Towards Domain General Detection of Transactive Knowledge Building Behavior

James Fiacco
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA
jfiacco@cs.cmu.edu

Carolyn Rosé
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA
cprose@cs.cmu.edu

ABSTRACT

Support of discussion based learning at scale benefits from automated analysis of discussion for enabling effective assignment of students to project teams, for triggering dynamic support of group learning processes, and for assessment of those learning processes. A major limitation of much past work in machine learning applied to automated analysis of discussion is the failure of the models to generalize to data outside of the parameters of the context in which the training data was collected. This limitation means that a separate training effort must be undertaken for each domain in which the models will be used. This paper focuses on a specific construct of discussion based learning referred to as Transactivity and provides a novel machine learning approach with performance that exceeds state-of-the-art performance within the same domain in which it was trained and a new domain, and does not suffer any reduction in performance when transferring to the new domain. These results stand as an advance over past work on automated detection of Transactivity and increase the value of trained models for supporting group learning at scale. Implications for practice in at-scale learning environments are discussed.

ACM Classification Keywords

I.2.7. Artificial Intelligence: Natural Language Processing

Author Keywords

Multitask learning; transactivity; limited dataset size; transfer learning; neural network; attention model; deep learning; entailment; natural language inference.

INTRODUCTION

Over the past decade, increasing interest in automated analysis of online discussion for learning, sometimes referred to as Discourse Analytics, has been featured in research on learning in at scale environments like Massive Open Online Courses (MOOCs). In particular, prior work in MOOCs has demonstrated that students can benefit from discussion encounters with other students [18]. Much of this prior work has targeted short synchronous collaborative discussion assignments or informal and unstructured discussion in asynchronous discussion forums [36]. More recently the topic of supporting team based project learning in MOOCs has emerged [53].

Prior work in Computer-Supported Collaborative Learning has demonstrated the value in automated analysis of discussion for enabling effective assignment of students to project teams [53], for triggering dynamic support of group learning [29], and for assessment of learning processes [32, 14]. Though a plethora of frameworks for analysis of discussion for learning are in operation, many include a dimension for collaborative knowledge construction where a valued conversational behavior is one where students explicitly make their reasoning visible in a way that connects back to ideas and reasoning expressed earlier in the encounter [23]. One popular and long standing such construct is that of Transactivity.

The concept of Transactivity originally grows out of a Piagetian theory of learning where this conversational behavior is said to reflect a balance of perceived power within an interaction [4, 15]. It is a property of discourse in an educational context that is associated with interactions that are beneficial for learning [2], and thus it has been of great interest within the learning sciences in the area of discussion based learning.

Transactive contributions demonstrate consideration of the earlier expressed ideas. Thus, it makes sense that recent work has demonstrated that automated models for Transactivity detection can be used as a foundation for highly effective assignment of students to project teams in MOOCs by estimating the collaborative potential of pairs of participants based on the exchange of Transactive contributions [53]. Even before this recent work, there was much interest in automated detection of transactivity in educational applications [25, 44, 32, 1, 22]. However, where there are reported successes, past work has failed to produce models that generalize well to new domains [34], which we address in this work.

A Transactive contribution to a discourse must meet two requirements[22]. First it must display reasoning, in other words revealing how a speaker thinks something works, which can be accomplished through an expressed evaluation, comparison, or reference to a causal mechanism. For example, "Use of coal increases pollution" displays a causal mechanism and "Use of wind power may not be reliable throughout the year" expresses

L@S 2018, June 26–28, 2018, London, United Kingdom

 ISBN 978-1-4503-5886-6/18/06…$15.00
DOI: https://doi.org/10.1145/3231644.3231655

an evaluation. But something like "I prefer coal power" does not express reasoning. A Piagetian perspective on learning would suggest that students display their reasoning more when they are in a safe environment where they feel their ideas are valued and respected [15, 2].

The second requirement for a Transactive contribution is that it references and idea expressed earlier in a discourse. Students reference the ideas of another student when they are listening to that student. It is a sign that the student takes the other student seriously enough to consider their ideas and how their respective ideas relate to one another [15, 2]. If earlier a speaker said, "Wind is my choice because it is sustainable", a Transactive reply would be "Wind is sustainable, but it fails to be reliable throughout the year". On the other hand, "Use of coal is cheap and reliable" would not be a Transactive reply. In one case, the speaker shows consideration of another student's ideas, while in the second case we do not see this consideration. From our technical perspective, an important aspect of the operationalization that we leverage in the work reported in this paper is the *idea relatedness* of the Transactive contribution and the earlier contribution it refers transactively to.

In the remainder of the paper we review prior work that lays a technical foundation for our machine learning approach to automated Transactivity detection. Next we describe our novel approach. We then present an evaluation that demonstrates that the novel approach beats a state-of-the-art baseline both within the domain in which it was trained and a separate domain, without any drop in performance when moving to the separate domain. We discuss implications for practice in at-scale learning environments. We conclude with limitations and directions for continued research.

RELATED WORK

This paper presents a technical approach in the Neural Network modeling paradigm, which has experienced a recent resurgence of interest. Within that sphere, this work draws from additional related bodies of work. Thus, we begin with a general discussion of how our work is situated within the general sphere of Neural Network modeling, specifically in recent work in Deep Learning. Next, we describe prior work in computational modeling of Textual Entailment, as our approach is heavily influenced by Deep Learning work done in that area. As we will explain, the concept of Entailment shares the important foundation in *idea relatedness* introduced above in the operationalization of Transactivity. Finally, we transition into a discussion of Transfer Learning, as our method falls in the general class of models centered around domain transfer of machine learning.

Neural Network Modeling

Neural network modeling was the central form of machine learning in the 1980s and early 1990s [46] but gave way to Probabilistic Graphical Models beginning at the end of the 1990s [28]. It fell out of favor in the interim but then experienced a resurgence of interest under the name of Deep Learning in recent years [47]. Specifically within the area of computational modeling of discourse, deep learning methods have considerably advanced the state-of-the-art in the past few years. Most notably, it has seen strong results in social chatbots [30], speech act classification [26], and work on understanding conversational processes [17]. However, for many researchers working on understanding specific discussion processes, deep learning approaches can remain impractical due to the inordinate amount of data it typically requires to be effective [55], often tens or even hundreds of thousands of instances. Typical annotated corpora in research on discussion based learning even in at scale learning environments like MOOCs are at least an order of magnitude smaller if not two or three orders of magnitude smaller. In this paper we have, therefore, set out to explore options for drastically reducing the amount of training data required as well as to increase the domain generality of trained models in order to reduce the preparatory effort required to employ machine learned models in at scale learning environments. The largest Transactivity coded data set used in our experiments had fewer than 500 instances.

Transfer learning in the field of machine learning is a method for applying learning from a source task to some target task that may differ in surface features but share structure at a deep level [38]. The motivation, in general, is to leverage the knowledge from some task that is more fundamental for use in a more challenging, often more specialized setting. The advantage is that one might expect the more fundamental task to have broad applicability as a preparatory task for many more specific target tasks. In that way, the effort to prepare a large training corpus for the more fundamental task pays off tremendously with each new target task where only a small effort to prepare training data will then be required. We strive to apply this concept to Transactivity detection, as one of many detection problems in the field of computational discourse analysis where there is a dirth of public datasets large enough to make effective use of many published deep learning approaches. This stands in contrast to some more well studied natural language tasks that have already benefited from deep learning, such as parsing [11], sentiment analysis [20], and textual entailment [6].

Entailment

In our work, we use the Entailment task as the more fundamental task that forms a foundation for Transactivity detection. The Entailment task, specifically, comprises of deciding whether the concepts presented in one text can be determined to be true given some context or premise given in a different text [13]. For example, if an object is a shoe, then we can assume it was made to be worn on the foot. Therefore, *shoe* entails *made to be worn on the foot*. Because the task requires inferring abstract connections between ideas within two snippets of text, we considered it a good candidate for transferring learning to more specific applied discourse tasks where it is important to identify forms of *idea relatedness*, such as Transactivity.

One text entails another text if there is a conceptual link via an inference that associates those two texts. Similarly, a Transactive contribution to a discussion is one that displays reasoning and uses that reasoning display to evaluate, extend, transform,

or refer substantively to an assertion made earlier in the discourse. The simple way of thinking about what constitutes a reasoning display is that it has to communicate an expression of some causal mechanism or express an evaluation or comparison. Transactive contributions are reasoning displays where the contribution either explicitly refers linguistically in some way to a prior statement, such as through the use of a pronoun or deictic expression, or implicitly by referring to a related idea. Thus, both Transactivity detection and entailment detection share the notion of concepts linked via inference.

What makes detection of Transactivity challenging in a domain general way is identification of the relevant conceptual links between ideas related by inference. Instead, using state-of-the-art approaches to Transactivity detection, such as linear Support Vector Machine models with n-gram features [44] is that rather than learn the general task of identifying idea relatedness, the models tend to learn which concepts in the training domain are related to one another, and to identify them from their associated words. Thus, the learned associations are not useful anymore in a different domains since the set of related concepts that are relevant in the new domain will be different. Our work is based on the premise that networks trained to perform the Entailment task may need to learn internal text encoding representations that enable measurement of "closeness by inference" rather than "closeness in meaning", in other words identification of abstract connections between expressed ideas. Since Transactive contributions build on or evaluate assertions made earlier in a discourse, the sub-problem of detecting idea relatedness is a foundational task. Note that the concept of *idea relatedness* used here as in the operationalization of Transactivity goes beyond text similarity. The idea is not that the two concepts are rephrases of one another, but that they are related to one another through some inference.

Transfer Learning

What we have alluded to above is that we employ a transfer learning paradigm the builds upon the Entailment task in order to train a Transactivity detector with a relatively small training set.

Transfer learning, the process of transitioning learning from one task to another, has long been studied in the context of reinforcement learning and robotics [49], but has more recently began have strong influences in other domains [38]. In natural language processing, transfer learning has been shown to support a variety of basic tasks including chunking, named entity recognition, and semantic role labeling [12, 41, 43]. More recently, deep learning models in the paradigm of sequence-to-sequence modeling have been shown to be able to leverage multi-task learning [31, 56]. However, many of these multi-task transfers have been less challenging that what we attack in this paper since both the initial task and the transfer task made use of very large datasets. Here we approach a transfer task in two domains where there is not a very large corpus in either domain.

The most comparable work to ours is by Mou et al. [33] where they examined several methods of multi-task learning on entailment and paraphrase detection. They showed that the success of transfer learning between NLP tasks appears to be correlated with the structural relatedness of the pairs of tasks with one another. Our contribution builds upon this concept by applying transfer learning to real world data, selecting and structuring the model to properly leverage structural relatedness of the chosen tasks.

In our work, we employ a transfer method that builds on training for the Entailment task as pretraining task. In particular, for the entailment pretraining we specifically consider the simple attention model proposed in the Language Technologies community [40]. It is notable because despite the presented model's simple structure and relatively small number of parameters, it performs comparably with far more complex models that have orders of magnitude more parameters. As we are looking to work with small datasets, models that are both simple and effective are the natural choice.

In recent years, large datasets for the textual entailment task have been developed and made available for researchers [6]. State of the art performance on these datasets have been rising steadily with use of complex recurrent neural networks [48], neural attention models [40], tree based neural models [35], and hybrid methods using both of those approaches [51, 8]. Models trained on such a corpus to identify concepts linked through inference across a plethora of domains are required through the training process to build conceptual representations for words that make identification of conceptual links possible. The idea behind our computational approach is to leverage this tendency in a pretraining step for training to detect Transactivity in one topic domain so that rather than learn just the associations between specific pairs of concepts, the model would learn to leverage the entailment representation space that enables computation of idea relatedness of texts across domains. The hope is that a model trained to detect transactivity in one domain but building on this general purpose representation space would be able to transfer to another domain where the relevant set of linked concepts is different but still within the broad range of topics covered inside the very broad and diverse entailment corpus.

MODELS

Decomposable Attention Model

For our modeling work, we adapt the previously published Decomposable Attention Model presented by [40] for the purpose of transfer learning from the Entailment task to Transactivity detection. This model was chosen as a starting point for this work by virtue of it demonstrating benefits including using an order of magnitude fewer trainable parameters than other common methods in for approaching textual entailment while maintaining a high level of performance. Furthermore, the model was not bound by a task specific architecture or feature set that makes it a good candidate for multi-task learning with pairwise comparisons. The Decomposable Attention Model operates in four stages: input, attention, comparison, and aggregation. We will provide an overview of their model to provide context for our adaptations and an intuitive explanation for the benefit of the reader. See Figure 1 for a visualization of the structure of the model.

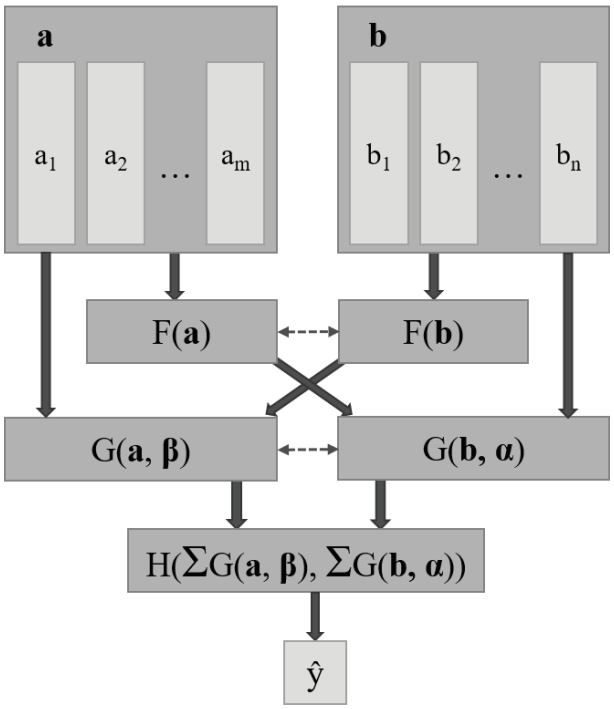

Figure 1. Decomposable Attention Model. Arrows with dotted lines indicate networks with shared weights.

Input: The model is defined with input of two text segments, $\mathbf{a} = (a_1, ..., a_m)$ and $\mathbf{b} = (b_1, ..., b_n)$ where m and n are the lengths of the respective segments. Each vector a_i and b_j are real value, d dimensional vector embeddings for each word in their respective text segments. For words not in the vocabulary, an embedding is assigned randomly based on the word's shape. The output of the model is defined as $y = (y_1, ...y_C)$ where C is the number of output classes for the dataset.

Attend: At the first stage of the model, each input is passed into network F where a soft alignment between word embeddings is computed via a type of neural attention [3]. The attention mechanism weights the importance of each word in each sentence for how it will be used in the subsequent computations. The network, F is a simple feed forward neural network with rectified linear activation [19]. This results in a matrix of dimension $m \times n$, $e_{ij} = F(a_i, b_j)$, where each cell contains a score of how important each given word in a text segment is, given that it co-occurs with another word in the other text segment.

The matrix is then normalized for each direction to obtain two vectors, α and β, to represent the aligned subphrases from **b** to **a** and **a** to **b**:

$$\begin{aligned} \beta_i &= \sum_{j=1}^{n} \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{ik})} b_j, \\ \alpha_j &= \sum_{i=1}^{m} \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{kj})} a_i. \end{aligned} \tag{1}$$

Compare: In the next stage, each aligned phrase is compared separately by an additional feed forward neural network, G:

$$\begin{aligned} v_{1,i} &= G([a_i, \beta_i]) \quad \forall i \in [1, ..., m], \\ v_{2,j} &= G([b_j, \alpha_j]) \quad \forall j \in [1, ..., n]. \end{aligned} \tag{2}$$

There are now two sets of vectors that encode a comparison between the input and the aligned subphrases of each input text segment.

Aggregate: The final stage of the model compresses the two sets of vectors via summation, giving a vector representation for each text segment with respect to the other.

$$\mathbf{v}_1 = \sum_{i=1}^{m} v_{1,i} \quad , \quad \mathbf{v}_2 = \sum_{j=1}^{n} v_{2,j}. \tag{3}$$

These two vector representations are then concatenated and fed into a final feedforward neural network with softmax activation, H, to predict probabilities of class values: $\hat{\mathbf{y}} = H([\mathbf{v}_1, \mathbf{v}_2])$. The predicted class is thus $\hat{y} = \arg\max_i \hat{\mathbf{y}}_i$

Transferable Attention Model

We refer to our adaptation of the original Decomposable Attention Model as the Transferable Attention Model. Specifically, we adapted the model described above for the purpose of transfer learning. We started by separating the model into two modules. The fist module includes both the attention and comparison components, which generate sentence representations from the input representations referred to in deep learning work as word embeddings. The second module includes the classification step, which takes in the two text segment comparison vectors and makes a prediction for the text pair's class.

The reason we needed to separate these components of the model is that, while performing transfer learning, we need to be able to dynamically manipulate the weights or structures of the classification stage while maintaining the integrity of the parameters learned in the representation stage. This allowed us the flexibility to have varying numbers of classes between our source task and our target task. The modularity also allows for varying types of classifiers or bindings to other models that we consider for future work.

METHOD

Throughout the experimental work reported in this paper, we used five datasets to demonstrate first task transfer and then domain generalizability. Short descriptions of each are provided here. We will refer to two main tasks: the Entailment task, which is our source task, and Transactivity Detection, which is our target task. We also refer to two domains in which we perform the Transactivity task. The source domain, which is a Power Plan domain, is where the training for the Transactivity task is performed. And the target domain, which is the Superheroes domain, is the domain for the Transactivity task where we do the test of domain generality of the trained Transactivity task model.

Data

Stanford Natural Language Inference Corpus: As our primary dataset for the Entailment task, we selected the Stanford Natural Language Inference Corpus (SNLI), version 1.0 [6]. This corpus contains over 570 thousand annotated text pairs for the recognizing textual entailment task. Pairs consist of a premise and a hypothesis, and are labeled as *entailment* if the hypothesis is definitely true given the premise, *contradiction* if the hypothesis is definitely false given the premise, and *neutral* if the hypothesis could be true, but is not guaranteed to be given the premise. The premises were captions from the Flickr30k corpus [57] and the hypotheses were generated via an Amazon Mechanical Turk task where workers were asked to write three alternate captions that followed certain rules to create appropriate hypotheses for the entailment task.

Multi-Genre NLI Corpus: The Multi-genre Natural Language Inference Corpus (MultiNLI), version 0.9 [54] consists of over 390 thousand text pairs annotated in the same way as the SNLI corpus described above. However, this dataset includes text segments from four different categories: fiction, government texts, magazine articles about popular culture, and transcripts of telephone speech. As this is a comparable dataset to SNLI, we determined that it was a valid alternative for pretraining in our experiments.

Power Plant Transactivity Corpus: Our larger annotated Transactivity dataset, which is a shared dataset we used as a target task to transfer our Entailment model, comprises 426 annotated text segments [53]. These text segments come in the form of posts made by participants from Amazon's Mechanical Turk working in teams where they needed to determine which type of power source(s) a city should make use of given a set of characteristics that the city possesses. For each instance, the labeled post is in reply to a previous post which is also included in the representation of the instance for reference. Each instance was annotated as *Transactive* or *not Transactive* with respect to the context.

Superhero MOOC Transactivity Corpus: This set of annotated Transactivity data consists of 57 annotated text segments from a Massive Open Online Course in which students design superheroes and discuss them with other members of the course [52]. The data are collected conversations between students. Each contribution was annotated as *Transactive* or *not Transactive* with respect to the conversation. Each instance was annotated as *Transactive* or *not Transactive*, just like in the previous corpus.

Microsoft Research Paraphrase Corpus: The Microsoft Research Paraphrase corpus [16], has 5801 annotated sentence pairs that are either labeled as *paraphrase* or *not a paraphrase*. This will be used in one of our validation experiments.

Training

Training the Transferable Attention Model is performed in three stages: first training the model on the source task for a given number of iterations, then dynamically changing the classification module to match the target task, and finally training the new model on the target task until convergence. During the training of the target task, error is propagated backwards through both modules of the model to allow for fine tuning of the attention and comparison networks for the Transactivity task. Input word embeddings are held fixed throughout the training. This backpropagation method is a standard training approach for neural network models.

Implementation Details

We implemented the Transferable Attention Model using the Keras deep learning library [10] with the Theano tensor library [50] as a foundation. Each network, F, G, and H were 2 layer feed forward densely connected networks with 200 hidden units per layer. The structure of H was the same for both target and source task with the exception that the output dimension of the target task was 2 while in the source task the output dimension was 3. Text segments were fixed at 100 tokens with zero vectors left padding the text segments if the length was shorter and truncating if the length was longer. Word embeddings were 300 dimension pretrained GloVe [42] embeddings.

Our model was trained with the Adam optimizer [27] with a learning rate of 0.001. Training was done on a NVIDIA GeForce GTX 760 with CUDA 8.0 [37] and CuDNN 5 [9]. One iteration of pretraining on the source task was performed, classification weights were reset with a random Gaussian distribution, and ten iterations of training on the target task were performed per fold during the experiments. Metrics for the tenth iteration of target dataset training were reported in all cases.

EXPERIMENTS

Beyond demonstrating the performance of our model on the given task, we also motivated our experiments with validating that our model operated as our intuitions predicted.

The metrics that we collect throughout our experiments are accuracy, to see the percent correct of the predictions each model makes, and Cohen's kappa, to evaluate the models' accuracy in a way that controls for agreement by chance. Results are reported in the Results and Discussions section and in each of the corresponding subsections.

Cross Domain Generality

In order to evaluate our method on the task of Transactivity detection we test our method of transfer learning against several baselines, which are described below.

After pretraining the model on the SNLI corpus, we perform a standard ten-fold cross validation over our Transactivity training corpus, in each fold beginning with the model weights generated by the pretraining. After each fold, we evaluate the trained model on the held out Transactivity data from the source domain (i.e., the Power Plan data). We also apply the model trained in each fold to the data in the target domain (i.e., the Superheroes data). As is a standard practice for evaluation by cross-validation, results for all the folds are averaged together for our final metric, reported in the Cross Domain Generality subsection.

Baselines

Logistic Regression with Unigrams: Previous work by Joshi et al. [25] on predicting transactivity used a simple unigram model [39] with logistic regression, trained on the Power Plant Transactivity Corpus. We therefore use this as a baseline to connect our method with previous work in that field.

Basic Neural Network: As our model consists of only feed forward neural networks, we evaluated the performance of a basic neural net architecture without an attention mechanism using the same GloVe word embeddings. We use a 2 layer feed forward neural network with 200 hidden units per layer, as that is the equivalent structure for the classification step of our model. We allowed this model to be pretrained as with our Transferable Attention Model.

Bidirectional Long Short-Term Memory: Many systems in entailment use LSTM [24], and the bidirectional variant, BLSTM [21] based models with word embeddings [7]. We also evaluated our model with sentence embeddings generated by single layer BLSTMs with 128 hidden units each direction, then classified with a densely connected layer. This model was also pretrained on the SNLI corpus.

Lexical Overlap

In early work with textual entailment, it was shown that simple word overlap is a strong predictor of entailment [5]. Because of the similarity between entailment and transactivity, we hypothesized that this may hold for our task as well so we investigated to ensure our model was making inferences beyond that naïve method. To eliminate this possibility, we removed all overlapping words between target and context sentences for both the entailment dataset and the transactivity datasets during test and training. We then report the results of our model, trained and evaluated as in our first experiment above.

This makes the task considerably more difficult as the model loses access to a large amount of content based context. It therefore must rely on non-overlapping structural information in the texts, synonyms, or more abstractly connected words.

Dataset Alignment

In the SNLI dataset, there are three classes, *entailment*, *contradiction*, and *neutral*, one of which can be applied to each text pair. However, in the transactivity dataset, each pair can only be identified by either *transactive* or *not transactive*. When arranging the data between pretraining on entailment and training on transactivity data, we need to decide how these classes map to one another to give the pretraining the most impact. *Entailment* and *neutral* are easy to correspond to *transactive* and *not transactive* respectively given that the former indicates a logical connection between the two while the latter indicates there is not. *Contradiction*, on the other hand, is more difficult to determine. The hypothesis can either be considered connected to the premise through logic that makes the hypothesis impossible or it can be considered not connected as it is not entailment.

We tested applying the contradiction component of the pretraining data differently to evaluate which performed the best for the transfer learning. The conditions that we evaluated were relabeling the contradiction cases as either entailment (*contradiction positive*) or neutral (*contradiction negative*) before evaluating as in the Cross Domain Generality experiment. We also pretrained with all three entailment classes and just ignored the contradiction label while training and evaluating on transactivity. Discussion of these results can be found below in the Results section.

Ablation

This set of experiments was designed to make sure that the transfer learning was having sufficient impact to warrant their inclusion in the model. We first tested to ensure that the pretraining was being utilized by the model and not simply being overwritten by the training that the model performs over the transactivity dataset. To accomplish this, we executed the experiment as in the Cross Domain Generality case without pretraining the model on the entailment dataset. We then evaluated on only the in domain data.

To ensure that the model was not simply applying textual entailment to our transactivity dataset and that it learned something meaningful from the small dataset, we ran the experiment with only weights learned on the entailment task and evaluating on the in domain transactivity test data. Both of these experiments are reported below in the Results section.

Alternative Datasets

The last set of experiments were motivated by the possibility that the entailment task was not necessarily the explanation for the performance of the model. We considered two alternative explanations: that the SNLI corpus may be particularly suited for transfer leaning in this domain, or that any sentence comparison task would transfer sufficiently for transactivity to be predicted.

To evaluate the first consideration, we tested the model using an alternate source dataset, the MultiNLI corpus. In this evaluation, our source task was the same as before, but the data used to pretrain was different.

To evaluate the second consideration, we evaluated our model when the source task was changed to the similar, though not identical, task of paraphrase detection using the MSRP corpus. Key differences between paraphrase detection and entailment is that entailment represents a directed relationship between text pairs, while paraphrase detection is undirected. Paraphrase detection also has only two output classes compared to entailment's three.

One issue that we needed to control for when pretraining with paraphrase detection was that the dataset was significantly smaller than either entailment corpus. To provide a fair comparison, we randomly selected an equivalent number of SNLI and MultiNLI examples to pretrain with and reported those results as well.

Data Set Size

One of the most frequent questions asked about automated approaches to discussion analysis that require training is how much data is required. Thus we include one additional experiment that manipulates the amount of training data and shows how performance varies as a result.

	Accuracy		Cohen's Kappa	
Models	*In domain*	*Out of domain*	*In domain*	*Out of domain*
Unigrams with LR	0.795	0.667	0.510	0.376
Basic Neural Network	0.798	0.721	0.498	0.305
Bidirectional LSTM	0.814	0.782	0.543	0.472
Transferable Attention (TA)	**0.840**	**0.832**	**0.607**	**0.611**

Table 1. Model performance in domain versus out of domain compared to baselines.

	Accuracy		Cohen's Kappa	
Models	*In domain*	*Out of domain*	*In domain*	*Out of domain*
Unigrams with LR	0.781	0.667	0.476	0.363
Basic Neural Network	0.761	0.733	0.412	0.309
Bidirectional LSTM	0.812	0.772	**0.524**	0.442
Transferable Attention (TA)	**0.828**	**0.810**	0.475	**0.551**

Table 2. Model performance in domain versus out of domain compared to baselines with no lexical overlap between target and context.

RESULTS

Models	Accuracy	Kappa
TA	**0.840**	**0.607**
- pretraining	0.700	0.035
- transactivity training	0.307	0.005

Table 3. Model performance with varying training stages removed.

Cross Domain Generality

Table 1 shows the results for our comparison of our model's performance on the in-domain transactivity dataset to the out of domain transactivity data set after pretraining on the SNLI corpus for the entailment task. We find that our model outperforms the baselines in all metrics, with over 80% accuracy and a kappa of over 0.6 indicating good agreement with annotators. When comparing accuracy between tests, we can see that our model loses less than one percentage point, while the unigram baseline drops over 12 percentage points when evaluating on the out of domain set. The simple word embedding based baselines also appeared to drop across domains, though not as dramatically as the unigram model.

From this, we can infer that learning to operate over general semantic vectors can influence the domain generality of classification models. We also demonstrate that transferring learned representations from a deep model trained on a general source task can improve performance on multiple domains of a target task even if the model was only trained on a single domain of the target task.

Lexical Overlap

A similar story is seen in Table 2 with lexical overlap between target and context text segments is removed. All of the tested models dropped performance modestly, though our model still managed to get an accuracy of over 80%. This provides compelling results that the reasoning our modeling is doing between the two text segments is more abstract that simply measuring word overlap.

Dataset Alignment

Because the source task is a three class classification and the target task is a two class classification, we considered alternative alignments between categories, which we found to have different implications for performance in the two transactivity datasets. The results presented in Table 4 make sense when the data is examined qualitatively.

In the condition in which contradiction was used as a positive example, the model obtained a notably higher kappa on the in domain dataset that contained more competitive transacts, demonstrating disagreement. However, when contradictions were treated as negative examples, the model performed much better on the out of domain dataset which contains a lower percentage of competitive transacts. When contradiction is given a separate class during source task training and not used in target task training, the kappa is higher for both target task datasets indicating that the model was free to make a determination on the role of learned contradiction-type relationships as it applies to the transactivity task.

Ablation

Table 3 reveals that the pretraining on the source task and the training on the target task are both critical for the performance of the model. This indicates that the model learned important representation structure from the large amount of data provided with the source task. It also can be seen to not only classify the target task as if it were the source task, but rather it learned about the difference between the tasks sufficiently to adapt to the new task.

Alternative Datasets

In our final set of experiments as reported in Table 5, we can see that there are comparable results between using SNLI and MultiNLI for pretraining. An interesting observation is that pretraining on the MultiNLI corpus seemed to perform better for in domain transactivity detection while pretraining on the SNLI corpus had stronger results for out of domain prediction. This raises some interesting questions regarding how the domain of the source data sets can influence the generalizability of target datasets while transferring learning.

We can also see that with a smaller number of source task text pairs, it appears that SNLI provides the best performance, followed by MultiNLI, then MSRP performs the worst. This provides some evidence that the entailment task is providing more valuable pretraining as compared to paraphrase task.

	Accuracy		Cohen's Kappa	
Models	*In domain*	*Out of domain*	*In domain*	*Out of domain*
TA with contradiction negative	**0.848**	0.824	0.542	0.586
TA with contradiction positive	0.828	0.791	0.598	0.511
TA with three classes	0.840	**0.832**	**0.607**	**0.611**

Table 4. Model performance with respect to how contradiction was treated in task transfer.

	Accuracy		Cohen's Kappa	
Models	*In domain*	*Out of domain*	*In domain*	*Out of domain*
TA with full SNLI training set	0.840	**0.832**	0.607	**0.611**
TA with full MultiNLI training set	**0.869**	0.804	**0.647**	0.544
TA with both SNLI and MultiNLI	0.833	0.828	0.536	0.585
TA with truncated SNLI	**0.781**	**0.786**	**0.328**	**0.464**
TA with truncated MultiNLI	0.764	0.761	0.255	0.383
TA with MSRP training set	0.752	0.751	0.210	0.345

Table 5. Model performance with respect to dataset used for pretraining.

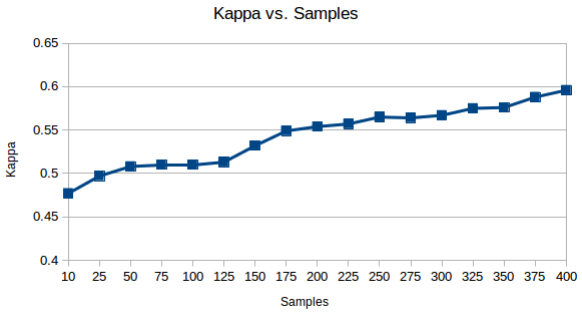

Figure 2. Graph of the change in kappa score over varying number of transactivity training instances.

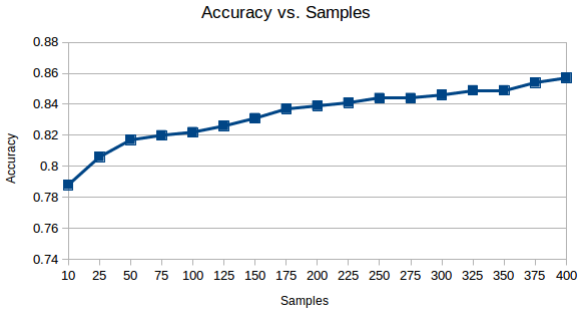

Figure 3. Graph of the change in accuracy score over varying number of transactivity training instances.

Dataset Size

Here we address the question of how much data is required for training in order to achieve the best performance. We ran a series of cross-validation experiments using the full Transfer Attention Model where we manipulated the number of training instances sampled from the maximal training set on each fold of the cross-validation. The results are displayed in Figures 2 and 3 for Kappa and Accuracy respectively. Here we see progressive improvement as more and more data is used, without a substantial plateau. Thus, it is possible even better performance could have been achieved had we provided more data, and a smaller training set size would have yielded poorer performance.

DISCUSSION AND IMPLICATIONS

The results presented in this paper demonstrate that the novel neural approach to classification we present achieves an improvement in accuracy as well as generalizability over previously published work on automated Transactivity detection [25, 44, 1, 34, 22].

Automated Transactivity detection has a variety of applications in online learning environments, especially where discussion is part of the learning process. The presence of Transactivity is a significant predictor that a collaborative discussion is conducive to learning [4, 2, 25, 22, 45]. That makes Transactivity a construct that is particularly valuable to be able to detect.

In general, automated detection of discussion processes that are either positively or negatively related to learning can be applied to problems such as automated assignment of students to project teams [53], for triggering dynamic support of group learning processes [29, 18], and for assessment of those learning processes [45]. Raising the level of accuracy at this detection increases the feasibility of offering these forms of automated support in massive online learning environments.

The generalizability result has particular implications for learning at scale. Scale is not just about reaching a large number of students in one course or offering the same course many times, but being able to apply a form of learning support broadly across courses. Without the ability to generalize a model's performance to new data sources, it would be necessary to train a new Transactivity detection model for every course, or maybe even every assignment where the model will be used. Clearly, a solution that requires retraining over and over is more costly to use than one that can be trained once and then reused many times in many different contexts.

CONCLUSIONS AND FUTURE WORK

We have demonstrated a method to utilize a general inference task with a large corpus of annotated data to learn representations that can be used as pretraining for a small discourse task with strong domain generality. We have also validated our

approach to control for alternate explanations of the performance of the model that would indicate that it is not learning a sufficiently abstract representation of the data.

We have also began to explore the use of other source tasks for the transfer learning, though have found thus far that for the purposes of Transactivity detection, using the entailment task as a source appears to have the best results likely due to the structural similarity between the tasks.

Though the results presented in this paper are promising, one limitation is that the domain transfer was only tested on one transfer domain (i.e., transfer from the Power Plan domain to the Superheroes domain) and one transfer task (i.e., Transactivity detection). In future, we will test this paradigm on a wider variety of domains and tasks.

The promising results presented in this work inspire a number of future research directions. For example, examining the feasibility of injecting domain specific information into the model during training to improve the ability of the model to adapt to complex target domains is a potentially interesting direction of study. Lastly, incorporating this model or representations learned by this model as a component of a larger system may be of interest for specific applications.

ACKNOWLEDGMENTS

This research was funded in part by NSF grants ACI-1443068 and IIS 1546393 and funding from the Schmidt Foundation.

REFERENCES

1. Hua Ai, Marietta Sionti, Yi-Chia Wang, and Carolyn Penstein Rosé. 2010. Finding transactive contributions in whole group classroom discussions. In *Proceedings of the 9th International Conference of the Learning Sciences-Volume 1*. International Society of the Learning Sciences, 976–983.
2. Margarita Azmitia and Ryan Montgomery. 1993. Friendship, transactive dialogues, and the development of scientific reasoning. *Social development* 2, 3 (1993), 202–221.
3. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
4. Marvin W Berkowitz and John C Gibbs. 1983. Measuring the developmental features of moral discussion. *Merrill-Palmer Quarterly (1982-)* (1983), 399–410.
5. Johan Bos and Katja Markert. 2005. Recognising Textual Entailment with Logical Inference. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 628–635. DOI: http://dx.doi.org/10.3115/1220575.1220654
6. Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).
7. Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021* (2016).
8. Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038* (2016).
9. Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. 2014. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759* (2014).
10. François Chollet. 2015. Keras. https://github.com/fchollet/keras. (2015).
11. Ronan Collobert. 2011. Deep Learning for Efficient Discriminative Parsing.. In *AISTATS*, Vol. 15. 224–232.
12. Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.
13. Cleo Condoravdi, Dick Crouch, Valeria De Paiva, Reinhard Stolle, and Daniel G Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning-Volume 9*. Association for Computational Linguistics, 38–45.
14. Mihai Dascalu, Stefan Trausan-Matu, Danielle S McNamara, and Philippe Dessus. 2015. ReaderBench: Automated evaluation of collaboration based on cohesion and dialogism. *International Journal of Computer-Supported Collaborative Learning* 10, 4 (2015), 395–423.
15. Richard De Lisi and Susan L Golbeck. 1999. Implications of Piagetian theory for peer learning. (1999).
16. William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proc. of IWP*.
17. Nia M Dowell, Arthur C Graesser, and Zhiqiang Cai. 2016. Language and discourse analysis with Coh-Metrix: Applications from educational material to learning environments at scale. *Journal of Learning Analytics* 3, 3 (2016), 72–95.
18. Oliver Ferschke, Diyi Yang, Gaurav Tomar, and Carolyn Penstein Rosé. 2015. Positive impact of collaborative chat participation in an edX MOOC. In *International Conference on Artificial Intelligence in Education*. Springer, 115–124.
19. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011a. Deep Sparse Rectifier Neural Networks.. In *Aistats*, Vol. 15. 275.

20. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011b. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 513–520.
21. Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5 (2005), 602–610.
22. Gahgene Gweon, Mahaveer Jain, John McDonough, Bhiksha Raj, and Carolyn P Rosé. 2013. Measuring prevalence of other-oriented transactive contributions using an automated measure of speech style accommodation. *International Journal of Computer-Supported Collaborative Learning* 8, 2 (2013), 245–265.
23. Cindy E Hmelo-Silver. 2013. *The international handbook of collaborative learning*. Routledge.
24. Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
25. Mahesh Joshi and Carolyn Penstein Rosé. 2007. Using transactivity in conversation for summarization of educational dialogue.. In *SLaTE*. 53–56.
26. Hamed Khanpour, Nishitha Guntakandla, and Rodney Nielsen. 2016. Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, 2012–2021. `http://aclweb.org/anthology/C16-1189`
27. Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
28. Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.
29. Rohit Kumar, Carolyn Penstein Rosé, Yi-Chia Wang, Mahesh Joshi, and Allen Robinson. 2007. Tutorial dialogue as adaptive collaborative learning support. *Frontiers in artificial intelligence and applications* 158 (2007), 383.
30. Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541* (2016).
31. Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* (2015).
32. Bruce M McLaren, Oliver Scheuer, Maarten De Laat, Rakheli Hever, Reuma De Groot, and Carolyn Penstein Rosé. 2007. Using machine learning techniques to analyze and support mediation of student e-discussions. *Frontiers in Artificial Intelligence and Applications* 158 (2007), 331.
33. Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How Transferable are Neural Networks in NLP Applications? *arXiv preprint arXiv:1603.06111* (2016).
34. Jin Mu, Karsten Stegmann, Elijah Mayfield, Carolyn Rosé, and Frank Fischer. 2012. The ACODEA framework: Developing segmentation and classification schemes for fully automatic analysis of online discussions. *International journal of computer-supported collaborative learning* 7, 2 (2012), 285–305.
35. Tsendsuren Munkhdalai and Hong Yu. 2016. Neural Tree Indexers for Text Understanding. *arXiv preprint arXiv:1607.04492* (2016).
36. Matti Nelimarkka and Arto Vihavainen. 2015. Alumni & tenured participants in MOOCs: Analysis of two years of MOOC discussion channel activity. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*. ACM, 85–93.
37. John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. 2008. Scalable parallel programming with CUDA. *Queue* 6, 2 (2008), 40–53.
38. Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.
39. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 79–86.
40. Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. *CoRR* abs/1606.01933 (2016). `http://arxiv.org/abs/1606.01933`
41. Hao Peng, Sam Thomson, and Noah A Smith. 2017. Deep Multitask Learning for Semantic Dependency Parsing. *arXiv preprint arXiv:1704.06855* (2017).
42. Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–1543.
43. Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108* (2017).
44. Carolyn Rosé, Yi-Chia Wang, Yue Cui, Jaime Arguello, Karsten Stegmann, Armin Weinberger, and Frank Fischer. 2008. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International journal of computer-supported collaborative learning* 3, 3 (2008), 237–271.

45. Carolyn Penstein Rosé, Iris Howley, Miaomiao Wen, Diyi Yang, and Oliver Ferschke. 2017. Assessment of Discussion in Learning Contexts. In *Innovative Assessment of Collaboration*. Springer, 81–94.

46. David E Rumelhart, James L McClelland, PDP Research Group, and others. 1987. *Parallel distributed processing*. Vol. 1. MIT press Cambridge, MA, USA:.

47. Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

48. Lei Sha, Baobao Chang, Zhifang Sui, and Sujian Li. 2016. Reading and Thinking: Re-read LSTM Unit for Textual Entailment Recognition. (2016).

49. Matthew E Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10, Jul (2009), 1633–1685.

50. Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688 (May 2016). `http://arxiv.org/abs/1605.02688`

51. Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. *arXiv preprint arXiv:1702.03814* (2017).

52. Miaomiao Wen. 2016. *Investigating Virtual Teams in Massive Open Online Courses: Deliberation-based Virtual Team Formation, Discussion Mining and Support*. Ph.D. Dissertation. Carnegie Mellon University.

53. Miaomiao Wen, Keith Maki, Xu Wang, Steven Dow, James D. Herbsleb, and Carolyn Penstein Rosé. in press. Transactivity as a Predictor of Future Collaborative Knowledge Integration in Team-Based Learning in Online Courses. In *Proceedings of the 21st ACM Conference on Computer-Supported Cooperative Work and Social Computing*.

54. Adina Williams, Nikita Nangia, and Samuel R Bowman. 2016. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. (2016). `https://www.nyu.edu/projects/bowman/multinli/paper.pdf`

55. Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. 2016. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

56. Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345* (2017).

57. Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2 (2014), 67–78.