhat, and Dan Roth. 2019. DiAd: Domain Adaptation for Learning at Scale. In *The 9th International Learning Analytics Knowledge Conference (LAK19), March 4–8, 2019, Tempe, AZ, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3303772.3303810

1 INTRODUCTION

Massive online learning platforms are ubiquitous and have transformed the educational landscape in ways that have never been possible before. This transformation has led to numerous contemporary studies focusing, among other areas, on inferential and predictive learning analytics derived from the clickstream and discussion fora to understand learner behavior and disengagement, and to improve learning outcomes.

Discussion fora constitute an important online course component that students use to interact with each other, to seek help, as well as to express their sense of satisfaction/dissatisfaction with the course. Numerous studies have highlighted the benefits of processing forum posts to provide key data-driven insights into learning behaviors and guide timely interventions, including [2, 5, 17–20]. The predictive analyses in these works rely on developing supervised machine learning models. With the rapid proliferation of courses in a variety of learning-at-scale platforms and the lack of immediate access to labeled data, developing supervised machine learning models to automate decision-making based on discussion fora can be resource-constraining–be it in the context of a subsequent offering of the same course, or the offering of a new course altogether. This calls for efficient mechanisms for *unsupervised* Domain Adaptation–the process of adapting a classifier trained for a *source* course to a *target* course without using any manually labeled training data from the target course. Particularly, in this paper, we study the problem of domain adaptation in the specific area of at-scale discussion forum analysis–that of *confusion detection.*

Confusion detection refers to the binary classification problem of automatically identifying if a given forum post expresses confusion or not. Because of the important role of discussion fora in affecting learner satisfaction and outcomes in a course, timely and efficient detection and resolution of confusion not only helps in bringing instructor immediacy to online courses but also positively impacts learner experience. Given the issues of scalability in online course platforms, automating the decision-making process brings about the much needed efficiency in aspects needing instructor immediacy, which in turn requires the use of scalable and adaptive algorithms. Thus, mechanisms to easily extend models developed for one course over to other courses are of fundamental necessity.

In the recent years, a number of studies [2, 24] have focused on the tasks of confusion detection and identified the need for domain adaptation (DA). Despite recognizing this need and DA not being a novel idea, the effect of DA in the area of online courses remains largely under-explored.

Domain adaptation is a challenging task especially in the discussion fora domain because different courses demonstrate different

LAK19, March 4–8, 2019, Tempe, AZ, USA

ACM ISBN 978-1-4503-6256-6/19/03…$15.00
https://doi.org/10.1145/3303772.3303810

Table 1: Sample posts labeled as *Confusion* from the three courses to highlight the difference in the nature of the three domains

Course	Example post
HS	I had this on test all wrong as I must have not understood anything. 1. housing market: i put shift in demand curve, apparently wrong. I supossed less income would cause less people wanting to buy a house making them cheaper. 2. Tea market. if one of the resources to make tea is running low the supply would be affected not the demand. more people wont buy tea because a drought in Brasil. I had almost all wrong :(
EDU	I can draw and reason out the cubes, but the algebraic notation is far beyond my skill level working alone. I would like to participate in an exercise like this to see if I could contribute. I guess I have been in elementary math too long! I had to watch this video several times, and came away with the onion peeling theory that could be adapted for my students and my understanding. It was very frustrating. I still do not have it all.
MED	When calculating the normal approximation for the binomial do we use half intervals e.g. 111.5 instead of 112 if we're looking for the area to the right in the curve; and can we use a different normal distribution applet than the ones recommended. If so, what are the answer tolerances. Can one be several tenths of a percentage point off. This ambiguity is killing me. Literally, I'm dying.

Figure 1: Wordcloud of confused posts from EDU, MED and HS (in order from left to right). All courses contain different words related to course-specific topics making it challenging for a classifier trained on one course to perform well on another course.

idiosycracies. For example, consider the task of confusion detection. Table 1 shows examples of posts that express confusion from three different types of courses broadly classified as Humanities (HS), Education (EDU) and Medicine (MED). We can see that they share some commonalitites, e.g., *confused* posts often express frustration, but the way of expressing this frustration and confusion exhibits tremendous variability. This variability can depend on factors including the course topic, instruction style, level of formality of the forum and background of learners. While the commonalities present a promise for domain adaptation methods, the differences demonstrate the critical need for careful thought and experimentation in designing these adaption procedures. For example, in the Medicine domain, several confused posts discussed technical problems related to the topics in 'probability and statistics' aspects of the course. Hence, any statistical machine learning system operating on related posts would learn to associate words related to the topics of probability and statistics with the *confused* class. This is further illustrated in Figure 1, by way of word-clouds of only the *confused* posts for the three courses. We can see that in posts from the EDU course, the prominent words are specific to the Education domain and include *students, math, school, teaching, grade*. Similarly, for the MED course the prominent words include *data, mean, valu(e), standard (deviation), sample, calcul(ate), number* and for HS posts these include *data, issu(e), women, girls*. In summary, we can see significant differences in distributions of words among confused posts from different courses.

Naturally, a classifier trained on the vocabulary from one course will be limited to course-specific features and patterns of that course and have limited generalizability. That is, such course-specific regularities may not translate successfully to other courses (for instance, to HS where 'probability and statistics' may not even be a relevant topic and the most confusing topic could be related to, say, 'markets' or 'women issues'). Hence, a classifier trained on labeled data from Medicine courses can not be expected to perform well on data from Humanities (we also demonstrate this quantitatively in our experiments). This constitutes the primary technical challenge of the problem considered here, i.e., the ability to tune a classifier, which is trained on a particular course, to learn not only those patterns indicative of confusion of that course, but also of a new course.

Fortunately, there could be linguistic commonalitites, such as expressions of frustration or presence of keywords like *question* (as indicated by the word-clouds of the three courses), which could help the model in identifying *some* posts expressing confusion in the new course. Thereafter, statistical regularities and correlations found in these newly identified posts could be harnessed to learn confusing course-specific patterns in the new course. The resulting classifier is now more general, with the increased capability achieved without using any labeled data from the new course.

With this motivation in mind, we propose DiAd, an algorithm that *differentially adapts* the source-trained classifier to the target domain by relying on instances that it is most confident on, and those that are dissimilar to the source. While the proposed methodology for adapting a classifier is independent of its goal, here we validate its applicability for the purpose of confusion detection. To the best of our knowledge this study is a first step in the direction of DA for educational domain, while also being an effort to draw the community's attention to this important and under-explored problem. In this sense, this constitutes a unique contribution to existing literature on discussion forum analysis and associated domain adaptation.

We summarize our contributions below:

- In the case of confusion detection, we observe that the source and target domains tend to have similar features. This means that a transformation to resolve the difference between feature space statistics alone is not sufficient for domain adaptation; what is additionally needed is the transformation

of the labels given the features. We achieve this by the use of surrogately labeled instances to slowly move the source model towards the target domain.
- We propose a self-training algorithm, DiAd, to adapt the supervised model of confusion detection to a new course, thus rendering it more widely applicable.
- We demonstrate that existing unsupervised DA techniques like hassle-free domain adaptation (HF) [23] and some of the popular supervised DA methods modified to work in an unsupervised setting [6] are largely inadequate for the confusion detection task. We quantitatively demonstrate the efficacy of our approach by comparing it with variations of existing DA techniques using data from multiple courses.
- We find that DiAd not only adapts a classifier to new domains without using any labeled data, the resulting classifier does not compromise on its performance in the source domain. This is especially important in the education domain, where, while adapting a classifier to a new course, we would still like to use it for making predictions in the source course (for instance, future offerings of the same course).

2 RELATED WORK

We situate this work at the intersection of two active research areas: analytics derived from course discussion fora and unsupervised domain adaptation methods.

Analytics from MOOCs and discussion fora: Empirical studies based on online course discussion fora include understanding learner persistence by mining their sentiments expressed in discussion forum content (e.g., [20]), analyzing how fora evolve [15], predicting student performance from emotional expressions [22], predicting instructor interventions [5], predicting learner performance using engagement patterns [17], recommending forum threads to users [21], and detecting confusion and frustration [2, 24]. Other studies include detecting indices of cognitive presence of learners [12], understanding help-seeking behaviors and new principles for supporting help-seeking [11], discovering the importance of social presence in learning experience [7, 9], understanding the perceptions of social presence in diverse MOOC populations [16], and more general studies focussed on understanding and improving engagement [8, 10, 14]. Depending on the task, a few of the findings have been valid more generally in more than one course regardless of the course, while others (e.g., [2] and [24]) have explicitly identified the need for domain adaptation in order to render the tasks more widely applicable (e.g., across different courses offered on the same platform). Accordingly, our present study is a natural extension of prior work and addresses a critical need in this area.

Domain adaptation methods: This work is similar in spirit to the unsupervised DA approach proposed in [3, 23], where a subset of the target instances gets added to the training set during the adaptation stage, but differs in the manner in which the subset is selected. Moreover, as will be evident from the baselines considered, our approach can be likened to the more popular supervised adaptation methods such as Frustratingly easy and Source-and-Target [6] (described in Section 4.3), with the difference being the use of surrogately labeled target instances in place of their true labels.

We would like to point out that our approach is different from the structural correspondence learning approach [4], which automatically induces correspondences among features from the source and target domains. We do not rely on the correspondence at the feature level, but instead harness the dissimilarity of a collection of confidently predicted target instances to enable the adaptation.

3 APPROACH

In this section we describe two variations of our domain adaptation approach, DiAd, DiAd-Radius and DiAd-Sample, in detail. Our setting in both cases is that of unsupervised domain adaptation. In other words, we assume that we have labeled data, $\mathbb{L}$, in the source domain but unlabeled data, $\mathbb{U}$, in the target domain. Like all domain adaptation settings, the label set in the source domain is the same as the (desired) label set in the target domain. The goal is to output a trained classifier, f, which has been adapted to the target domain.

The underlying idea behind our iterative self-training approach is that the source and target courses (domains) are similar, and a classifier trained on the source course would perform reasonably well on the target course. The goal of this work is to improve the performance of this basic classifier. We use a classifier trained on labeled data from the source course to get *surrogate* labels on the unlabeled instances from the target domain. These labels are so termed because they are not ground-truth labels but are, instead, predicted by the existing classifier. The goal in both the variants presented here is to use these surrogately labeled target instances, in addition to the labeled source instances, to further train the classifier. However, instead of overwhelming the classifier with the source and target data all at once, we *slowly* adapt the classifier to the new domain. For achieving this, we introduce the surrogately labeled target instances iteratively. In each iteration, we introduce only a subset of these instances to the existing training set.

Understandably, the performance of the final classifier would depend on the (surrogately labeled) instances that were added to the training set. Therefore, it is important to carefully select this subset of instances introduced to the training data in each iteration. For choosing this subset, we adapt the classifier *differentially* to the instances of the target domain. Specifically, we choose instances that satisfy the following two constraints:

(1) the classifier should have been very confident while assigning the respective surrogate labels, and
(2) the instances in the subset should be *dissimilar* to those in the existing training set.

The first constraint encourages the addition of good quality 'labeled' instances by adding instances that are likely to be correctly labeled as judged by the existing classifier. However, intuitively, it is likely that the classifier would be more confident while labeling instances that are similar to the ones it has already seen in the training set. Therefore, simply adding these instances to the training set would encourage the classifier to relearn what it already knows. What is additionally needed is to introduce the classifier to *new* types of instances that are different from the source domain and are representative of the idiosyncrasies of the target domain. The second constraint attempts to capture this notion by choosing instances that are most dissimilar to the existing training set. In the rest of this section we first describe the general algorithmic framework

and then describe the two variations of our method: DiAd-Radius and DiAd-Sample.

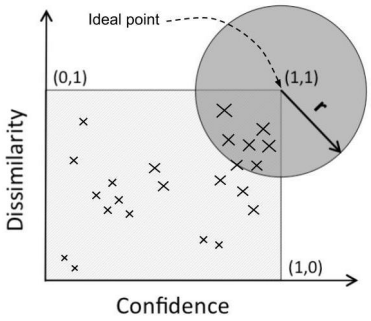

Figure 2: Pictorial representation of the process of choosing surrogately labeled instances to the existing set of labeled instances. Each instance is represented by its predicted confidence on the x-axis and its dissimilarity from the training set on the y-axis (both normalized). All points depicted by x lie in the shaded rectangle. (1,1) is the ideal candidate for addition to the training set. DiAd-Radius chooses all points that lie inside the circle. DiAd-Sample samples points with probabilities proportional to the distance from the ideal point. In this representation, the size of a point is proportional to this probability.

3.1 General Algorithmic Framework for DiAd

Algorithm 1 presents the pseudocode of our general approach. The algorithm accepts labeled data, $\mathbb{L}$, and a set of unlabeled instances, $\mathbb{U}$, as inputs. These sets are initialized with data from the source and target domains respectively. Our iterative algorithm proceeds by training a classifier (such as a Logistic Regression) on the labeled set, $\mathbb{L}$ (Step 1). It then predicts a label, l_u, and the corresponding prediction confidence, z_u, for each instance, u, in the unlabeled set, $\mathbb{U}$ (Step 3) [1]. For each instance in u, the algorithm also computes its dissimilarity, d_u, from the training set, $\mathbb{L}$ (Step 4). For the sake of readability, we define this dissimilarity measure later in this section.

Hence, for each instance in $\mathbb{U}$, we have: (i) its prediction confidence, z_u, and (ii) its dissimilarity, d_u, from the training set. We first normalize the dissimilarity and confidence values to lie between 0 and 1 (Steps 6 and 7). Now, each instance, u, from $\mathbb{U}$ can be viewed as a point in a two dimensional Euclidean space with confidence, z_u, and dissimilarity, d_u, corresponding to the x and the y coordinates respectively. Note that since the confidence and dissimilarity values were normalized, all points will lie in the shared rectangle shown in Figure 2. The ideal candidate for addition to the training set would be one that has high dissimilarity from the training set and also had a high corresponding prediction confidence. Geometrically, such a point in this space would correspond to the top right corner of this rectangle: (1,1). So, while selecting the set of instances, $\mathbb{H}$, to be appended to the training set of the classifier, we want instances that are as close to (1,1) as possible. The two proposed methods, DiAd-Radius and DiAd-Sample, differ in the way this set $\mathbb{H}$ is selected (selection described later). However, once the set of surrogately labeled points, $\mathbb{H}$, has been chosen, its elements (along with the corresponding surrogate labels) are added to the labeled set $\mathbb{L}$, and are removed from the unlabeled set, $\mathbb{U}$ (Steps 9 and 10). The algorithm then returns to Step 1 using these updated sets, $\mathbb{L}$ and $\mathbb{U}$. This iteration proceeds until there are no more instances left in the unlabeled set $\mathbb{U}$ that can be added.

[1] We extract the same features for the source and target domains, excepting the unigram features, and for this step, we use the vocabulary of the source domain.

Algorithm 1 Training algorithm for DiAd

Input:
Set $\mathbb{L}$ = Labeled data from source domain;
Set $\mathbb{U}$ = Unlabeled data from target domain
Output:
f = Domain adapted classifier

1: Train classifier, f, using $\mathbb{L}$
2: **for each** $u \in \mathbb{U}$ **do**:
3: $(l_u, z_u) = f(u)$ ▷ Surrogately label u using f while outputing prediction confidences, z_u
4: $d_u = dissimilarity(u, \mathbb{L})$
5: **end for**
6: Normalize d_u's ▷ $0 \leq d_u \leq 1, \forall d_u$
7: Normalize z_u's ▷ $0 \leq z_u \leq 1, \forall z_u$
8: Set $\mathbb{H} \subseteq \mathbb{U}$ ▷ Select $\mathbb{H}$ according to DiAd-Radius or DiAd-Sample
9: $\mathbb{L} = \mathbb{L} + \mathbb{H}$
10: $\mathbb{U} = \mathbb{U} - \mathbb{H}$
11: If $\mathbb{H} = \emptyset$ or $\mathbb{U} = \emptyset$ Stop, Otherwise goto Step 1

Choice of classifier, f: While DiAd makes no assumptions about the type of classifier, a key requirement is that the classifier should be able to estimate the confidence of its predictions. In our implementation, we used logistic regression because it yields a natural measure of confidence–the probability of the predicted class. One could conceivably use other classifiers like SVM, where the confidence is estimated by computing the distance from the classificatory hyperplane. However, this experimentation is left to future work.
Computing dissimilarity: In Step 4 of the above algorithm, we also compute the dissimilarity, d_u, of an instance, u, from the training set. As already mentioned, the dissimilarity measure is included to enable the classifier to encounter new 'types' of instances (for each predicted class). We design a simple measure to compute an instance's dissimilarity to other instances in the current training set that belong to the same class. For any instance, u, with a predicted label, l_u, we consider the set of training instances that belong to the same predicted class:

$$\mathbb{L}^{l_u} = \{x | \text{label of } x = l_u, \forall x \in \mathbb{L}\}$$

We then define its dissimilarity, d_u, from the training set as the L_2 distance between the feature vector representations of u and the centroid of $\mathbb{L}^{l_u}$. This measure serves to identify those points that are *different* from other training points belonging to the same class and hence provide the classifier with crucial new information to learn once these new points are added to the training set.

3.2 DiAd-Radius

As mentioned earlier, the two variations of DiAd differ in the way set $\mathbb{H}$ is selected (Step 8 of Algorithm 1). In the first variation,

DiAd-Radius, we select the instances to be included in $\mathbb{H}$ by first constructing a circle centered at (1,1). The circle's radius, r, is a parameter in our model. The points of interest to us would lie in this circle. In other words, the set $\mathbb{H}$ contains all points that lie in the area that represents the intersection of the rectangle and the circle in Figure 2:

$$\mathbb{H} = \{u | u \in \mathbb{U} \text{ and } z_u^2 + d_u^2 \leq r^2\}$$

3.3 DiAd-Sample

This method uses an alternate way to choose the points for set $\mathbb{H}$ (Step 8). It first computes the Euclidean distance, p_u, of each point, u (represented in the Euclidean plane as (d_u, z_u)), from the 'ideal candidate' (1,1) as:

$$p_u = \sqrt{(z_u - 1)^2 + (d_u - 1)^2}, \forall u \in \mathbb{U}$$

It then constructs a set $\mathbb{H}$ of randomly sampled K points from the surrogately labeled set such that the sampling probability of a point, u, is proportional to this distance, p_u.

4 EMPIRICAL EVALUATION

In this section, we present our experiments in detail. We first describe the dataset used in our experiments (Section 4.1) and the features extracted for the classification task (Section 4.2). We then describe our baselines (Section 4.3) and the evaluation set-up (Section 4.4) and compare DiAd's performance with other methods in Section 4.5.

4.1 Confusion Detection Dataset

We conduct our experiments using forum posts from the Stanford MOOC Posts, a corpus composed of 29,604 anonymized learner forum posts from eleven Stanford University public online classes [1]. The courses were broadly classified into three course domains (henceforth simply referred to as courses) by the data curators: Humanities/Sciences (HS), Medicine (MED), and Education (EDU), with about 10,000 posts in each set. Each post was manually annotated with a value indicating the extent to which it expresses confusion, on a scale of 1 (expert knowledge) to 7 (extreme confusion). A score of 4 indicates neither knowledge nor confusion. Following previous work [24], we divided the posts into two groups– "confusion" and "not confusion" based on this score. A score above 4 was considered a *Confusion* post, whereas a score below 4 was regarded as a *Not confusion* one. For our domain adaptation experiments, we considered various ordered pairs of these courses to form the source and target domains pairs.

4.2 Feature Engineering

Our approach, DiAd, essentially adapts a classifier trained on the source domain to the target domain. Our first focus was to extract features that would be used for training this classifier. Towards this end, we used the features that were found to be indicative of confusion as mentioned in [24]. They were grouped into two categories: content-related and community-related features.

Content-related features: These features pertain to the textual content of the post:

(1) Automated readability index (ARI): For a given post, the ARI is a number, which approximates the grade level needed to comprehend the text. The inclusion of this feature was based on the assumption that posts encoding confusion have higher readability indices (i.e., are more difficult to read) than posts that do not encode confusion.
(2) Post length in words;
(3) Unigrams: These binary features encode whether a word occurred in the post or not.
(4) Question mark: Since confusion is often expressed by asking questions, the presence of a question mark in the post was regarded as a feature.
(5) Sentiment Ratio: This feature measures the ratio of the number of positive and negative words in a post. The assumption was that a post expressing confusion would contain proportionately more words with negative sentiments as compared to words with positive sentiment, and hence would have a lower value for this feature than that of a post not expressing confusion. A word was classified as having positive or negative sentiment using a pre-trained lexicon [13].

Community-related features: These constituted a second set of features that measured the reactions of other students enrolled in the course and were available in the dataset. They were: the number of (i) reads and (ii) up-votes of the post. It is likely that a post expressing a confusion or seeking clarification would be of interest to other students taking the course and so would be read more and receive a higher number of up-votes.

4.3 Baselines

We would like to remind the readers that ours was an unsupervised domain adaptation framework (i.e. we assumed that we do not have labeled target data). For the task of confusion detection, we compared the performance of our approach with the following baselines. Also, when applicable, the baselines used the same features and the same classifier as DiAd for a fair comparison.

HF: The Hassle-Free Method (HF) [23] is an unsupervised domain adaptation method that has shown state-of-the-art performance and forms our primary baseline [2]. It randomly selects a subset of target instances, and normalizes them into an exemplar vector. Then each source instance is transformed into a new feature vector by computing its similarity to each instance in the exemplar vector. The new features are then appended to the original instances in the training set to form a new set of training instances.

SrcOnly: In this baseline we use a classifier without any domain adaptation. The classifier is trained only on ground-truth labeled data from the source domain, and tested on the held-out test sets (from source or target domains). While this is not a true domain adaptation baseline, we have included it to test the change in the performance of a model when it is tested on instances from an unseen course. It thus helps to underscore the need for DA.

Source-and-Target (S+T): This is a modification of a simple yet popular *supervised* DA baseline (used in [6]), in which the source and the target labeled data are pooled in together to train a classifier. Since in our case, the target domain does not have labeled data, we treat the predictions of the SrcOnly baseline on the target course data as *surrogate* labels. We apply S+T using

[2] HF is a particularly competitive baseline and has been shown to outperform other unsupervised domain adaptation methods [23]

Table 2: Performance comparison of our approach, DiAd, with baselines (like HF [23]) for the Confusion Detection task. DiAd outperforms other methods on the target domain, and achieves a performance similar to one that could be obtained using labeled data in the target domain.

Source	Target			Baselines			DiAd-Variations			
		SrcOnly	Oracle	HF	S+T	FE	DiAd-Conf	DiAD-Diss	DiAd-Radius	DiAd-Sample
HS	MED	79.03	82.33	73.86	79.93	79.77	**81.05**	80.40	80.46	80.84
	EDU	28.74	40.99	27.42	30.71	25.67	53.31	54.34	53.37	**55.03**
EDU	HS	83.12	85.64	44.94	**84.14**	83.16	78.41	78.13	80.88	79.45
	MED	73.19	82.15	42.44	73.26	**74.58**	67.75	72.23	69.25	68.75
MED	HS	81.98	86.89	76.87	82.53	79.64	**82.67**	82.40	82.62	82.42
	EDU	35.39	41.64	27.32	40.41	33.30	50.24	48.38	36.76	**50.37**
Average		63.57	69.94	48.81	65.16	62.69	68.91	69.31	67.22	**69.48**

ground-truth labeled data from source and surrogately labeled data from the target domain.

Frustatingly Easy (FE): This baseline is also a modification of another popular supervised DA method [6], where the feature space is augmented to enable the classifier to learn domain specific weight vectors. Like S+T, in our experiments, this approach trains the classifier using ground-truth labeled data from the source and surrogately labeled data from the target domain.

Oracle: Our final baseline represents a hypothetical scenario, where unlike the previous baselines, it has access to the ground-truth labeled data from the target domain and hence trains on labeled data from the target domain. The train and test sets are non-overlapping. While this too is not a DA approach, it helps us in getting an estimate of the performance of a classifier should labeled data from the target domain be available.

4.4 Evaluation

For each course, we used 80% of the data for training out of which 5% was used for development. While training the classifier, we also balanced the dataset by up-sampling the minority class. The test-set comprised of the remaining 20% of the data. For robustness, the reported results were averaged over 10 randomly held-out test sets.

From the perspective of helping students, the positive class is more important than the negative class. Thus, identifying all confusion posts so that they can be brought to the instructor's notice (high recall) would be desirable. Simultaneously, a high precision is also desirable so that the instructor's valuable time is not wasted in analyzing false-positives. Therefore, in line with previous studies [24], we evaluate the various approaches using the F-measure of the positive (confusion) class.

4.5 Quantitative Results

We emphasize that our goal in this study was not to understand the task of confusion detection, which in itself is a hard problem, nor do we address improving a solution to that problem. Instead, our goal is to address the problem of domain adaptation for this field. Our experiments were designed to help us answer the following questions:

Q1: Can we quantitatively demonstrate the need for DA in discussion forum analytics?

Q2: How does DiAd adapt the classifier to new target domains?

Q3: What is the role of confidence and dissimilarity in DiAd's performance?

Q4: How well does the state-of-the-art unsupervised approach to DA (HF) work on this problem?

Q5: DiAd iteratively uses surrogately labeled instances from the target domain to adapt to it. How does its performance compare to alternative ways (S+T or FE) of using surrogate labels for this task?

Q6: How does the adaptation process affect the performance on the source domain?

Q7: Overall, what are the most unexpected components of our findings?

Table 2 compares the performance of various methods on the confusion detection task. The first two columns represent the source and the target courses. The remaining columns represent the performance of the baselines and the DiAd variants. For example, the number in the 5^{th} column of the 1^{st} row indicates the performance of the HF model when it is adapted from HS to MED.

Q1: Need for domain adaptation

As mentioned in the introduction, the major challenge in utilizing supervised methods in this field is the lack of labeled data from new courses. A naive DA solution to this problem could be training a classifier on a previously labeled course and utilizing it in the new course. We first demonstrate the efficacy of this approach. The third and the fourth columns in Table 2 report the performance on the held-out test sets when the model is trained on labeled data from only the source (SrcOnly) and only the target (Oracle) respectively. We can see that in all cases, testing a classifier on an unseen course hurts its performance (at times very drastically) indicating an urgent need for DA. For example, the performance of a classifier trained on instances from the Medicine course (MED) is 82.33 when tested in the same domain (row 1, column 4) and drops to 79.03 (row 1, column 3) in the absence of labeled data from the testing domain (MED) but is instead trained on labeled data from another course, Humanities (HS).

Q2: Does DiAd help in adapting to new courses?

The answer to the previous question indicated a need for DA for the confusion detection task. We now investigate whether our proposed approach, DiAd and its variations, help us in alleviating this problem. Considering average performances (last row of Table 2) we can see that in general, the performance of DiAd-Sample (69.48

in last row and last column) is much better that the SrcOnly performance (63.57) (except when adapting from EDU to MED or HS and we discuss this while answering Q5.).

Comparing the performances of the DiAd variants–DiAd-Radius and DiAd-Sample–represented by the last two columns respectively, we see that the average performance of DiAd-Radius (67.22) [3] is inferior to that of DiAd-Sample (69.48). A possible explanation for this difference could be that DiAd-Sample has a better control over the number of surrogately labeled instances it adds to the training set in each iteration. Our analysis for answering Q5 below demonstrates that it is not sufficient to just add surrogately labeled instances from the target domain, but it is important to add them *slowly*. Since in DiAd-Radius the parameter is the radius of the circle and not the number of instances to be added to the training set, it loses control over how *slowly* it adds the surrogately labeled instances. In our experiments, not reported here due to space constraints, we saw that the surrogately labeled instances were indeed spread out very unevenly across various iterations in DiAd-Radius. So, even though the total number of iterations in DiAd-Radius and DiAd-Sample were comparable, DiAd-Radius added most of the instances in the first few iterations (making it more similar to S+T), and the later iterations only added a handful of instances. Thus, DiAd-Radius was not adapting the classifier *slowly* to the target domain while DiAd-Sample was. However, we acknowledge that on an average DiAd-Radius outperforms SrcOnly. It is also notable that the average performances of the two variations of DiAd, especially DiAd-Sample, are very close to that of the Oracle (69.94) which represents the hypothetical scenario when we have access to labeled data from the target domain. In particular, for individual course-pairs, we see that in all cases, the performance of DiAd-Sample is close to that of Oracle and sometimes even better. The improvement is likely because of the availability of additional data.

To summarize, comparing the performances of DiAd-Random and DiAd-Sample with SrcOnly and Oracle, we can conclude that they are indeed useful in adapting to unseen courses.

Q3: Role of confidence and dissimilarity

Both DiAd-Sample and DiAd-Radius choose instances to be appended to the training set based on two criteria: the prediction confidence of the current classifier, and the dissimilarity from the current train set. We seek to understand the contributions of these two criteria, individually, to the performance of DiAd. Toward this, we designed two other models that only use one of the constraints while selecting instances from the target–the confidence (DiAd-Conf) or the dissimilarity from the training set (DiAd-Diss). When we consider performances with individual course pairs, DiAd-Conf (column 8 of Table 2) outperforms DiAd-Sample (last column), when the source and target courses are HS and MED respectively. However, the improvements seem marginal.

On an average, we can see that the performance of both of these methods is worse than DiAd-Sample (last row of Table 2). This indicates that, in general, both constraints together constructively contribute towards the performance of DiAd.

Q4: Performance of state-of-the-art

The answer to question Q2 showed that DiAd is indeed helpful in adapting a classifier to an unseen course. However, it remains to be seen if there is a need for a new DA method for this task. Here we explore the performance of a recently proposed unsupervised DA method, HF, which is represented in Column 5 of Table 2. We note that HF is not particularly useful for this task and that DiAd significantly outperforms HF. This is especially remarkable when adapting from HS (or MED) to EDU when DiAd-Sample achieved F-scores of 55.03 and 50.37 respectively. On the other hand, HF could only achieve an F-score of 27.42 and 27.32 respectively. HF also particularly underperforms DiAd when adapting from EDU to MED or HS, confirming DiAd's utility.

Q5: Role of the iterative process

We now investigate the need to iteratively introduce surrogately labeled target instances to DiAd, as proposed in our approach. A simpler alternative would have been to present all these surrogately labeled instances to the model using a standard supervised DA approach such as S+T or FE [6]. Comparing columns 6 and 7 of Table 2, we can see that the performances of S+T and FE are very similar (except when adapting from HS or MED to EDU). On an average, looking at the last row of the table, we can see that S+T's performance (65.16) is better than SrcOnly (63.57). This, however, is not true for FE's average performance (62.69). This indicates that S+T, but perhaps not FE, is helpful in adapting a confusion-detection classifier to a new course.

More interestingly, comparing S+T's average performance (6^{th} column of the table) with that of several variations of DiAd's (last columns of the table), we see that DiAd mostly outperforms S+T (and also FE). [4]

For a different perspective of how the classifier performance changes as DiAd progresses, we plot the performance after every iteration for several course pairs in Figure 3. For example, MED $->$ EDU in the figure indicates that the source course was MED and the target course was EDU. The blue line (with square markers) indicates the performance of the classifier on the target domain as it adapts in every iteration of DiAd (DiAd-sample). The left-most point corresponds to the initial unadapted classifier which is, in principle, equivalent to SrcOnly. In almost all cases, we can see an initial performance spurt between iterations 0 and 1. This happens because during iteration 1 the model is exposed to (surrogately) labelled instances from the target course resulting in a sudden improvement in target performance. This reemphasizes the need for DA by introducing target instances even if they are surrogately labeled. We also see another bump in target performance around iteration 5. This might be because by this iteration, a significant number of target instances have been introdiced to the classifier

[3] While reporting the performance of DiAd-Radius, we experimented with several values of the user-provided parameter, r, and report results with $r = 0.7$.

[4] An exception to this occurs while adapting from EDU to HS or MED. For these two course pairs we can see that the performance of S+T or FE is better than SrcOnly indicating that the surrogately labeled instances were useful. However, their performance is also better than that of DiAd indicating that the iterative process did not help. This could possibly be explained by a more careful look at the data sizes which indicated that the size of EDU's training set was much larger than any other dataset (negative/positive instances in EDU: 6714/640, HS: 1358/2257, MED: 1581/1598). That is, when we iteratively add too few instances (as a fraction of the source training instances), they act more like outliers and degrade the model. The same instances when added together during S+T would form a sizable cluster and help the adaptation. To alleviate this issue, we recommend using a small labeled development set (5% of train data) in the target domain, and prefer S+T or FE over DiAd if the iterative process hurts performance on the development set.

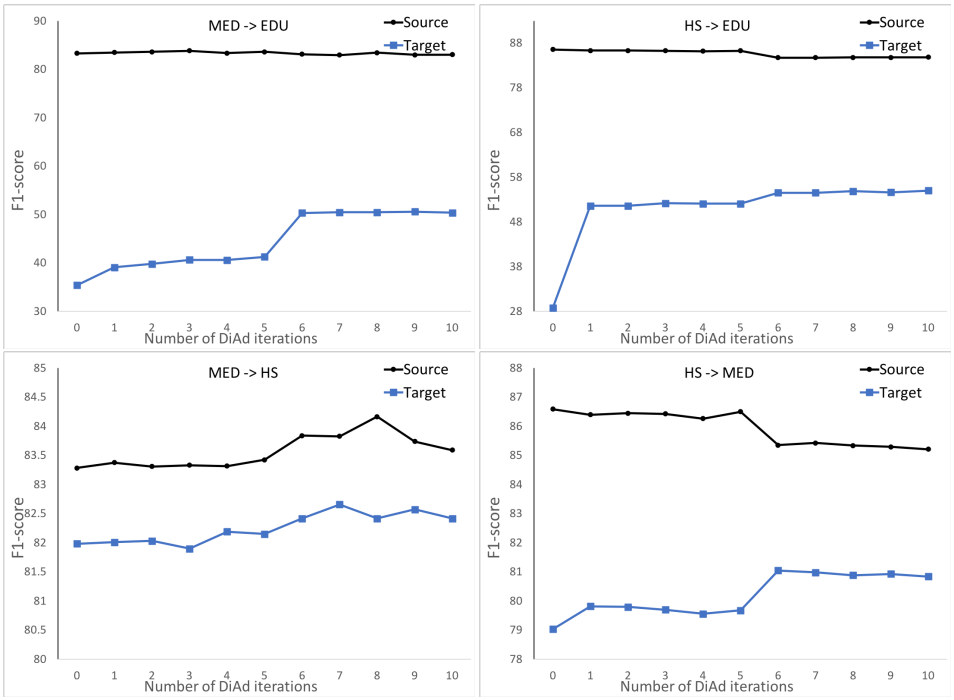

Figure 3: Changes in the Source and the Target performances with DiAd iterations. We can see that the Target performance improves while the Source performance almost remains the same (or degrades slightly) as DiAd adapts the classifier.

Table 3: Performance comparison of our approach, DiAd, with baselines for the confusion detection task on held-out testsets in the source domain. During the adaptation process, DiAd leads to a very slight drop in performance on the source side.

Source	Target		Baselines		DiAd-Variations			
		SrcOnly	S+T	FE	DiAd-Conf	DiAD-Diss	DiAd-Radius	DiAd-Sample
HS	MED	86.58	85.99	86.31	85.20	85.40	86.34	85.20
	EDU	86.58	86.37	86.70	84.42	85.02	85.96	84.83
EDU	HS	40.42	38.97	39.73	34.13	34.52	39.09	35.71
	MED	40.42	37.55	39.04	34.10	38.29	38.07	34.95
MED	HS	83.28	84.29	84.13	83.69	84.06	83.84	83.59
	EDU	83.28	84.32	83.82	82.89	83.31	84.42	82.99
Average		70.09	69.75	69.96	67.41	68.43	69.62	67.88

enabling it to perform reasonably well on the target domain. In general, we see that as the iterative process of DiAd continues, the performance on the target domain improves slowly.

Overall, the average performance of DiAd (69.48) over various course pairs is better than S+T's (65.16) and FE's (62.69) indicating that while it is advantageous to use surrogately labeled instances, it is imperative to slowly introduce the model to the new domain.

Q6: How does the adaptation process affect performance on the source domain?

The discussions so far highlighted the utility of DiAd in adapting a classifier to a new domain. However, we also want to ensure that such an adaptation does not result in a significant deterioration in the performance of the classifier on the source domain. This is especially important in the domain of online courses, where, while adapting a classifier to a new course, we would still like to use it for making predictions in (subsequent offerings of) the source course.

In order to answer this question, we compare performances of the adapted classifier on the held-out test sets from the source domain, in Table 3. For example, the 4^{th} column of the 1^{st} row indicates the performance (85.99) of S+T adapted from HS to MED on a held-out test set from the HS course. Broadly speaking, a model that does not adapt to the target domain is expected to have the best performance on the held-out test set from the source domain. Since SrcOnly, by design, does not adapt *at all* to the target domain, its performance remains the same for a given source irrespective of the target (e.g. see the first two rows of the third column). The more a classifier changes while adapting to the target domain, the more it deviates from an *optimal* classifier in the source domain and hence the lower its expected performance on the source test set, confirmed in the last row of Table 2. Here we can see that

Example post	SrcOnly	Oracle	DiAd-Sample
I'm a parent for kindergarten kid. Could you provide resources to formative assessment for kindergarten.	X	✓	✓
We do tracking, but only for one course. After that course, students can choose to take an AP Calc class or move to a Pre-Calc class. ... But if that benefits students more, should tracking be our choice?"	X	✓	X
How do we ignite this same creativity and level of comfort with older students who haven't had positive math experiences?	✓	X	✓

Table 4: Sample of *Confused* posts from EDU when adapting from HS.

the average performance of DiAd-Sample on the test set from the target course (69.48) was better than that of S+T (65.16), indicating that DiAd-Sample changed *more* during the adaptation process. Accordingly, in Table 3 we see that DiAd-Sample's performance on the source test set (67.88) is worse than that of S+T (69.75). Nevertheless, we see that the average performance of DiAd-Sample on the source test set is close to that of SrcOnly (70.09).

Figure 3 also shows the performance change of DiAd-Sample on the source test set as it adapts the classifier to the new domain. We can see that, in general, as the number of iterations increases, the performance on the source course either remains almost the same, or degrades very slightly. Surprisingly, we even see a slight improvement in the source performance when adapting from MED to HS. This could be attributed to the availability of more training data, even if not from the same course and is surrogately labeled.

Q7: Most unexpected aspects of our findings

The most interesting observation in our experiment was how some courses can be inherently difficult to predict on. For instance, if in-domain training data on the target is available (corresponding to the 'Oracle' column of Table 2), the F1 scores for HS and MED are reasonable on an absolute scale (> 80). However, prediction on EDU is very difficult (F1 score of about 40), which is the case even if target-domain training data is available. This problem becomes even more severe when target-domain training data is not available (F1 scores of 29 and 35). This indicates a need for future exploration of the cause of this poor performance and design of more robust algorithms, irrespective of the question of domain adaptation.

Another surprising result was the remarkable performance of DiAd when adapting from HS or MED to EDU. Our experiments revealed that the performance of DiAd-Sample for these pairs (55.03 and 50.37) was significantly better than performance of the Oracle on the same pairs (40.99 and 41.64). This is surprising because the Oracle has access to manually annotated target-domain training data, which DiAd does not have and only estimates surrogate labels on the target-domain. This superior performance cannot simply be attributed to the availability of more training data (manually annotated or surrogate). This is because HF, S+T and FE also have access to same data as DiAd (surrogate labels in target domain), but their performance is much worse than that of Oracle. This leads us to hypothesize that the rate and order in which DiAd iteratively adapts to the new domain may lead to fundamentally stronger models with the same amount of data.

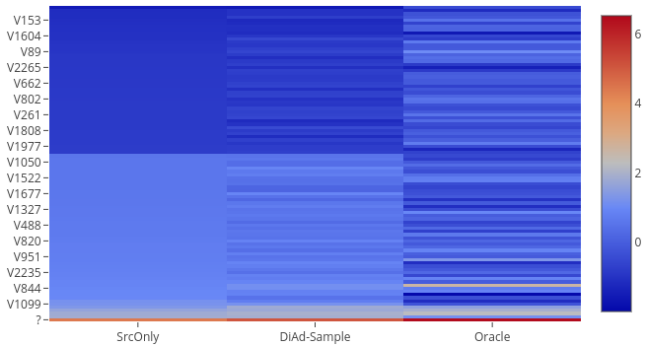

Figure 4: Visualization of feature-weights for different models. We can see that SrcOnly and Oracle are very different from each other while DiAd's profile (visually) intermediate between them

4.6 Error Analysis

Table 4 shows examples of confused posts that were predicted correctly/incorrectly by various models. Due to space constraints, we only show examples of posts when adaptating from HS to EDU. The first row is an example of successful adaptation, where the post was incorrectly predicted (predicted as not-confused) by SrcOnly but correctly predicted by Oracle and DiAd-Sample. This happened possibly because the post was about EDU-specific content (teaching kindergarten kids), and so SrcOnly, trained on HS instances, could not predict it correctly but when adapted to EDU, could yield correct prediction. The instance in the second row was correctly predicted by Oracle but incorrectly predicted by SrcOnly and DiAd-Sample. This could have happened because even though the post was target (EDU) specific, it was unusually long (... represents content we ommitted) which might have resulted in adapted signals getting lost or mixed with other signals. This is an example of insufficient adaptation. The last row represents an interesting case when after adaptation, DiAd could make correct prediction on the instance even though Oracle could not. This could have happened because the post was a short and well-formed question and it is likely that similar posts from the labeled corpus from the source course could have helped the model in correctly predicting this instance.

4.7 Model Visualization

The heatmap in Figure 4 demonstrates the (normalized) weights for the 100 most important features for adaptation from HS to MED. The Y-axis represents the features and X-axis represents the different models (one column each for the three models). Blue and red colors represent small and large weights respectively, and for ease of visualization the features are sorted according to their weights learned by SrcOnly. We see that the feature-weight profiles of SrcOnly (left column) and Oracle (right column) are most different from each other indicating a need for DA. The profile of DiAd (middle column), on the other hand, is (visually) intermediate between that of SrcOnly and Oracle. This indicates that DiAd steers the SrcOnly model to be more like the Oracle.

5 DISCUSSION AND CONCLUSION

The necessity for domain adaptation: DA is essential in predictive learning analytic models in online course contexts, considering the short life cycle of a typical course and the cost of manual annotations for a course. Addressing this problem of DA enables the application of classification models developed for one course to courses without labeled data. This is especially likely in at-scale learning environments, where new courses are constantly being added without access to timely labeled data. In this paper we have experimentally demonstrated the need for sophisticated DA methods for this field with confusion detection as a case.

Design justification for DiAd: We have shown the inadequacy of existing unsupervised DA methods based on transforming a classifier's features, which could be explained because most of the features are largely transferable. The difference possibly lies in adapting to the new conditional distribution of the label space given the inputs and we propose to tackle that using surrogately labeled instances from the target domain. Our proposed approach, DiAd, tackles this problem by iteratively and gradually introducing surrogately labeled instances from the target domain to the classifier, and slowly adapting it to the target course. Our experiments also demonstrated the utility of DiAd as compared to other alternatives.

Target audience and impact on teaching and learning activities: Our intent here was to draw the community's attention towards the dire need for DA in this field to enable practical applications. DA is amenable for use at an institutional- or learning-platform level for predictive/classification tasks that can be applied at-scale to more than one course. Additionally, online education technology designers and instructional personnel may mutually benefit from an exchange of course data and DA results for effective course design and enhanced productivity.

Future work could study DA for various problems in at-scale learning leveraging their commonalities and differences. While this study explored the use of DA for multiple courses, limitations in the dataset prevented us from exploring the use of DA for subsequent offerings of the same course, which we suggest as another direction for future exploration. To conclude, we addressed the problem of adapting a classifier to new courses for the confusion detection task. Our proposed approach, DiAd, tackles this by iteratively and gradually introducing surrogately labeled instances from the target domain to the classifier. We demonstrated that DiAd outperforms all baselines without considerably compromising on its performance in the source course. Also, its performance is very close to that of a hypothetical classifier which uses labeled target data.

6 ACKNOWLEDGEMENTS

This work was supported by the IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) – a research collaboration as part of the IBM AI Horizons Network.

REFERENCES

[1] Akshay Agrawal and Andreas Paepcke. 2014. The Stanford MOOC Posts Dataset. (2014). Available from http://datastage.stanford.edu/StanfordMoocPosts/.

[2] Akshay Agrawal, Jagadish Venkatraman, Shane Leonard, and Andreas Paepcke. 2015. YouEDU: Addressing Confusion in MOOC Discussion Forums by Recommending Instructional Video Clips. In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015*. 297–304.

[3] Himanshu Sharad Bhatt, Deepali Semwal, and Shourya Roy. 2015. An Iterative Similarity based Adaptation Technique for Cross-domain Text Classification. In *Proceedings of the 19th Conference on Computational Natural Language Learning, CoNLL 2015*. 52–61.

[4] John Blitzer, Ryan T. McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In *EMNLP 2007, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. 120–128.

[5] Snigdha Chaturvedi, Dan Goldwasser, and Hal Daumé III. 2014. Predicting Instructor's Intervention in MOOC forums. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1501–1511.

[6] Hal Daumé III. 2007. Frustratingly Easy Domain Adaptation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

[7] Dan Davis, Ioana Jivet, René F. Kizilcec, Guanliang Chen, Claudia Hauff, and Geert-Jan Houben. 2017. Follow the Successful Crowd: Raising MOOC Completion Rates Through Social Comparison at Scale. In *Proceedings of the 7th International Learning Analytics & Knowledge Conference (LAK '17)*. 454–463.

[8] Jacqueline L. Feild, Nicholas Lewkow, Sean Burns, and Karen Gebhardt. 2018. A generalized classifier to identify online learning tool disengagement at scale. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge, LAK 2018*. 61–70.

[9] Srećko Joksimović, Dragan Gašević, Vitomir Kovanović, Bernhard E Riecke, and Marek Hatala. 2015. Social presence in online discussions as a process predictor of academic performance. *Journal of Computer Assisted Learning* 31, 6 (2015), 638–654.

[10] René F. Kizilcec and Sherif Halawa. 2015. Attrition and Achievement Gaps in Online Learning. In *Proceedings of the Second ACM Conference on Learning @ Scale, L@S 2015*. 57–66.

[11] Theodore J Kopcha, Michael Orey, Wendy Dustman, and others. 2015. Exploring college students' online help-seeking behavior in a flipped classroom with a web-based help-seeking tool. *Australasian Journal of Educational Technology* 31, 5 (2015).

[12] Vitomir Kovanović, Srećko Joksimović, Zak Waters, Dragan Gašević, Kirsty Kitto, Marek Hatala, and George Siemens. 2016. Towards automated content analysis of discussion transcripts: A cognitive presence case. In *Proceedings of the sixth international conference on learning analytics & knowledge*. 15–24.

[13] Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the Web. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005*. 342–351.

[14] Quan Nguyen, Michal Huptych, and Bart Rienties. 2018. Linking Students' Timing of Engagement to Learning Design and Academic Performance. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. 141–150.

[15] Oleksandra Poquet, Nia Dowell, Christopher Brooks, and Shane Dawson. 2018. Are MOOC Forums Changing?. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge (LAK '18)*. 340–349.

[16] Oleksandra Poquet, Vitomir Kovanović, Pieter de Vries, Thieme Hennis, Srećko Joksimović, Dragan Gašević, and Shane Dawson. 2018. Social presence in massive open online courses. *International Review of Research in Open and Distributed Learning* 19, 3 (2018).

[17] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé III, and Lise Getoor. 2014. Learning Latent Engagement Patterns of Students in Online Courses. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. 1272–1278.

[18] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé III, and Lise Getoor. 2014. Understanding MOOC Discussion Forums using Seeded LDA. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2014*. 28–33.

[19] Xu Wang, Diyi Yang, Miaomiao Wen, Kenneth R. Koedinger, and Carolyn Penstein Rosé. 2015. Investigating How Student's Cognitive Behavior in MOOC Discussion Forum Affect Learning Gains. In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015*. 226–233.

[20] Miaomiao Wen, Diyi Yang, and Carolyn Penstein Rosé. 2014. Sentiment Analysis in MOOC Discussion Forums: What does it tell us?. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014*. 130–137.

[21] Diyi Yang, Mario Piergallini, Iris K. Howley, and Carolyn Penstein Rosé. 2014. Forum Thread Recommendation for Massive Open Online Courses. In *Proceedings of the International Conference on Educational Data Mining, EDM 2014*. 257–260.

[22] Jae-Bong Yoo and Jihie Kim. 2014. Capturing Difficulty Expressions in Student Online Q&A Discussions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. 208–214.

[23] Jianfei Yu and Jing Jiang. 2015. A Hassle-Free Unsupervised Domain Adaptation Method Using Instance Similarity Features. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, Volume 2: Short Papers*. 168–173.

[24] Ziheng Zeng, Snigdha Chaturvedi, and Suma Bhat. 2017. Learner Affect Through the Looking Glass: Characterization and Detection of Confusion in Online Courses. In *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017*.