Lightly-Supervised Modeling of Argument Persuasiveness

Isaac Persing and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{persingq,vince}@hlt.utdallas.edu

Abstract

We propose the first lightly-supervised approach to scoring an argument's persuasiveness. Key to our approach is the novel hypothesis that lightly-supervised persuasiveness scoring is possible by explicitly modeling the major errors that negatively impact persuasiveness. In an evaluation on a new annotated corpus of online debate arguments, our approach rivals its fully-supervised counterparts in performance by four scoring metrics when using only 10% of the available training instances.

1 Introduction

Argumentation mining is a relatively new and active area of research in the natural language processing (NLP) community, focusing on extracting argument components (e.g., claims, premises) and determining the relationships (e.g., support, attack) between them. Recently, researchers have begun work on modeling an intriguing linguistic phenomenon, the *persuasiveness* of arguments.

In this paper, we examine argument persuasiveness in the context of an under-investigated task in argument mining, *argument persuasiveness scoring*. Given a text consisting of an argument written for a particular topic, the goal of argument persuasiveness scoring is to assign a score to the text that indicates how persuasive the argument is. An argument persuasiveness scoring system can be used in a variety of situations. In an online debate, for instance, an author's primary goal is to convince others of the argument expressed in her comment(s). Similarly, in persuasive essay writing, an author should establish convincing arguments. In both situations, a persuasiveness scoring system could provide useful feedback to these authors on how persuasive their arguments are.

Being a *discourse-level* task, argument persuasiveness scoring is potentially more challenging than many NLP tasks. Oftentimes, argument persuasiveness can only be determined by understanding the discourse, not by the presence or absence of lexical cues. As an example, consider the debate argument shown in Table 1, which is composed of the author's assertion and her justification of the assertion written in response to a debate motion. It is fairly easy for a human to determine that this argument should be assigned a low persuasiveness score because the argument could be more clear. However, the same is not true for a machine, primarily because it is not possible to determine the persuasiveness of this argument merely by considering the words or phrases appearing in it.

Given the difficulty of the task, it is conceivable that *unsupervised* argument persuasiveness scoring is very challenging. Nevertheless, a solution to unsupervised argument persuasiveness scoring is of practical significance. This is because of the high cost associated with manually creating persuasiveness-annotated data needed to train classifiers in a supervised manner. This contrasts with tasks such as polarity classification and stance classification. In these tasks, large amounts of annotated data can be harvested from the Web, as it is typical for a user to explicitly indicate her polarity/stance while writing her comments in a discussion/debate forum.

We propose a *lightly-supervised* approach to argument persuasiveness scoring. To our knowledge, this is the first lightly-supervised approach to the task: virtually all previous work involving argument persuasiveness has adopted supervised approaches, training models with a large number of surface features that encode lexico-syntactic information. Note that learning from a large number of lexico-syntactic features is difficult, if not im-

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 594–604,
Taipei, Taiwan, November 27 – December 1, 2017

Motion	This House would ban teachers from interacting with students via social networking websites.
Assertion	Acting as a warning signal for children at risk.
Justification	It is very difficult for a child to realize that he is being groomed; they are unlikely to know the risk. After all, a teacher is regarded as a trusted adult. But, if the child is aware that private electronic contact between teachers and students is prohibited by law, the child will immediately know the teacher is doing something he is not supposed to if he initiates private electronic contact. This will therefore act as an effective warning sign to the child and might prompt the child to tell a parent or another adult about what is going on.

Table 1: The motion, assertion, and justification text of a debate argument.

possible, when annotated data is scarce. Hence, we explore a different idea, addressing lightly-supervised argument persuasiveness scoring via an *error-modeling* approach. Specifically, guided by theoretical work on persuasiveness, we begin by defining a set of *errors* that could negatively impact an argument's persuasiveness. The key step, then, is to model an argument's errors: given an argument, we predict the presence and severity of the errors it possesses in an unsupervised manner by bootstrapping from a set of heuristically labeled seeds. Finally, we learn a persuasiveness predictor for each error-labeled argument from a small amount of persuasiveness-annotated data.

Our contributions are two-fold. First, we propose the first lightly-supervised approach to persuasiveness scoring that rivals its supervised counterparts in performance on a new dataset consisting of 1,208 online debate arguments. Second, we make our annotated dataset publicly available.[1] Given the difficulty of obtaining annotated data for this task, we believe that our dataset will be a valuable resource to the NLP community.

2 Related Work

There have been several recent attempts to address tasks related to argument persuasiveness. Habernal and Gurevych (2016a,b) *rank* a pair of arguments w.r.t. persuasiveness, but ranking alone cannot tell us *how* persuasive an argument is. Persing and Ng (2015) score a student essay based on whether it makes a (un)convincing argument for its thesis. Using the conversations in the ChangeMyView subreddit, Tan et al. (2016) study factors affecting whether a challenger can successfully persuade a commenter to change the view she expressed in her original post.

While Wei et al. (2016) also predict the persuasiveness of debate posts, their work differs from ours in several aspects. First, many of their debate posts are written in response to a preceding comment in the conversation. Hence, it is not uncommon to see emotional rather than logical arguments or even insults and personal attacks. In addition, it may not always be possible to understand what the argument is and why the author made a particular argument without understanding the (preceding) context. In contrast, the debate comments in our corpus are written in response to a given debate topic. In other words, each comment is written independently of the other comments and therefore can be understood without them.

In a broader sense, our error-modeling approach is related to work on holistically scoring an essay via detecting and totaling up specific errors in it. For details, we refer the reader to Shermis et al. (2010) and Leacock et al. (2014).

3 Corpus and Annotation

We use as our corpus a randomly selected subset of 165 debates obtained from the International Debate Education Association (IDEA) website[2]. These debates cover a wide range of topics including politics, economics, religion, and science. Each debate consists of a *Motion*, which expresses a stance on the debate's topic, and an average of 7.3 arguments, each of which either agrees or disagrees with the motion's stance. Each of the 1,208 arguments consists of an *Assertion*, which expresses in one sentence why the author agrees or disagrees with the motion, and a *Justification*, which explains in an average of 6.9 sentences why the author believes her assertion.

We ask two native speakers of English to annotate each of the 1,208 arguments with a persuasiveness score after familiarizing them with the (topic- and domain-independent) scoring rubric (see Table 2). Specifically, we ask our annotators to score each argument's persuasiveness on a scale of $1-6$. The example argument in Table 1 gets a persuasiveness score of 2 because it could be expressed more clearly.

[1] See http://www.hlt.utdallas.edu/~persingq/Debate/ for a complete list of our annotations.

[2] http://idebate.org/

Score	Description of Argument Persuasiveness
6	A **very persuasive**, **clear** argument. It would persuade most previously uncommitted readers and is devoid of problems that might detract from its persuasiveness or make it difficult to understand.
5	A **persuasive**, or only **pretty clear** argument. It would persuade most previously uncommitted readers, but may contain some minor problems that detract from its persuasiveness or understandability.
4	A **decent**, or only **fairly clear** argument. It could persuade some previously uncommitted readers, but problems detract from its persuasiveness or understandability.
3	A **poor**, or only **mostly understandable** argument. It might persuade readers who are already inclined to agree with it, but contains severe problems that detract from its persuasiveness or understandability.
2	A **very unpersuasive** or **very unclear** argument. It is unclear what the author is trying to argue or the argument is just so riddled with problems as to be completely unpersuasive.
1	The author **does not make an argument** or it is **unclear what the argument is**. It could not persuade any readers because there is nothing to be persuaded of.

Table 2: Descriptions of argument persuasiveness scores.

	1	2	3	4	5	6
AP	3	12	20	21	20	24

Table 3: Distribution of error/argument persuasiveness scores as percentages.

Table 3 shows the distribution of scores for argument persuasiveness. To measure inter-annotator agreement, we select a subset of 69 arguments and ask both annotators to score them w.r.t. argument persuasiveness. The average difference between the annotator-assigned scores is 0.899. For the sake of our experiments, when annotators disagree on a score, we average their annotations together, rounding up to the nearest whole number to obtain the gold score.

4 Error Types

Key to our approach to persuasiveness scoring is the unsupervised modeling of the errors that could negatively impact persuasiveness. In this section, we define five such error classes, which are motivated by theoretical work on persuasiveness.[3]

Grammar Error (GE) Connor and Lauer (1985) note that grammar and/or mechanical errors can interrupt the flow of discourse in persuasive essays, so we give arguments a GE score of 1 if they contain GEs severe enough to make the argument hard to understand, and 0 otherwise. The argument in Table 1 gets a GE score of 0 because it contains no severe GEs.

Lack of Objectivity (LO) Oktavia et al. (2014) consider the use of personal opinions as evidence in argumentative writing a fallacy, so we give arguments a LO score of 1 if they display an inappropriate lack of objectivity, and 0 otherwise.[4] The argument in Table 1 receives a LO score of 1 because the author weaves a scenario in which she repeatedly speculates on what a child thinks or will do.

Inadequate Support (IS) Petty and Cacioppo (1984) find that arguments with more support are more persuasive, so we give arguments an IS score of 0 if they offer adequate support to justify their assertion, 1 if they do not offer enough support, or 2 if they offer almost no support. The example argument gets an IS score of 2 because the author's scenario is completely unsupported.

Unclear Assertion (UA) In Connor's (1990) criteria for judging assertions in persuasive writing, the lowest score is assigned to essays which did not clearly assert the problem they address. So we give an argument an UA score of 1 if it is not clear how the assertion is related to the motion without reading the justification, or 2 if the assertion is incomprehensible without reading the justification. It receives a score of 0 otherwise. The example argument gets a UA score of 1 because it is not clear how the assertion is related to the motion.

Unclear Justification (UJ) Because a smooth flow of ideas throughout an argument is important to its persuasiveness, Connor (1990) also evaluates persuasive essays' coherence. Since it is not clear what an incoherent argument is arguing for, we give an argument an UJ score of 2 if the justification appears unrelated to the assertion, 1 if it does not concisely justify the assertion, or 0 if the justification is clear. The example argument gets

[3] We also annotated the 1,208 arguments in our corpus with these five errors even though they were not used in the experiments in this paper. See Persing and Ng (2017) for details on the error annotations.

[4] Note, however, that other forums may try to craft emotional debates on purpose for their effectiveness. For instance, Lukin et al. (2017) show that emotional arguments can indeed be very persuasive and that they resonate with different audiences due to audience/reader preset biases and their own personality traits.

an UJ score of 0 as it is easy to understand the author's point in the justification.

5 Approach

In this section, we present our approach to persuasiveness scoring. Broadly, it first predicts the *presence* and *severity* of the aforementioned errors (Section 5.1), then uses these predictions to assign persuasiveness scores (Section 5.2).

5.1 Prediction of Error Types

Our process for predicting error types consists of two steps. First, for each error type, we heuristically apply error severity values to a set of training arguments that can be confidently error-labeled (Section 5.1.1). Using these error-labeled arguments as seeds, we then apply the expectation maximization (EM) algorithm (Dempster et al., 1977) to predict the error severity values of the remaining training arguments (Section 5.1.2).

5.1.1 Heuristics

In this subsection, we describe our heuristics.

Grammar Error (GE) To detect GEs, we use the LanguageTool proofreading program[5] to detect all GEs (e.g., redundant phrases and typos) in all training set justifications. We then calculate the frequency with which GEs occur per sentence in each justification, clustering these values using k-means clustering[6]. Finally, we label the training set arguments in the highest cluster with a GEs value of 1, and training set arguments in the lowest cluster with a GEs value of 0. This makes intuitive sense because GEs can hinder persuasiveness if they occur very frequently, and cannot hinder persuasiveness if they never occur.

Lack of Objectivity (LO) We count how frequently the word "morally" appears in justifications per token. We employ k-means clustering on these frequencies to help us identify which justifications use it most. The justifications falling in the highest cluster's arguments are heuristically labeled with a LO severity of 1. We do the same with the word "certain". Finally, if an author uses less than five definite articles in her justification, we heuristically label her argument with a severity of 1. These rules make sense because arguments that are too concerned with the author's morality or in which the author seems too certain, or in which the author is rarely specific are likely to display a LO.[7]

To find arguments not displaying a LO (severity $= 0$), we count and k-means cluster the frequency of first person plural pronouns in the justifications. Arguments whose justifications are in the lowest cluster are labeled with a LO severity of 0. This makes sense because justifications that lack objectivity often rely on stories about the writer's personal experiences. We use plural pronouns to capture this rather than singular ones because thesis statements (which are not inherently subjective) often begin with "I believe" or "I think".[8]

Inadequate Support (IS) To assign IS severities, we first need to know how many sources an argument cites.[9] An argument that cites no references is assigned an IS severity of 2. If the argument cites only one reference, it gets a score of 1. Finally, we cluster arguments by the number of sources they cite. Arguments in the highest cluster are assigned an IS severity of 0. These rules make sense because arguments that cite a lot of sources are probably adequately supported.

Unclear Assertion (UA) UAs typically consist of very short sentence fragments (e.g. "Europe"). For this reason, we heuristically assign an argument an unclear assertion severity of 2 if they are less than four words long.

To identify arguments with an UA severity of 1, we first identify all content lemmas (nouns, pronouns, verbs, adjectives, adverbs) in the assertion. If none of these lemmas are mentioned in the justification, the argument gets a severity of 1. Since this heuristic necessarily conflicts with the previous one, when applying UA heuristics, rules with greater severity take precedence.

Finally, we k-means cluster the counts of assertion content lemmas appearing in the justification and assertion lengths.[10] If an argument is not in the lowest cluster in either of these, it gets labeled

[5] https://languagetool.org/

[6] Unless otherwise noted, $k = 4$ in k-means clustering.

[7] We note that these lexical features are potentially specific to this particular domain. There have been a number of works examining objectivity and subjectivity that go beyond lexical features and use syntactic structures (Riloff and Wiebe, 2003; Wilson et al., 2005) and emotional and factual arguments (Oraby et al., 2015).

[8] Since more than one heuristic might apply to a given argument, we leave an argument unlabeled if the heuristics tell us to apply inconsistent labels to it. This is also how we handle contradictory heuristics for the remaining errors.

[9] We develop heuristics for extracting references from the justification. See the Appendix for these heuristics.

[10] We use $k = 6$ for assertion length clustering because assertions vary greatly in length.

1	# of grammar errors per sentence in justification (GE)
2	# of times the word "morally" appears in justification (LO)
3	# of times the word "certain" appears in justification (LO)
4	# of definite articles in justification (LO)
5	# of first person plural pronouns in justification (LO)
6	# of references cited in justification (IS)
7	# of words in assertion (UA)
8	# of content lemmas in assertion that also appear in justification (UA)
9	# of sentences in justification (UJ)
10	# of times words that lemmatically match the assertion's subject appear in the first argument of a contingency-cause discourse relation in justification (UJ)

Table 4: Features used by the generative model. The error type for which each feature is originally developed is shown in parentheses.

with an UA severity of 0.

Unclear Justification (UJ) As with UAs, UJs are often very short. For this reason, we k-means cluster the sentence counts in our training set justifications, and label arguments whose justifications fall into the lowest cluster with an UJ severity of 2. As in the previous error, this rule takes precedence over other rules.

To identify arguments with UJ severities of 1 or 0, we first dependency and discourse parse our assertions and justifications using Stanford CoreNLP (Manning et al., 2014) and Lin et al.'s (2014) PDTB-style discourse parser, respectively. Using the dependency parse, we identify the assertion's main subject, which we assume is the first word that is a child in an nsubj or nsubjpass relationship. Next, we count the number of times words that lemmatically match the subject appear in the first argument of a contingency-cause discourse relation in the justification. Finally, we k-means cluster these counts, assigning arguments with justifications in the highest cluster an UJ severity of 0, and those in the lowest cluster a severity of 1. These rules make sense because a justification that discusses its assertion's topic's effects frequently is likely to be very topically coherent, thus having a clear justification.

5.1.2 Bootstrapping using EM

Recall that for each error type t, our heuristics only label a subset of the training arguments with error severity values for t.[11] To label the remaining training arguments, we apply EM to bootstrap from the heuristically labeled seeds for t.

Specifically, we initialize the model parameters using only the seeds for t. After that, we iterate the E-step and the M-step until convergence. In the E-step, we probabilistically (re)label each unlabeled training argument with its error severity value for t using the current model parameter values. Then, in the M-step, we re-estimate the model parameters using both the seeds and the training arguments probabilistically (re)labeled in the E-step.

To understand what the model parameters are, we need to specify the generative model. In our experiments, we employ Naive Bayes as the underlying generative model, effectively assuming that each feature value is conditionally independent of other feature values given the class value (which in this case is the severity value for t).

To fully specify the model parameters, we need to specify the features used to represent each argument. Specifically, we employ the 10 features used in the heuristics described in the previous subsection. For the sake of clarity, we list them again in Table 4. Note that all of them have numerical values. Hence, to reduce data sparseness, we k-means cluster the values, and use the 10 k-valued features in the EM-based bootstrapping process.

Regardless of which error type we train the model for, the same set of 10 k-valued features will be used. In other words, the generative models for the five error classes differ only w.r.t. the set of seeds used to initialize the model parameters. After learning, we employ the model learned for each error type to error-label the test arguments.

5.2 Persuasiveness Prediction

Like many other unsupervised and weakly-supervised models, we make a modeling assumption in our approach in order to facilitate learning in an environment where annotated data is scarce. Specifically, we assume that the persuasiveness score of an argument inversely correlates with the sum of its severity scores over all errors.[12] This assumption intuitively makes sense: as the number and severity of the errors increase, the corresponding argument becomes less persuasive.

Given this assumption, we train a lightly supervised persuasiveness predictor as follows. First, we cluster the training arguments by the sum of

[11] The heuristics for GE, LO, IS, UA, and UJ can label 39%, 66.%, 86%, 50%, and 87% of the training arguments, respectively.

[12] For instance, if an argument has an IS severity of 2 and a LO severity of 1, the sum of its severity scores will be 3.

severity scores over all errors.[13] Then, we randomly select n arguments from each cluster c (where $1 \leq n \leq 12$ in most of our experiments), and manually label them with their persuasiveness scores. Finally, we assign to each c a persuasiveness score that is the average of the persuasiveness scores of the n manually labeled arguments in c.

During testing, we compute the sum of severity scores over all errors for each test argument, assign it to the corresponding cluster, and predict its persuasiveness score as the score of the cluster it is assigned to. Since our system assigns the Average persuasiveness of training arguments having the same Error Severity count, we call it ASE.

6 Evaluation

In this section, we evaluate our approach to persuasiveness prediction. Since there is an element of randomness in our algorithm and the baselines (in which arguments get labeled), we report results using 5 repetitions of 5-fold cross validation.

6.1 Scoring Metrics

We employ four evaluation metrics for persuasiveness scoring, namely E, ME, MSE, and PC.

The simplest metric, E, measures the frequency at which a system predicts the wrong score. ME and MSE measure the mean error and mean squared error of our persuasiveness predictions, respectively. The formulas below illustrate how we calculate E, ME, and MSE, respectively:

$$\frac{1}{N}\sum_{A_j \neq E'_j} 1, \quad \frac{1}{N}\sum_{j=1}^{N} |A_j - E_j|, \quad \frac{1}{N}\sum_{j=1}^{N} (A_j - E_j)^2$$

where A_j, E_j, and E'_j are the annotator assigned, system predicted, and rounded system predicted persuasiveness scores[14] respectively for argument j, and N is the number of arguments.

The last metric, PC, computes Pearson's correlation coefficient between a system's predicted scores and the annotator-assigned scores. A positive (negative) PC implies that the two sets of predictions are positively (negatively) correlated.

Note that E, ME, and MSE are *error* metrics, so lower scores on them imply better performance. In contrast, PC is a *correlation* metric, so higher correlation implies better performance.

6.2 Baseline Systems

We employ six baseline systems. All baselines are support vector regressors (Drucker et al., 1997) trained using LibSVM (Chang and Lin, 2001) with default parameters, differing only in terms of the features used by the learner.

Bag of words (BOW) In the first baseline, we use as features the bag of words extracted from the argument's assertion and justification.

Word n-grams (WNG) The second baseline uses word n-grams (n=1,2,3) extracted from the argument's assertion and justification as features.

Bag of part-of-speech tags (BOPOS) Our third baseline employs as features the bag of POS tags in the argument's assertion and justification.

Style Our fourth baseline captures aspects of an argument's style. Specifically, it employs four types of features that are motivated by Tan et al.'s (2016) Style baseline, namely:

Length-based features: As longer arguments can be more detailed, we encode as a feature the length in tokens and sentences of an argument's assertion and justification.

Word category-based features: For each of the following categories of words/tokens, we employ as features the absolute count and frequency per token in an argument's justification: (1) definite and indefinite articles and first and second person pronouns, both of which we learned in Section 5.1 can be useful for detecting lack of objectivity; (2) question marks and quotations, which indicate how an argument is structured; (3) positive and negative sentiment words as determined by Mohammad and Yang (2011) since excessive emotion can also signal a lack of objectivity; (4) URLs, since these may be another way of citing evidence; (5) hedge words[15], which can be used to express argument uncertainty; and (6) phrases that indicate the author is giving an example ("e.g.", "for instance", "for example").

[13] While the highest possible error severity count is 8, there is no argument in our corpus for which we predict that count. Hence, we only end up with 8 clusters, one for each error severity count (0−7).

[14] Since a regressor assigns each argument a real value rather than an actual valid score, it would be difficult to obtain a reasonable E score without rounding the system estimated score to one of the possible values. For that reason, we round the estimated score to the nearest valid persuasiveness score (1−6 at one-point increments) when calculating E. For other scoring metrics, we round the predictions to 1.0 or 6.0 if they fall outside the 1.0−6.0 range.

[15] The hedge words are taken from http://english-language-skills.com/item/177writing-skills-hedge-words.html.

Word complexity features: These features capture the justification's complexity of word choice, namely the justification's word entropy, type-token ratio, and grade level (Kincaid et al., 1975).

Word score-based features: Warriner et al. (2013) and Brysbaert et al. (2014) associate each word in a lexicon with four real-valued numbers describing how abstract, intensely emotional, pleasant, and vulnerability-evoking the word is. We extract as features the average value of the words in an argument for each of these qualities.

Duplicated Tan et al. (Tan) As our fifth baseline, we employ our re-implementation of Tan et al.'s (2016) system. Their feature set comprises all the features described in the *Style*, *BOW*, and *BOPOS* baselines, as well as a set of word score-based features exactly like those described above, except that they involve first quartering the justification, then calculating the word scores on each quarter of the text. These are useful because, for example, successful arguments begin by using calmer words.

Persing and Ng (P&N) The sixth baseline is the system we previously designed and implemented for scoring argument persuasiveness in student essays (Persing and Ng, 2015). This system employs five types of features: (1) POS unigrams, bigrams and trigrams, which capture the syntactic generalizations of an argument's justification; (2) frame-semantic features, which capture the semantic generalizations of the justification; (3) features computed based on the frequency of occurrence of transitional phrases in the justification, which encode its degree of coherence; (4) topic relevance features, which capture the relevance of the justification to its motion based on the number of overlapping entities; and (5) argument label features, which are n-grams of sentence-based argument labels (e.g., CLAIM, SUPPORT) derived from the justification.

6.3 Results and Discussion

Five-fold cross-validation results of the six baselines when trained on 100% of the training data (966 arguments) are shown in the first six rows of Table 5. While BOW and WNG serve as strong baselines for many NLP tasks, the same is not true for persuasiveness scoring: they are among the worst baselines. This is perhaps not surprising given the discussion in the introduction: since persuasiveness scoring is a discourse-level task, in many cases an argument's persuasiveness cannot be determined solely from its words and phrases. The best baselines are Style and Tan, a system that builds upon Style. These systems offer considerably better performance than BOW, WNG, BOPOS, and P&N w.r.t. all four scoring metrics.

System	E	ME	MSE	PC
BOW	0.786	1.218	2.087	0.073
WNG	0.786	1.218	2.088	0.063
BOPOS	0.786	1.217	2.084	0.089
Style	0.748	1.102	1.776	0.408
Tan	0.744	1.109	1.799	0.398
P&N	0.785	1.198	2.045	0.252
ASE	0.744	**1.097**	**1.753**	**0.422**

Table 5: Five-fold cross-validation results for persuasiveness scoring. Each baseline is trained on 100% of the training data (966 arguments), while ASE is trained on 96 arguments (10% of the available training data).

Results of our system, ASE, are shown in the last row of Table 5. These results are obtained when n is set to 12. Recall that n is a parameter of ASE that specifies the number of persuasiveness-labeled training arguments used to compute each cluster's persuasiveness score. Since we have eight clusters, these results are obtained when ASE is trained on 96 persuasiveness-labeled arguments (10% of the training data). Although ASE is lightly-supervised, it outperforms all the baseline systems by all four metrics. The improvements it yields are highly significant w.r.t. three of the four scoring metrics.[16] These results provide suggestive evidence for the efficacy of our error-modeling approach to persuasiveness scoring.

6.4 Additional Experiments

To gain additional insights into ASE, we perform additional experiments.

Lightly-supervised baselines. To be fair in our comparison with the baselines, we retrain them on 10% of the arguments randomly sampled from the training data and compare their performances against ASE. Results are shown in Table 6. In comparison to the results in Table 5, almost all baselines suffer from performance deterioration, particularly w.r.t. ME, MSE, and PC. ASE continues to significantly outperform all baselines

[16] Unless otherwise stated, boldfaced results are highly significant compared to the best baseline ($p < .01$, paired t-test).

System	E	ME	MSE	PC
BOW	0.788	1.245	2.216	0.011
WNG	0.789	1.245	2.217	0.013
BOPOS	0.789	1.245	2.213	0.044
Style	0.755	1.239	2.343	0.261
Tan	0.755	1.238	2.340	0.267
P&N	0.791	1.291	2.432	0.147
ASE	0.744	**1.097**	**1.753**	**0.422**

Table 6: Results for persuasiveness scoring when all systems are trained on 10% of the training instances.

F #	E	ME	MSE	PC
1	0.749	1.112	1.795	0.415
2	0.751	1.112	1.803	0.405†
3	0.745	1.096	1.764	0.415
4	0.752	1.104	1.759	0.416
5	0.752†	1.114	1.802†	0.401†
6	0.748	1.109	1.798	0.407
7	0.744	1.11	1.811	0.413
8	0.753	1.105†	1.772	0.412
9	0.753	1.113	1.817†	0.400†
10	0.755	1.118	1.811	0.398

Table 7: Results for persuasiveness scoring when one feature is removed from ASE's generative model. F # indicates which feature is being referred to (as indexed in Table 4).

w.r.t. these three scoring metrics.

Feature ablation. In order to determine each feature's contribution to ASE's generative model, we perform ablation experiments wherein we retrain the model using all but one of the features. Table 7 shows how ASE performs after each feature is removed.[17]

From these results, we gather that no feature makes a negative contribution to the model, as no feature's removal significantly improves performance on any metric. Occurrences of "morally", first person plural pronouns, the number of content lemmas appearing in both the assertion and the justification, and justification length (features 2, 5, 8 and 9) make significant contributions to performance according to at least one metric.

Error ablation. Recall that ASE predicts persuasiveness based on a summation of the predicted severity scores over all errors. To determine the importance of each error's contribution, we perform five ablation experiments wherein we exclude each one of the errors from this summation.

[17]Unless otherwise stated, results that are significantly worse than that of the original model ($p < .01$, paired t-test) are marked with a dagger.

Error	E	ME	MSE	PC
GE	0.745	1.082	1.71	0.443
LO	0.746	1.098	1.758	0.416
IS	0.766	1.171†	1.954†	0.317†
UA	0.743	1.106†	1.789†	0.409†
UJ	0.763†	1.132†	1.862†	0.367†

Table 8: Results for persuasiveness scoring when ASE predicts persuasiveness based on a summation of severity scores over all but one error. The error shown in each row is the ablated error.

n	E	ME	MSE	PC
1	0.771	1.430	3.393	0.240
2	0.750	1.304	2.688	0.299
3	0.758	1.221	2.302	0.320
4	0.746	1.177	2.113	0.348
5	0.747	1.153	1.980	0.361
6	0.749	1.151	1.969	0.374
7	0.752	1.143	1.918	0.379
8	0.747	1.119	1.842	0.386
9	0.754	1.104	1.790	0.407
10	0.754	1.112	1.808	0.413

Table 9: Learning curve results.

Table 8 shows that the IS, UA, and UJ errors make the most important contributions to persuasiveness scoring since removing them from consideration significantly harms performance compared to ASE. Removing GE and LO, by contrast, harms performance the least since these are likely the two least frequent errors and therefore have less impact on performance.

Learning curve. Table 9 shows how our ASE system performs when n increases from 1 to 10. As we can see, the scores for all metrics with the exception of E follow the expected trajectory of a learning curve, with worse scores for $n = 1$ progressively becoming better as n approaches 10.

Fully-supervised results. Can ASE perform better given more training data? Table 10 shows the results of ASE when it is trained on 10% (ASE(10)) and 100% (ASE(100)) of the training data. As we can see, the answer is yes: ASE(100) significantly outperforms ASE(10) w.r.t. all but the E metric. While it is not surprising to see diminishing returns, what is perhaps surprising is the relative small performance gap between ASE(10)

System	E	ME	MSE	PC
ASE(10)	0.744	1.097	1.753	0.422
ASE(100)	0.745	**1.078**	**1.678**	**0.441**

Table 10: Results for persuasiveness scoring when ASE is trained on 10% (row 1) and 100% (row 2) of the training data.

Train data	E	ME	MSE	PC
10%	0.739	1.110	1.901	0.408
100%	0.731	1.053	1.724	0.454

Table 11: Results for persuasiveness scoring when ASE predicts persuasiveness by means of a support vector regressor that is trained on 10% (row 1) and 100% (row 2) of the training data using only the error severity values as features.

and ASE(100): it suggests that ASE learns very fast from a small amount of labeled data.

Training a persuasiveness predictor with errors as features. Recall that ASE predicts persuasiveness based on the sum of severity scores. Can we instead predict persuasiveness by training a regressor using only the errors as features? To answer this question, we train a support vector regressor using the 13 binary features that correspond to the 13 severity values of the five errors. The value of a feature is 1 if and only if the argument is assigned the corresponding severity value.

Results of the regressor are shown in Table 11. When the regressor is trained on 10% of the training data, its results are worse than the ASE(10) results in Table 10 w.r.t. all but the E metric. However, when it is trained on 100% of the training data, its results are slightly better than the ASE(100) results in Table 10 w.r.t. all but the MSE metric. We speculate that being discriminatively trained, the support vector regressor can yield better results than the simplistic modeling assumption made by ASE only when training data is plentiful. Additional experiments are needed to determine the reason, however.

Correlation. Recall that ASE assumes that the persuasiveness score of an argument correlates with the sum of its severity scores over all errors. To better understand the extent to which this assumption is true, we cluster *all* 1,208 arguments by the sum of severity scores. For each of the eight resulting clusters, we average the gold persuasiveness scores of the arguments in the cluster.

Results are shown in Table 12. As we can see,

SS	Average	SD	SS	Average	SD
0	5.138	1.050	4	3.111	1.331
1	4.886	1.190	5	3.245	1.392
2	4.194	1.338	6	2.909	1.446
3	3.763	1.306	7	3.000	0.000

Table 12: Average persuasiveness score and its standard deviation (SD) against the sum of severity score (SS).

the data satisfies our assumption: average persuasiveness decreases as the sum of severity scores increases. The only exception occurs when SS=7, presumably due to the small sample size.

6.5 Error Analysis

Next, we conduct a qualitative error analysis.

While there is a definite correlation between error severity count and persuasiveness, error severity count is likely not the only factor that impacts persuasiveness. An examination of some essays whose persuasiveness scores are far from their ASE predicted scores shows that factors that are harder to analyze such as logical soundness and the presence of claims that seem to contradict the assertion also play a role in persuasiveness.

In other arguments, persuasiveness prediction error can be attributed to our system's misprediction of the presence/absence/severity of an error. Error annotated data might help address this problem by allowing us to tune our error severity heuristics.

Finally, our error severity scales are pretty coarse-grained. It is reasonable to expect, for example, that a real argument could have an unclear justification error whose severity is halfway between 2 and 1, but our EM algorithm for predicting error severities does not allow this. The introduction of a regression system into our algorithm might address this problem.

7 Conclusion

We proposed a lightly-supervised approach to the under-studied problem of predicting argument persuasiveness scores on debate arguments. Experimental results on 1,208 arguments demonstrated that our approach significantly outperformed six fully-supervised baselines by three out of four scoring metrics when using only 10% of the training data. To stimulate research on this task, we make our annotated data publicly available.

Acknowledgments

We thank the three anonymous reviewers for their detailed comments. We also thank our annotators, Dino Occhialini and Christopher Knoll. This work was supported in part by NSF Grants IIS-1219142 and IIS-1528037. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of NSF.

References

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods* 46(3):904–911.

Chih-Chung Chang and Chih-Jen Lin. 2001. *LIBSVM: A library for Support Vector Machines*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Ulla Connor. 1990. Linguistic/rhetorical measures for international persuasive student writing. *Research in the Teaching of English* pages 67–87.

Ulla Connor and Janice Lauer. 1985. Understanding persuasive essay writing: Linguistic/Rhetorical approach. *Text-Interdisciplinary Journal for the Study of Discourse* 5(4):309–326.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39:1–38.

Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in Neural Information Processing Systems* 9, pages 155–161.

Ivan Habernal and Iryna Gurevych. 2016a. What makes a convincing argument? Empirical analysis and detecting attributes of convincingness in Web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223.

Ivan Habernal and Iryna Gurevych. 2016b. Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599.

J. Peter Kincaid, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for Navy enlisted personnel. Technical report, DTIC Document.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel R. Tetreault. 2014. *Automated Grammatical Error Detection for Language Learners*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, second edition.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184, 2014.

Stephanie Lukin, Pranav Anand, Marilyn Walker, and Steve Whittaker. 2017. Argument strength is in the eye of the beholder: Audience effects in persuasion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. pages 742–753.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. pages 55–60.

Saif Mohammad and Tony Yang. 2011. Tracking sentiment in mail: How genders differ on emotional axes. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 70–79.

Witri Oktavia, Anas Yasin, et al. 2014. An analysis of students' argumentative elements and fallacies in students' discussion essays. *English Language Teaching* 2(3).

Shereen Oraby, Lena Reed, Ryan Compton, Ellen Riloff, Marilyn Walker, and Steve Whittaker. 2015. And that's a fact: Distinguishing factual and emotional argumentation in online dialogue. In *Proceedings of the 2nd Workshop on Argumentation Mining*. pages 116–126.

Isaac Persing and Vincent Ng. 2015. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 543–552.

Isaac Persing and Vincent Ng. 2017. Why can't you convince me? Modeling weaknesses in unpersuasive arguments. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. pages 4082–4088.

Richard E. Petty and John T. Cacioppo. 1984. The effects of involvement on responses to argument quantity and quality: Central and peripheral routes to persuasion. *Journal of Personality and Social Psychology* 46(1):69.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. pages 105–112.

Mark D. Shermis, Jill Burstein, Derrick Higgins, and Klaus Zechner. 2010. Automated essay scoring: Writing assessment and instruction. In *International Encyclopedia of Education (3rd edition)*, Elsevier, Oxford, UK.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International World Wide Web Conference*, pages 613–624.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior Research Methods* 45(4):1191–1207.

Zhongyu Wei, Yang Liu, and Yi Li. 2016. Is this post persuasive? Ranking argumentative comments in online forum. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 195–200.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*. pages 34–35.

Appendix: Reference Extraction and Internal Citation Cleanup

Given an argument, we employ the following steps to extract references and clean up internal citations.

1. We identify digits and locations in the justification that appear to refer to references. Particularly, we identify each digit in the text that satisfies all the following conditions: (a) it appears next to a punctuation (because the digits usually occur right before or after a sentence's end punctuation); (b) it does not appear next to another digit (because these are short arguments, a digit next to another digit is probably not a citation); (c) if we take the string consisting of the digit, the punctuation, the character next to the digit other than the punctuation, and the character on the other side of the punctuation, this string is not parsable as a floating point number (if it is part of a floating point number, it is probably not a citation); (d) a '$' does not appear before it; (e) a '%' does not appear after it; (f) a '/' does not appear before it; and (g) a '/' does not appear after it.
2. We make a list of all digits identified in step 1 that occur at least twice in the text. (A digit needs to occur twice in order to be a citation because the first time it occurs in the justification text and the last time it occurs right before the reference.)
3. We sort the digits from step 2 in numerical order. We remove 0 from the list if it is present. If there are any gaps in the list (e.g., if '1', '2', and '4' appear in the list), we discard any digits after the gap. (People do not use 0 to make references. And if there are gaps, it usually means that whatever number appears after the gap was erroneously identified as a citation because people do not skip digits when numbering their references.)
4. We sequentially scan the list from step 3. If, at any point in the list, the last location (in the justification) of the digit we are examining occurs before the last location (in the justification) of the previous digit in the list, we discard the digit and all the digits after it in the list. (We expect references to begin with the last occurrences of their corresponding numbers. If one of the digits' last occurrences seems out of order, that means there is a problem with the list so we cannot rely on it beyond this point.)
5. We split the text according to the locations in the justification of the digits that remain in the list from step 4. The first text segment is the justification's text, and all the remaining segments are individual references.
6. Finally, we do some cleanup of the text. We remove all the digits identified in step 1 occurring in the justification's text. (This should help with parsing because the digits make sentences grammatically incorrect.) We also remove all occurrences of '[' or ']' in the text (because some people surround their citation digits with them). Finally we replace any urls (starting with "http:") with "url".