Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP), Beijing, China, July 2015, pp. 543--552.

Modeling Argument Strength in Student Essays

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

While recent years have seen a surge of interest in automated essay grading, including work on grading essays with respect to particular dimensions such as prompt adherence, coherence, and technical quality, there has been relatively little work on grading the essay dimension of argument strength, which is arguably the most important aspect of argumentative essays. We introduce a new corpus of argumentative student essays annotated with argument strength scores and propose a supervised, feature-rich approach to automatically scoring the essays along this dimension. Our approach significantly outperforms a baseline that relies solely on heuristically applied sentence argument function labels by up to 16.1%.

1 Introduction

Automated essay scoring, the task of employing computer technology to evaluate and score written text, is one of the most important educational applications of natural language processing (NLP) (see Shermis and Burstein (2003) and Shermis et al. (2010) for an overview of the state of the art in this task). A major weakness of many existing scoring engines such as the Intelligent Essay Assessor™(Landauer et al., 2003) is that they adopt a holistic scoring scheme, which summarizes the quality of an essay with a single score and thus provides very limited feedback to the writer. In particular, it is not clear which dimension of an essay (e.g., style, coherence, relevance) a score should be attributed to. Recent work addresses this problem by scoring a particular dimension of essay quality such as coherence (Miltsakaki and Kukich, 2004), technical errors, relevance to prompt (Higgins et al., 2004; Persing and Ng, 2014), organization (Persing et al., 2010), and thesis clarity (Persing and Ng, 2013). Essay grading software that provides feedback along multiple dimensions of essay quality such as E-*rater*/Criterion (Attali and Burstein, 2006) has also begun to emerge.

Our goal in this paper is to develop a computational model for scoring the essay dimension of *argument strength*, which is arguably the most important aspect of argumentative essays. Argument strength refers to the strength of the argument an essay makes for its thesis. An essay with a high argument strength score presents a strong argument for its thesis and would convince most readers. While there has been work on *designing* argument schemes (e.g., Burstein et al. (2003), Song et al. (2014), Stab and Gurevych (2014a)) for *annotating* arguments manually (e.g., Song et al. (2014), Stab and Gurevych (2014b)) and automatically (e.g., Falakmasir et al. (2014), Song et al. (2014)) in student essays, little work has been done on scoring the argument strength of student essays. It is worth mentioning that some work has investigated the use of automatically determined argument labels for heuristic (Ong et al., 2014) and learning-based (Song et al., 2014) essay scoring, but their focus is *holistic* essay scoring, not argument strength essay scoring.

In sum, our contributions in this paper are twofold. First, we develop a scoring model for the argument strength dimension on student essays using a *feature-rich* approach. Second, in order to stimulate further research on this task, we make our data set consisting of argument strength annotations of 1000 essays publicly available. Since progress in argument strength modeling is hindered in part by the lack of a publicly annotated corpus, we believe that our data set will be a valuable resource to the NLP community.

2 Corpus Information

We use as our corpus the 4.5 million word International Corpus of Learner English (ICLE) (Granger

Topic	Languages	Essays
Most university degrees are theoretical and do not prepare students for the real world. They are therefore of very little value.	13	131
The prison system is outdated. No civilized society should punish its criminals: it should rehabilitate them.	11	80
In his novel *Animal Farm*, George Orwell wrote "All men are equal but some are more equal than others." How true is this today?	10	64

Table 1: Some examples of writing topics.

et al., 2009), which consists of more than 6000 essays on a variety of different topics written by university undergraduates from 16 countries and 16 native languages who are learners of English as a Foreign Language. 91% of the ICLE texts are written in response to prompts that trigger argumentative essays. We select 10 such prompts, and from the subset of argumentative essays written in response to them, we select 1000 essays to annotate for training and testing of our essay argument strength scoring system. Table 1 shows three of the 10 topics selected for annotation. Fifteen native languages are represented in the set of annotated essays.

3 Corpus Annotation

We ask human annotators to score each of the 1000 argumentative essays along the argument strength dimension. Our annotators were selected from over 30 applicants who were familiarized with the scoring rubric and given sample essays to score. The six who were most consistent with the expected scores were given additional essays to annotate. Annotators evaluated the argument strength of each essay using a numerical score from one to four at half-point increments (see Table 2 for a description of each score).[1] This contrasts with previous work on essay scoring, where the corpus is annotated with a binary decision (i.e., *good* or *bad*) for a given scoring dimension (e.g., Higgins et al. (2004)). Hence, our annotation scheme not only provides a finer-grained distinction of argument strength (which can be important in practice), but also makes the prediction task more challenging.

[1]See our website at `http://www.hlt.utdallas.edu/~persingq/ICLE/` for the complete list of argument strength annotations.

Score	Description of Argument Strength
4	essay makes a **strong argument** for its thesis and would convince most readers
3	essay makes a **decent argument** for its thesis and could convince some readers
2	essay makes a **weak argument** for its thesis or sometimes even **argues against it**
1	essay **does not make an argument** or it is often **unclear what the argument is**

Table 2: Descriptions of the meaning of scores.

To ensure consistency in annotation, we randomly select 846 essays to have graded by multiple annotators. Though annotators exactly agree on the argument strength score of an essay only 26% of the time, the scores they apply fall within 0.5 points in 67% of essays and within 1.0 point in 89% of essays. For the sake of our experiments, whenever the two annotators disagree on an essay's argument strength score, we assign the essay the average the two scores rounded down to the nearest half point. Table 3 shows the number of essays that receive each of the seven scores for argument strength.

score	1.0	1.5	2.0	2.5	3.0	3.5	4.0
essays	2	21	116	342	372	132	15

Table 3: Distribution of argument strength scores.

4 Score Prediction

We cast the task of predicting an essay's argument strength score as a regression problem. Using regression captures the fact that some pairs of scores are more similar than others (e.g., an essay with an argument strength score of 2.5 is more similar to an essay with a score of 3.0 than it is to one with a score of 1.0). A classification system, by contrast, may sometimes believe that the scores 1.0 and 4.0 are most likely for a particular essay, even though these scores are at opposite ends of the score range. In the rest of this section, we describe how we train and apply our regressor.

Training the regressor. Each essay in the training set is represented as an instance whose label is the essay's gold score (one of the values shown in Table 3), with a set of baseline features (Section 5) and up to seven other feature types we propose (Section 6). After creating training instances, we train a linear regressor with regularization parameter c for scoring test essays using the linear SVM regressor implemented in the LIBSVM software package (Chang and Lin, 2001). All SVM-specific learning parameters are set to their default values except c, which we tune to maximize performance on held-out validation data.[2]

Applying the regressor. After training the regressor, we use it to score the test set essays. Test instances are created in the same way as the training instances. The regressor may assign an essay any score in the range of 1.0−4.0.

5 Baseline Systems

In this section, we describe two baseline systems for predicting essays' argument strength scores.

5.1 Baseline 1: Most Frequent Baseline

Since there is no existing system specifically for scoring argument strength, we begin by designing a simple baseline. When examining the score distribution shown in Table 3, we notice that, while there exist at least a few essays having each of the seven possible scores, the essays are most densely clustered around scores 2.5 and 3.0. A system that always predicts one of these two scores will very frequently be right. For this reason, we develop a *most frequent* baseline. Given a training set, Baseline 1 counts the number of essays assigned to each of the seven scores. From these counts, it determines which score is most frequent and assigns this most frequent score to each test essay.

5.2 Baseline 2: Learning-based Ong et al.

Our second baseline is a learning-based version of Ong et al.'s (2014) system. Recall from the introduction that Ong et al. presented a rule-based approach to predict the holistic score of an argumentative essay. Their approach was composed of two steps. First, they constructed eight heuristic rules for automatically labeling each of the sentences in their corpus with exactly one of the following argument labels: OPPOSES, SUPPORTS, CITATION, CLAIM, HYPOTHESIS, CURRENT STUDY, or NONE. After that, they employed these sentence labels to construct five heuristic rules to holistically score a student essay.

We create Baseline 2 as follows, employing the methods described in Section 4 for training, parameter tuning, and testing. We employ Ong et al.'s method to tag each sentence of our essays with an argument label, but modify their method to accommodate differences between their and our corpus. In particular, our more informal corpus does not contain CURRENT STUDY or CITATION sentences, so we removed portions of rules that attempt to identify these labels (e.g. portions of rules that search for a four-digit number, as would appear as the year in a citation). Our resulting rule set is shown in Table 4. If more than one of these rules applies to a sentence, we tag it with the label from the earliest rule that applies.

#	Rule
1	Sentences that begin with a comparison discourse connective or contain any string prefixes from "conflict" or "oppose" are tagged OPPOSES.
2	Sentences that begin with a contingency connective are tagged SUPPORTS.
3	Sentences containing any string prefixes from "suggest", "evidence", "shows", "Essentially", or "indicate" are tagged CLAIM.
4	Sentences in the first, second, or last paragraph that contain string prefixes from "hypothes", or "predict", but do not contain string prefixes from "conflict" or "oppose" are tagged HYPOTHESIS.
5	Sentences containing the word "should" that contain no contingency connectives or string prefixes from "conflict" or "oppose" are also tagged HYPOTHESIS.
6	If the previous sentence was tagged hypothesis and this sentence begins with an expansion connective, it is also tagged HYPOTHESIS.
7	Do not apply a label to this sentence.

Table 4: Sentence labeling rules.

After labeling all the sentences in our corpus, we then convert three of their five heuristic scoring rules into features for training a regressor.[3] The resulting three features describe (1) whether an essay contains at least one sentence labeled HYPOTHESIS, (2) whether it contains at least one sentence labeled OPPOSES, and (3) the sum of CLAIM sentences and SUPPORTS sentences divided by the number of paragraphs in the essay. If the value of the last feature exceeds 1, we instead assign it a value of 1. These features make sense because, for example, we would expect essays containing lots of SUPPORTS sentences to offer stronger arguments.

6 Our Approach

Our approach augments the feature set available to Baseline 2 with seven types of novel features.

1. POS N-grams (POS) Word n-grams, though commonly used as features for training text classifiers, are typically not used in automated essay

[2]For parameter tuning, we employ the following c values: 10^0 10^1, 10^2, 10^3, 10^4, 10^5, or 10^6.

[3]We do not apply the remaining two of their heuristic scoring rules because they deal solely with current studies and citations.

grading. The reason is that any list of word n-gram features automatically compiled from a given set of training essays would be contaminated with prompt-specific n-grams that may make the resulting regressor generalize less well to essays written for new prompts.

To generalize our feature set in a way that does not risk introducing prompt-dependent features, we introduce POS n-gram features. Specifically, we construct one feature from each sequence of $1-5$ part-of-speech tags appearing in our corpus. In order to obtain one of these features' values for a particular essay, we automatically label each essay with POS tags using the Stanford CoreNLP system (Manning et al., 2014), then count the number of times the POS tag sequence occurs in the essay. An example of a useful feature of this type is "CC NN ,", as it is able to capture when a student writes either "for instance," or "for example,". We normalize each essay's set of POS n-gram features to unit length.

2. Semantic Frames (SFR) While POS n-grams provide syntactic generalizations of word n-grams, FrameNet-style semantic role labels provide semantic generalizations. For each essay in our data set, we employ SEMAFOR (Das et al., 2010) to identify each semantic frame occurring in the essay as well as each frame element that participates in it. For example, a semantic frame may describe an event that occurs in a sentence, and the event's frame elements may be the people or objects that participate in the event. For a more concrete example, consider the sentence "I said that I do not believe that it is a good idea". This sentence contains a Statement frame because a statement is made in it. One of the frame elements participating in the frame is the Speaker "I". From this frame, we would extract a feature pairing the frame together with its frame element to get the feature "Statement-Speaker-I". We would expect this feature to be useful for argument strength scoring because we noticed that essays that focus excessively on the writer's personal opinions and experiences tended to receive lower argument strength scores.

As with POS n-grams, we normalize each essay's set of Semantic Frame features to unit length.

3. Transitional Phrases (TRP) We hypothesize that a more cohesive essay, being easier for a reader to follow, is more persuasive, and thus makes a stronger argument. For this reason, it would be worthwhile to introduce features that measure how cohesive an essay is. Consequently, we create features based on the 149 transitional phrases compiled by Study Guides and Strategies[4]. Study Guides and Strategies collected these transitions into lists of phrases that are useful for different tasks (e.g. a list of transitional phrases for restating points such as "in essence" or "in short"). There are 14 such lists, which we use to generalize transitional features. Particularly, we construct a feature for each of the 14 phrase type lists. For each essay, we assign the feature a value indicating the average number of transitions from the list that occur in the essay per sentence. Despite being phrase-based, transitional phrases features are designed to capture only *prompt-independent* information, which as previously mentioned is important in essay grading.

4. Coreference (COR) As mentioned in our discussion of transitional phrases, a strong argument must be cohesive so that the reader can understand what is being argued. While the transitional phrases already capture one aspect of this, they cannot capture when transitions are made via repeated mentions of the same entities in different sentences. We therefore introduce a set of 19 coreference features that capture information such as the fraction of an essay's sentences that mention entities introduced in the prompt, and the average number of total mentions per sentence.[5] Calculating these feature values, of course, requires that the text be annotated with coreference information. We automatically coreference-annotate the essays using the Stanford CoreNLP system.

5. Prompt Agreement (PRA) An essay's prompt is always either a single statement, or can be split up into multiple statements with which a writer may AGREE STRONGLY, AGREE SOMEWHAT, be NEUTRAL, DISAGREE SOMEWHAT, DISAGREE STRONGLY, NOT ADDRESS, or explicitly have NO OPINION on. We believe information regarding which of these categories a writer's opinion falls into has some bearing on the strength of her argument because, for example, a writer who explicitly mentions having no opinion has probably not made a persuasive argument.

For this reason, we annotate a subset of 830 of our ICLE essays with these agreement labels. We then train a multiclass maximum entropy classifier

[4] `http://www.studygs.net/wrtstr6.htm`

[5] See our website at `http://www.hlt.utdallas.edu/~persingq/ICLE/` for a complete list of coreference features.

using MALLET (McCallum, 2002) for identifying which one of these seven categories an author's opinion falls into. The feature set we use for this task includes POS n-gram and semantic frame features as described earlier in this section, lemmatized word 1-3 grams, the keyword and prompt adherence keyword features we described in Persing and Ng (2013) and Persing and Ng (2014), respectively, and a feature indicating which statement in the prompt we are attempting to classify the author's agreement level with respect to.

Our classifier's training set in this case is the subset of prompt agreement annotated essays that fall within the training set of our 1000 essay argument strength annotated data. We then apply the trained classifier to our entire 1000 essay set in order to obtain predictions from which we can then construct features for argument strength scoring. For each prediction, we construct a feature indicating which of the seven classes the classifier believes is most likely, as well as seven additional features indicating the probability the classifier associates with each of the seven classes.

We produce additional related annotations on this 830 essay set in cases when the annotated opinion was neither AGREE STRONGLY nor DISAGREE STRONGLY, as the reason the annotator chose one of the remaining five classes may sometimes offer insight into the writer's argument. The classes of reasons we annotate include cases when the writer: (1) offered CONFLICTING OPINIONS, (2) EXPLICITLY STATED an agreement level, (3) gave only a PARTIAL RESPONSE to the prompt, (4) argued a SUBTLER POINT not capturable by extreme opinions, (5) did not make it clear that the WRITER'S POSITION matched the one she argued, (6) only BRIEFLY DISCUSSED the topic, (7) CONFUSINGLY PHRASED her argument, or (8) wrote something whose RELEVANCE to the topic was not clear. We believe that knowing which reason(s) apply to an argument may be useful for argument strength scoring because, for example, the CONFLICTING OPINIONS class indicates that the author wrote a confused argument, which probably deserves a lower argument strength score.

We train eight binary maximum entropy classifiers, one for each of these reasons, using the same training data and feature set we use for agreement level prediction. We then use the trained classifiers to make predictions for these eight reasons on all 1000 essays. Finally, we generate features for our argument strength regressor from these predictions by constructing two features from each of the eight reasons. The first binary feature is turned on whenever the maximum entropy classifier believes that the reason applies (i.e., when it assigns the reason a probability of over 0.5). The second feature's value is the probability the classifier assigns for this reason.

6. Argument Component Predictions (ACP) Many of our features thus far do not result from an attempt to build a deep understanding of the structure of the arguments within our essays. To introduce such an understanding into our system, we follow Stab and Gurevych (2014a), who collected and annotated a corpus of 90 persuasive essays (not from the ICLE corpus) with the understanding that the arguments contained therein consist of three types of argument components. In one essay, these argument components typically include a MAJOR CLAIM, several lesser CLAIMs which usually support or attack the major claim, and PREMISEs which usually underpin the validity of a claim or major claim.

Stab and Gurevych (2014b) trained a system to identify these three types of argument components within their corpus given the components' boundaries. Since our corpus does not contain annotated argument components, we modify their approach in order to simultaneously identify argument components and their boundaries.

We begin by implementing a maximum entropy version of their system using MALLET for performing the argument component identification task. We feed our system the same structural and lexical features they described. We then augment the system in the following ways.

First, since our corpus is not annotated with argument component boundaries, we construct a set of low precision, high recall heuristics for identifying the locations in each sentence where an argument component's boundaries might occur. The majority of these rules depend primarily on a syntactic parse tree we automatically generated for the sentence using the Stanford CoreNLP system. Since a large majority of annotated argument components are substrings of a simple declarative clause (an S node in the parse tree), we begin by identifying each S node in the sentence's tree.

Given one of these clauses, we collect a list of left and right boundaries where an argument component may begin or end. The rules we used to

(a) Potential left boundary locations

#	Rule
1	Exactly where the S node begins.
2	After an initial explicit connective, or if the connective is immediately followed by a comma, after the comma.
3	After nth comma that is an immediate child of the S node.
4	After nth comma.

(b) Potential right boundary locations

#	Rule
5	Exactly where the S node ends, or if S ends in a punctuation, immediately before the punctuation.
6	If the S node ends in a (possibly nested) SBAR node, immediately before the nth shallowest SBAR.[6]
7	If the S node ends in a (possibly nested) PP node, immediately before the nth shallowest PP.

Table 5: Rules for extracting candidate argument component boundary locations.

find these boundaries are summarized in Table 5.

Given an S node, we use our rules to construct up to $l \times r$ argument component candidate instances to feed into our system by combining each left boundary with each right boundary that occurs after it, where l is the number of potential left boundaries our rules found, and r is the number of right boundaries they found.

The second way we augment the system is by adding a boundary rule feature type. Whenever we generate an argument component candidate instance, we augment its normal feature set with two binary features indicating which heuristic rule was used to find the candidate's left boundary, and which rule was used to find its right boundary. If two rules can be used to find the same left or right boundary position, the first rule listed in the table is the one used to create the boundary rule feature. This is why, for example, the table contains multiple rules that can find boundaries at comma locations. We would expect some types of commas (e.g., ones following an explicit connective) to be more significant than others.

A last point that requires additional explanation is that several of the rules contain the word "nth". This means that, for example, if a sentence contains multiple commas, we will generate multiple left boundary positions for it using rule 4, and the left boundary rule feature associated with each position will be different (e.g., there is a unique feature for the first comma, and for the the second comma, etc.).

The last augmentation we make to the system is that we apply a NONE label to all argument component candidates whose boundaries do not exactly match those of a gold standard argument component. While Stab and Gurevych also did this, their list of such argument component candidates consisted solely of sentences containing no argument components at all. We could not do this, however, since our corpus is not annotated with argument components and we therefore do not know which sentences these would be.

We train our system on all the instances we generated from the 90 essay corpus and apply it to label all the instances we generated in the same way from our 1000 essay ICLE corpus. As a result, we end up with a set of automatically generated argument component annotations on our 1000 essay corpus. We use these annotations to generate five additional features for our argument strength scoring SVM regressor. These features' values are the number of major claims in the essay, the number of claims in the essay, the number of premises in the essay, the fraction of paragraphs that contain either a claim or a major claim, and the fraction of paragraphs that contain at least one argument component of any kind.

7. Argument Errors (ARE) We manually identified three common problems essays might have that tend to result in weaker arguments, and thus lower argument strength scores. We heuristically construct three features, one for each of these problems, to indicate to the learner when we believe an essay has one of these problems.

It is difficult to make a reasonably strong argument in an essay that is too short. For this reason, we construct a feature that encodes whether the essay has 15 or fewer sentences, as only about 7% of our essays are this short.

In the Stab and Gurevych corpus, only about 5% of paragraphs have no claims or major claims in them. We believe that an essay that contains too many of these claim or major claim-less paragraphs may have an argument that is badly structured, as it is typical for a paragraph to contain one or two (major) claim(s). For this reason, we construct a feature that encodes whether more than half of the essay's paragraphs contain no claims or major claims, as indicated by the previously generated automatic annotations.

[6]The S node may end in an SBAR node which itself has an SBAR node as its last child, and so on. In this case, the S node could be said to end with any of these "nested" SBARS, so we use the position before each (nth) one as a right boundary.

Similarly, only 5% of the Stab and Gurevych essays contain no argument components at all. We believe that an essay that contains too many of these component-less paragraphs is likely to have taken too much space discussing issues that are not relevant to the main argument of the essay. For this reason, we construct a feature that encodes whether more than one of the essay's paragraphs contain no components, as indicated by the previously generated automatic annotations.

7 Evaluation

In this section, we evaluate our system for argument strength scoring. All the results we report are obtained via five-fold cross-validation experiments. In each experiment, we use 60% of our labeled essays for model training, another 20% for parameter tuning and feature selection, and the final 20% for testing. These correspond to the training set, held-out validation data, and test set mentioned in Section 4.

7.1 Scoring Metrics

We employ four evaluation metrics. As we will see below, $S1$, $S2$, and $S3$ are *error* metrics, so lower scores on them imply better performance. In contrast, PC is a *correlation* metric, so higher correlation implies better performance.

The simplest metric, $S1$, measures the frequency at which a system predicts the wrong score out of the seven possible scores. Hence, a system that predicts the right score only 25% of the time would receive an $S1$ score of 0.75.

The $S2$ metric measures the average distance between a system's predicted score and the actual score. This metric reflects the idea that a system that predicts scores close to the annotator-assigned scores should be preferred over a system whose predictions are further off, even if both systems estimate the correct score at the same frequency.

The $S3$ metric measures the average square of the distance between a system's score predictions and the annotator-assigned scores. The intuition behind this metric is that not only should we prefer a system whose predictions are close to the annotator scores, but we should also prefer one whose predictions are not too frequently very far away from the annotated scores. The three error metric scores are given by:

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

System	$S1$	$S2$	$S3$	PC
Baseline 1	.668	.428	.321	.000
Baseline 2	.652	.418	.267	.061
Our System	.618	.392	.244	.212

Table 6: Five-fold cross-validation results for argument strength scoring.

where A_j, E_j, and E'_j are the annotator assigned, system predicted, and rounded system predicted scores[7] respectively for essay j, and N is the number of essays.

The last metric, PC, computes Pearson's correlation coefficient between a system's predicted scores and the annotator-assigned scores. PC ranges from -1 to 1. A positive (negative) PC implies that the two sets of predictions are positively (negatively) correlated.

7.2 Results and Discussion

Five-fold cross-validation results on argument strength score prediction are shown in Table 6. The first two rows show our baseline systems' performances. The best baseline system (Baseline 2), which recall is a learning-based version of Ong et al.'s (2014) system, predicts the wrong score 65.2% of the time. Its predictions are off by an average of .418 points, the average squared error of its predictions is .267, and its average Pearson correlation coefficient with the gold argument strength score across the five folds is .061.

Results of our system are shown on the third row of Table 6. Rather than using all of the available features (i.e., Baseline 2's features and the novel features described in Section 6), our system uses only the feature subset selected by the backward elimination feature selection algorithm (Blum and Langley, 1997) that achieves the best performance on the validation data (see Section 7.3 for details). As we can see, our system predicts the wrong score only 61.8% of the time, predicts scores that are off by an average of .392 points, the average squared error of its predictions is .244, and its average Pearson correlation coefficient with the gold scores is .212. These numbers correspond to relative error reductions[8] of 5.2%,

[7]We round all predictions to 1.0 or 4.0 if they fall outside the 1.0−4.0 range and round $S1$ predictions to the nearest half point.

[8]These numbers are calculated $\frac{B-O}{B-P}$ where B is the baseline system's score, O is our system's score, and P is a perfect score. Perfect scores for error measures and PC are 0 and 1 respectively.

6.2%, 8.6%, and 16.1% over Baseline 2 for S1, S2, S3, and PC, respectively, the last three of which are significant improvements[9]. The magnitudes of these improvements suggest that, while our system yields improvements over the best baseline by all four measures, its greatest contribution is that its predicted scores are best-correlated with the gold standard argument strength scores.

7.3 Feature Ablation

To gain insight into how much impact each of the feature types has on our system, we perform feature ablation experiments in which we remove the feature types from our system one-by-one.

We show the results of the ablation experiments on the held-out validation data as measured by the four scoring metrics in Table 7. The top line of each subtable shows what a system that uses all available features's score would be if we removed just one of the feature types. So to see how our system performs by the PC metric if we remove only prompt agreement (PRA) features, we would look at the first row of results of Table 7(d) under the column headed by PRA. The number here tells us that the resulting system's PC score is .303. Since our system that uses all feature types obtains $S1$, $S2$, $S3$, and PC scores of .521, .366, .218, and .341 on the validation data respectively, the removal of PRA features costs the complete system .038 PC points, and thus we can infer that the inclusion of PRA features has a beneficial effect.

From row 1 of Table 7(a), we can see that removing the Baseline 2 feature set (BAS) yields a system with the best $S1$ score in the presence of the remaining feature types in this row. For this reason, we permanently remove the BAS features from the system before we generate the results on line 2. We iteratively remove the feature type that yields a system with the best performance in this way until we get to the last line, where only one feature type is used to generate each result.

Since the feature type whose removal yields the best system is always the rightmost entry in a line, the order of column headings indicates the relative importance of the feature types, with the leftmost feature types being most important to performance and the rightmost feature types being least important in the presence of the other feature types. The score corresponding to the best system is boldfaced for emphasis, indicating that all feature types appearing to its left are included in the best system.[10]

(a) Results using the $S1$ metric

SFR	ACP	TRP	PRA	POS	COR	ARE	BAS
.534	.594	.530	.524	.522	.532	.529	**.521**
.530	.554	.526	.529	.526	.528	.525	
.534	.555	.525	.531	.528	.522		
.543	.558	.536	.530	.527			
.565	.561	.536	.529				
.563	.547	.539					
.592	.550						

(b) Results using the $S2$ metric

POS	PRA	ACP	TRP	BAS	SFR	COR	ARE
.370	.369	.375	.367	.367	.366	.366	.365
.369	.369	.375	.366	.366	.365	.365	
.370	.371	.372	.367	.366	**.365**		
.374	.374	.376	.368	.366			
.377	.375	.374	.368				
.381	.377	.376					
.385	.382						

(c) Results using the $S3$ metric

POS	PRA	ACP	TRP	BAS	COR	ARE	SFR
.221	.220	.225	.219	.218	.217	.217	.211
.220	.219	.221	.214	.212	.211	.211	
.218	.218	.220	.212	.211	**.209**		
.221	.216	.218	.212	.210			
.224	.217	.218	.212				
.228	.220	.219					
.229	.225						

(d) Results using the PC metric

POS	ACP	PRA	TRP	BAS	ARE	COR	SFR
.302	.270	.303	.326	.324	.347	.347	.356
.316	.300	.327	.344	.361	.366	.371	
.346	.331	.341	.356	.367	**.378**		
.325	.331	.345	.362	.375			
.297	.331	.339	.360				
.280	.320	.321					
.281	.281						

Table 7: Feature ablation results. In each subtable, the first row shows how our system would perform on the validation set essays if each feature type was removed. We then remove the least important feature type, and show in the next row how the adjusted system would perform without each remaining type.

It is interesting to note that while the relative importance of different feature types does not remain exactly the same if we measure performance in different ways, we can see that some feature types tend to be more important than others in a majority of the four scoring metrics.

From these tables, it is clear that POS n-grams

[9]All significance tests are paired t-tests with $p < 0.05$.

[10]The reason the performances shown in these tables appear so much better than those shown previously is that in these tables we tune parameters and display results on the validation set in order to make it clearer why we chose to remove each feature type. In Table 6, by contrast, we tune parameters on the validation set, but display results using those parameters on the test set.

	S1			S2			S3			PC		
Gold	.25	.50	.75	.25	.50	.75	.25	.50	.75	.25	.50	.75
1.0	2.90	2.90	2.90	2.74	2.74	2.74	2.74	2.74	2.74	2.74	2.74	2.74
1.5	2.69	2.78	2.89	2.36	2.67	2.78	2.52	2.63	2.71	2.52	2.63	2.81
2.0	2.61	2.72	2.85	2.54	2.69	2.79	2.60	2.69	2.78	2.60	2.70	2.80
2.5	2.64	2.71	2.85	2.65	2.75	2.86	2.66	2.75	2.85	2.69	2.79	2.89
3.0	2.73	2.84	2.92	2.71	2.81	2.91	2.70	2.80	2.90	2.72	2.83	2.90
3.5	2.74	2.85	2.97	2.78	2.89	3.02	2.79	2.90	3.00	2.81	2.90	2.98
4.0	2.75	2.87	3.10	2.76	2.85	3.09	2.76	2.83	3.08	2.81	2.86	3.19

Table 8: Distribution of regressor scores for our system.

(POS), prompt agreement features (PRA), and argument component predictions (ACP) are the most generally important feature types in roughly that order. They all appear in the leftmost three positions under the tables for metrics $S2$, $S3$, and PC, the three metrics by which our system significantly outperforms Baseline 2. Furthermore, removing any of them tends to have a larger negative impact on our system than removing any of the other feature types.

Transitional phrase features (TRP) and Baseline 2 features (BAS), by contrast, are of more middling importance. While both appear in the best feature sets for the aforementioned metrics (i.e., they appear to the left of the boldfaced entry in the corresponding ablation tables), the impact of their removal is relatively less than that of POS, PRA, or ACP features.

Finally, while the remaining three feature types might at first glance seem unimportant to argument strength scoring, it is useful to note that they all appear in the best performing feature set as measured by at least one of the four scoring metrics. Indeed, semantic frame features (SFR) appear to be the most important feature type as measured by the $S1$ metric, despite being one of the least useful feature types as measured by the other performance metrics. From this we learn that when designing an argument strength scoring system, it is important to understand what the ultimate goal is, as the choice of performance metric can have a large impact on what type of system will seem ideal.

7.4 Analysis of Predicted Scores

To more closely examine the behavior of our system, in Table 8 we chart the distributions of scores it predicts for essays having each gold standard score. As an example of how to read this table, consider the number 2.60 appearing in row 2.0 in the .25 column of the $S3$ region. This means that 25% of the time, when our system with parameters tuned for optimizing $S3$ (including the $S3$ feature set as selected in Table 7(c)) is presented with a test essay having a gold standard score of 2.0, it predicts that the essay has a score less than or equal to 2.60.

From this table, we see that our system has a bias toward predicting more frequent scores as the smallest entry in the table is 2.36 and the largest entry is 3.19, and as we saw in Table 3, 71.4% of essays have gold scores in this range. Nevertheless, our system does not rely entirely on bias, as evidenced by the fact that each column in the table has a tendency for its scores to ascend as the gold standard score increases, implying that our system has some success at predicting lower scores for essays with lower gold standard argument strength scores and higher scores for essays with higher gold standard argument strength scores. The major exception to this rule is line 1.0, but this is to be expected since there are only two essays having this gold score, so the sample from which the numbers on this line are calculated is very small.

8 Conclusion

We proposed a feature-rich approach to the new problem of predicting argument strength scores on student essays. In an evaluation on 1000 argumentative essays selected from the ICLE corpus, our system significantly outperformed a baseline system that relies solely on features built from heuristically labeled sentence argument function labels by up to 16.1%. To stimulate further research on this task, we make all of our annotations publicly available.

Acknowledgments

We thank the three anonymous reviewers for their detailed comments. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142. 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

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with E-rater v.2.0. *Journal of Technology, Learning, and Assessment*, 4(3).

Avrim Blum and Pat Langley. 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1–2):245–271.

Jill Burstein, Daniel Marcu, and Kevin Knight. 2003. Finding the WRITE stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39.

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`.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 948–956.

Mohammad Hassan Falakmasir, Kevin D. Ashley, Christian D. Schunn, and Diane J. Litman. 2014. Identifying thesis and conclusion statements in student essays to scaffold peer review. In *Intelligent Tutoring Systems*, pages 254–259. Springer International Publishing.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. 2009. *International Corpus of Learner English (Version 2)*. Presses universitaires de Louvain.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *Human Language Technologies: The 2004 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 185–192.

Thomas K. Landauer, Darrell Laham, and Peter W. Foltz. 2003. Automated scoring and annotation of essays with the Intelligent Essay Assessor™. In *Automated Essay Scoring: A Cross-Disciplinary Perspective*, pages 87–112. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

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.

Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. `http://mallet.cs.umass.edu`.

Eleni Miltsakaki and Karen Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*, 10(1):25–55.

Nathan Ong, Diane Litman, and Alexandra Brusilovsky. 2014. Ontology-based argument mining and automatic essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 24–28.

Isaac Persing and Vincent Ng. 2013. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269.

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543.

Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239.

Mark D. Shermis and Jill C. Burstein. 2003. *Automated Essay Scoring: A Cross-Disciplinary Perspective*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

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.

Yi Song, Michael Heilman, Beata Beigman Klebanov, and Paul Deane. 2014. Applying argumentation schemes for essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 69–78.

Christian Stab and Iryna Gurevych. 2014a. Annotating argument components and relations in persuasive essays. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510.

Christian Stab and Iryna Gurevych. 2014b. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 46–56.