Predicting Argumenthood of English Preposition Phrases

Najoung Kim†, Kyle Rawlins†, Benjamin Van Durme‡ and Paul Smolensky†∆

†Department of Cognitive Science, Johns Hopkins University
‡Center for Language and Speech Processing, Johns Hopkins University
∆Microsoft Research AI
{n.kim,kgr,vandurme,smolensky}@jhu.edu

Abstract

Distinguishing between core and non-core dependents (i.e., arguments and adjuncts) of a verb is a longstanding, non-trivial problem. In natural language processing, argumenthood information is important in tasks such as semantic role labeling (SRL) and preposition phrase (PP) attachment disambiguation. In theoretical linguistics, many diagnostic tests for argumenthood exist but they often yield conflicting and potentially gradient results. This is especially the case for syntactically oblique items such as PPs. We propose two PP argumenthood prediction tasks branching from these two motivations: (1) binary argument/adjunct classification of PPs in VerbNet, and (2) gradient argumenthood prediction using human judgments as gold standard, and report results from prediction models that use pretrained word embeddings and other linguistically informed features. Our best results on each task are (1) acc. = 0.955, F1 = 0.954 (ELMo+BILSTM) and (2) Pearson’s r = 0.624 (word2vec+MLP). Furthermore, we demonstrate the utility of argumenthood prediction in improving sentence representations via performance gains on SRL when a sentence encoder is pretrained with our tasks.

1 Introduction

In theoretical linguistics, a formal distinction is made between “core” and “non-core” elements with respect to a verb. For example, in the following example, the window is the core dependent of the verb open, whereas this morning and with Mary are both non-core dependents.

John opened [the window] [this morning] [with Mary].

What distinguishes core dependents (arguments or complements) from non-core dependents (adjuncts or modifiers), and why is this distinction important? Theoretically, the distinct representations given to arguments and adjuncts manifest in different formal behaviors (Chomsky, 1993; Steedman, 2000). There is also a range of psycholinguistic evidence which supports the psychological reality of the distinction (Tutunjian and Boland, 2008). In natural language processing (NLP), argumenthood information is useful in various applied tasks such as automatic parsing (Briscoe and Carroll, 1998) and PP attachment disambiguation (Merlo and Ferrer, 2006). In particular, automatic distinction of argumenthood could prove useful in improving structure-aware semantic role labeling, which has been shown to outperform structure-agnostic models in recent works (Marcheggiani and Titov, 2017). However, argument/adjunct distinction is one of the most difficult linguistic properties to annotate, and has remained unmarked in popular resources including the Penn TreeBank. PropBank (Palmer et al., 2005) addresses this issue to an extent by providing ARG-N labels, but does not provide full coverage (Hockenmaier and Steedman, 2002). Thus, there are theoretical and practical motivations in developing a systematic approach to predicting argumenthood.

We focus on PP dependents in this paper, which are known to be one of the most challenging verbal dependents to classify correctly (Abend and Rappoport, 2010). The paper is structured as follows. First, we discuss the theoretical and practical motivations for PP argumenthood prediction in more detail and review related works. Second, we formulate two different argumenthood tasks—binary and gradient—and describe how each dataset is constructed. Results for each task using various word embeddings and linguistic features as predictors are reported. Finally, we investigate whether PP argumenthood prediction is indeed useful for NLP. Through a controlled evaluation setup, we demonstrate that training sentence encoders on our proposed tasks improves the quality of learned representations.

2 Argumenthood prediction

Theoretical motivation. Although arguments and adjuncts are theoretically and practically important concepts, distinguishing arguments from adjuncts in practice is not a trivial problem even for linguists (Schütze, 1995). Numerous diagnostic tests have been proposed in the literature; for instance, omissibility and iterability tests are commonly used (Pollard and Sag, 1987). However, none of the existing diagnostic tests (or a set of tests) provide necessary or sufficient criteria to determine the status of a dependent. Moreover, it has long been noted that argumenthood is a gradient phenomenon rather than a strict dichotomy (e.g., some arguments are less argument-like than others, resulting in different syntactic and semantic behaviors of the dependents...
resentations (Ribeiro et al., 2018; Ettinger et al., 2016; Gu-

task (Schütze, 1995; Rissman et al., 2015)). This raises many in-

tion 5) is inspired by a recent line of efforts on

probing performance gains from PP argumenthood information (Sec-

on predicting gradient argumenthood judgment data from frames. Experiment 2, which is a smaller-scale experiment

from adjuncts by an informed selection of linguistic fea-

tion task (Experiment 1) are more comparable to those of

mal grammar in our models, although the use of distribu-

tures. However, we do not use logical forms or explicit for-

from adjuncts distinction errors as well as PP at-

chickle and improve the quality of resources being ported

between different grammars, such as CCGBank.

Argument-adjunct distinction is also closely connected to Semantic Role Labeling (SRL). He et al. (2017) report that even state-of-the-art deep models for SRL still suffer from argument-adjunct distinction errors as well as PP at-

chickle errors. They also observe that errors in widely-

used automatic parsers pose challenges to improving perfor-

in syntax-aware neural models (Marcheggiani and Titov, 2017). This suggests that improving parsers with better argumenthood distinction would lead to better SRL per-

formance.

Related work. Our tasks share a similar objective with Villavicencio (2002), which is to distinguish PP arguments from adjuncts by an informed selection of linguistic fea-

ures. However, we do not use logical forms or explicit for-

grammar in our models, although the use of distributional word representations may capture some syntax. The scale and data collection procedure of our binary classification task (Experiment 1) are more comparable to those of Merlo and Ferrer (2006) or Belinkov et al. (2014), where the authors construct a PP attachment database from Penn TreeBank data. Our binary classification dataset is similar in scale, but is based on VerbNet (Kipper-Schuler, 2005) frames. Experiment 2, which is a smaller-scale experiment on predicting gradient argumenthood judgment data from humans, is a novel task to the extent of our knowledge. The crowdsourcing protocol for collecting human judgments is inspired by Rissman et al. (2015).

Our evaluation setup to measure downstream task perfor-

ance gains from PP argumenthood information (Section 5) is inspired by a recent line of efforts on probing task approaches for evaluating the quality of sentence rep-

resentations (Ribeiro et al., 2018; Ettinger et al., 2016; Gu-

lordava et al., 2018). In order to investigate whether PP ar-

gument/adjunct distinction task has a practical application, we attempt to improve performances on downstream tasks such as SRL with an ultimate goal of making sentence rep-

resentations better and more generalizable using a set of linguistically-motivated tasks (as opposed to extremely fine-
tuned representations for a very specific task). The method is to pretrain a sentence encoder with a linguistic task of in-

rest, fix the encoder weights and then train a classifier for tasks other than the pretraining task, using the representa-

ions from the frozen encoder. This enables us to compare the utility of information from different pretraining tasks (e.g., PP argument/adjunct distinction) on another task (e.g., SRL) or even multiple external tasks.

3 Experiment 1: Binary classification

3.1 Task formulation

Class labels. We use VerbNet subcategorization frames to define the argument-adjunct status of a verb-PP combina-

This means if a certain PP is listed as a subcategorization frame under a certain verb entry, the PP is considered to be an argument of the verb. If it is not listed as a sub-
categorization frame, it is considered an adjunct of the verb. This way of defining argumenthood has been proposed and evaluated by McConville and Dzikovska (2008). Although it should be noted that VerbNet is not an exhaustive list of subcategorization frames and not all frames listed are strictly arguments, McConville and Dzikovska (2008)’s evaluation suggests that VerbNet frame membership is a reasonable proxy of argumenthood. We chose VerbNet over PropBank ARG-N and AM labels, which are also a popular proxy for argumenthood (Abend and Rappoport, 2010), since VerbNet’s design goal of exhaustively listing frames for each verb bet-
ter matches our task that requires broad type-level coverage of V-PP constructions.

Verbs and prepositions. 2714 unique verbs and 60 unique prepositions are used to generate all possible combinations of {VERB, PREP.}. These are all unique verb entries that pertain to a single set of VerbNet class and all prepositions that appear in VerbNet PP frames excluding multi-

word prepositions (e.g., all over, on top of). Some frames are only defined by features such as {+SPATIAL}, with-

out specifying which specific prepositions these features correspond to (see Figure 1). For such featurally-marked frames, a manual mapping was made to specific preposition sets constructed approximately based on PrepWiki sense
annotations (Schneider et al., 2015). As a result, we have \(2714 \times 60 = 162,480\) different \{VERB, PREP.\} tuples that either correspond to \(\rightarrow\text{ARG}\) or do not correspond to \(\rightarrow\text{ADJ}\) a subcategorization frame.

**Dataset and task objective.** Since each verb subcategorizes only a handful of prepositions out of the possible 60, the distribution of labels ARG and ADJ is heavily skewed towards ADJ. The ratio of ARG:ADJ labels in the whole dataset is approximately 1:10. For this reason, we use randomized subsampling of the negative cases to construct a balanced dataset. Since there were 13,544 datapoints with label 1 in the whole set, the same number of label 0 datapoints were randomly subsampled. This balanced dataset \((n = 27,088)\) is randomly split into 70:15:15 train:dev:test sets.

The task is to predict whether a given \{VERB, PREP.\} pair is an argument or an adjunct construction (i.e., whether it is an existing VerbNet subcategorization frame or not). Performance is measured by classification accuracy \((\text{acc.})\) and \(F_1\) on the test set.

**Full-sentence variants of the task.** The meaning of the complement noun phrase (NP) of the preposition (NP under PP) can also be a clue to determining argumenthood. We did not include the NP under PP as a part of the input in our main task dataset because the argumenthood labels we are using are type-level (or frame-level) and not labels for individual instantiation of the frames (token-level). However, the NP meanings, especially thematic roles, do play crucial roles in argumenthood. To address this concern, we propose variants of the main task that give prediction models access to information from the NP by providing full sentence inputs.

We report additional results on one of the full-sentence variants of the main task dataset because the argumenthood labels we are included the NP under PP as a part of the input in our main task dataset because the argumenthood labels we are using are type-level (or frame-level) and not labels for individual instantiation of the frames (token-level). However, the NP meanings, especially thematic roles, do play crucial roles in argumenthood. To address this concern, we propose variants of the main task that give prediction models access to information from the NP by providing full sentence inputs. We report additional results on one of the full-sentence variants (ternary classification) using two of the best-performing model setups for the main task (Table 2).

The full-sentence variants of the main task dataset are constructed by performing a heuristic search through the Stanford Natural Language Inference (SNLI; (Bowman et al., 2015)) and Multi-genre Natural Language Inference (MNLI; (Williams et al., 2018)) datasets using the syntactic parse trees provided, to find sentences that contain a particular \{VERB, PREP.\} construction. Note that the full sentence data is noisier compared to the main dataset (1) because the trees are parser outputs and (2) the original type-level gold labels given to the \{VERB, PREP.\} were unchanged regardless of what the NP may be. Duplicate entries for the same \{VERB, PREP.\} were permitted as long as the sentences themselves were different. In the first task variant, for case where no examples of the input pair is found in the dataset, we retained the original label. In the second variant, we assign a new \(\text{(UNOBSERVED)}\) label to such cases. This helps filter out overgenerated adjunct labels in the original dataset, where the \{VERB, PREP.\} is not listed as a frame because it is an impossible combination. It also eliminates the need for subsampling, since the ratio of ARG:ADJ:UNOBSERVED labels were reasonably balanced.

We chose NLI datasets as the source of full sentence inputs over other parsed corpora such as the Penn TreeBank for the following two reasons. First, we wanted to avoid using the the same source text as several downstream tasks we test in Section 5 (e.g., CoNLL-2005 SRL (Carreras and Márquez, 2005) uses sections of the Penn TreeBank), in order to separate out the benefits of seeing the source text at train time from the benefits of structural knowledge gained from learning to distinguish PP arguments and adjuncts. Second, we wanted both simple, short sentences (SNLI) and complex, naturally-occurring sentences (MNLI) from datasets of a consistent structure.

### Table 1: Test set performance on the binary classification task \((n = 4064)\).

<table>
<thead>
<tr>
<th>Classification model</th>
<th>w2v</th>
<th>GloVe</th>
<th>fastText</th>
<th>ELMo</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM + MLP</td>
<td>94.0</td>
<td>94.0</td>
<td>94.5</td>
<td>94.4</td>
</tr>
<tr>
<td>Concatenation + MLP</td>
<td>92.4</td>
<td>92.3</td>
<td>92.8</td>
<td>92.6</td>
</tr>
<tr>
<td>BoW + MLP</td>
<td>91.4</td>
<td>91.4</td>
<td>91.4</td>
<td>91.3</td>
</tr>
<tr>
<td>Concatenation + Random forest</td>
<td>91.5</td>
<td>91.2</td>
<td>90.8</td>
<td>90.8</td>
</tr>
<tr>
<td>Concatenation + SVM (no hparam tuning)</td>
<td>81.6</td>
<td>81.3</td>
<td>82.3</td>
<td>82.1</td>
</tr>
<tr>
<td>Majority class (== chance)</td>
<td>50.0</td>
<td>50.0</td>
<td>50.0</td>
<td>50.0</td>
</tr>
</tbody>
</table>

2https://code.google.com/archive/p/word2vec/
https://nlp.stanford.edu/projects/glove/
https://github.com/facebookresearch/fastText
https://github.com/allenai/allennlp
We also test models that use the same MLP classifier (Eq. 2) with concatenated input vectors (Concatenation + MLP) or bag-of-words encodings of the input (BoW + MLP).

\[
i_t = \sigma(W_{ix}x_t + b_i + W_{ih}h_{t-1} + b_h) \\
\]

\[
f_t = \sigma(W_{ix}x_t + b_i + W_{if}h_{t-1} + b_f) \\
\]

\[
g_t = \tanh(W_{ix}x_t + b_i + W_{ig}h_{t-1} + b_i) \\
\]

\[
a_t = \sigma(W_{oh}h_t + b_o) \\
\]

\[
c_t = f_t c_{t-1} + i_t g_t \\
\]

\[
h_t = a_t \tanh(c_t) \tag{1}
\]

\[
l = \text{argmax}\{\sigma(W_v[\tanh(W_h[V_{vp}] + b_v)] + b_v)\} \tag{2}
\]

\(V_{vp}\) is a linear projection of the output of Eq. 1 (or the projection of the BoW encoder output or a concatenated word vector; \(d = 512\)) followed by max pooling. \(\sigma\) is the softmax activation function. The output of Eq. 2 is the label (1 or 0) that is more likely given \(V_{vp}\). The models are trained using Adadelta (Zeiler, 2012) with cross-entropy loss and batch size = 32. Several other off-the-shelf implementations from the Python library scikit-learn were also tested in place of MLP, but since MLP consistently outperformed other classifiers, we only list a subset of the models that use non-MLP classifiers out of the models we tested (Table 1).

### 3.3 Results

Table 1 compares the performance of the tested models. The most trivial baseline is chance-level classification, which would yield 50% accuracy and \(F_1\). Since the labels in the dataset are perfectly balanced by randomized subsampling, majority class classification is equivalent to chance-level classification. All nontrivial models outperform chance, and for all models tested ELMo yielded the best performance. Using concatenated inputs that preserve the linear order of the inputs increase both accuracy and \(F_1\) by around 1.5% over bag-of-words, and using a BiLSTM to encode the V+P representation gives another 1.5% improvement across the board.

We additionally report results on the full-sentence, ternary-classification variant of the task (description in Section 3.1) from the best model on the main task. The results are given in Table 2. We did not test the concatenation model since the dimensionality of the vectors would be too large with full sentence inputs. All tested models perform over chance, with ELMo achieving the best performance once again. We observe a similar gain of around 3% by replacing BoW with BiLSTM as in the main task.

### 4 Experiment 2: Gradient argumenthood prediction

As previously discussed in Section 2, there is much work in theoretical linguistics literature that suggests argument-adjunctness is a continuum rather than a dichotomy. We propose a gradient argumenthood prediction task and test regression models that use a combination of embeddings and lexical features. Since there is no publicly available gradient argumenthood data, we collected our own dataset via crowdsourcing\(^3\). Due to the resource-consuming nature of human judgment collection, the size of the dataset is limited (\(n = 305\), 25-way redundant). This task serves as a small-scale, proof-of-concept test for whether a reasonable prediction of gradient argumenthood judgments is possible with informed selection of lexical features (and if so, how well different models perform and what features are informative).

![Distribution of argumenthood scores in our gradient argumenthood dataset.](image)

### 3.3 Results

Table 1 compares the performance of the tested models. The most trivial baseline is chance-level classification, which would yield 50% accuracy and \(F_1\). Since the labels in the dataset are perfectly balanced by randomized subsampling, majority class classification is equivalent to chance-level classification. All nontrivial models outperform chance, and for all models tested ELMo yielded the best performance. Using concatenated inputs that preserve the linear order of the inputs increase both accuracy and \(F_1\) by around 1.5% over bag-of-words, and using a BiLSTM to encode the V+P representation gives another 1.5% improvement across the board.

We additionally report results on the full-sentence, ternary-classification variant of the task (description in Section 3.1) from the best model on the main task. The results are given in Table 2. We did not test the concatenation model since the dimensionality of the vectors would be too large with full sentence inputs. All tested models perform over chance, with ELMo achieving the best performance once again. We observe a similar gain of around 3% by replacing BoW with BiLSTM as in the main task.

### 4 Experiment 2: Gradient argumenthood prediction

As previously discussed in Section 2, there is much work in theoretical linguistics literature that suggests argument-adjunctness is a continuum rather than a dichotomy. We propose a gradient argumenthood prediction task and test regression models that use a combination of embeddings and lexical features. Since there is no publicly available gradient argumenthood data, we collected our own dataset via crowdsourcing\(^3\). Due to the resource-consuming nature of human judgment collection, the size of the dataset is limited (\(n = 305\), 25-way redundant). This task serves as a small-scale, proof-of-concept test for whether a reasonable prediction of gradient argumenthood judgments is possible with informed selection of lexical features (and if so, how well different models perform and what features are informative).

![Distribution of argumenthood scores in our gradient argumenthood dataset.](image)

### 3.3 Results

Table 1 compares the performance of the tested models. The most trivial baseline is chance-level classification, which would yield 50% accuracy and \(F_1\). Since the labels in the dataset are perfectly balanced by randomized subsampling, majority class classification is equivalent to chance-level classification. All nontrivial models outperform chance, and for all models tested ELMo yielded the best performance. Using concatenated inputs that preserve the linear order of the inputs increase both accuracy and \(F_1\) by around 1.5% over bag-of-words, and using a BiLSTM to encode the V+P representation gives another 1.5% improvement across the board.

We additionally report results on the full-sentence, ternary-classification variant of the task (description in Section 3.1) from the best model on the main task. The results are given in Table 2. We did not test the concatenation model since the dimensionality of the vectors would be too large with full sentence inputs. All tested models perform over chance, with ELMo achieving the best performance once again. We observe a similar gain of around 3% by replacing BoW with BiLSTM as in the main task.

### 4 Experiment 2: Gradient argumenthood prediction

As previously discussed in Section 2, there is much work in theoretical linguistics literature that suggests argument-adjunctness is a continuum rather than a dichotomy. We propose a gradient argumenthood prediction task and test regression models that use a combination of embeddings and lexical features. Since there is no publicly available gradient argumenthood data, we collected our own dataset via crowdsourcing\(^3\). Due to the resource-consuming nature of human judgment collection, the size of the dataset is limited (\(n = 305\), 25-way redundant). This task serves as a small-scale, proof-of-concept test for whether a reasonable prediction of gradient argumenthood judgments is possible with informed selection of lexical features (and if so, how well different models perform and what features are informative).

![Distribution of argumenthood scores in our gradient argumenthood dataset.](image)

### 4.1 Data

The gradient argumenthood judgment dataset consists of 305 sentences that contain a single main verb and a PP dependent of that verb. All sentences are adaptations from example sentences in VerbNet PP subcategorization frames

---

\(^3\)See Supplemental Material for examples of questions given to participants. Detailed protocol and theoretical analysis of the data are omitted; it will be discussed in a separate theoretically-oriented paper in preparation.

\(^4\)https://github.com/idio/wiki2vec
argument-like compared to hid in a hurry clamped on his ankle. However, it is still less argument-like than a more clearly argumenthood. For instance, the absolute values of the scores, but how the scores compare to home shuttle from home in argumenthood. For instance, the absolute values of the scores, but how the scores compare to home shuttle from home in argumenthood.

The actual numbers range between \([-1, 1]\), roughly centered around zero \((\mu = 1.967e-11, \sigma = 0.526)\). The distribution is slightly left-skewed (right-leaning), with more positive values than negative values \((z = 2.877, p < .01\); see Figure 2). Here are some examples with varying degrees of argumenthood, indicated by the numbers:

<table>
<thead>
<tr>
<th>Example</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>I whipped the sugar</em> with cream.</td>
<td>0.35</td>
</tr>
<tr>
<td><em>The witch turned him</em> into a frog.</td>
<td>0.57</td>
</tr>
<tr>
<td><em>The children hid</em> in a hurry.</td>
<td>-0.41</td>
</tr>
<tr>
<td><em>It clamped</em> on his ankle.</td>
<td>0.66</td>
</tr>
<tr>
<td><em>Amanda shuttled</em> the children from home.</td>
<td>-0.1</td>
</tr>
</tbody>
</table>

We refrain from assigning definitive interpretations to the absolute values of the scores, but how the scores compare to each other gives us insight into the relative difference in argumenthood. For instance, shuttled from home with a score of \(-0.1\) is more adjunct-like than a higher-scoring construction such as clamped on his ankle \((0.66)\), but more argument-like compared to hid in a hurry with a score of \(-0.41\). This matches the intuition that a locative PP from home would be more argument-like to a change-of-location predicate shuttle, compared to a manner PP like in a hurry. However, it is still less argument-like than a more clearly argument-like PP on his ankle that is selected by clamp.

### 4.2 Model

**Features.** We use the same sets of word embeddings we used in Experiment 1 as base features, but we reduced their dimensionality to 5 via Principal Component Analysis. This reduction step is necessary due to the large dimensionality \((d \geq 300)\) of the word vectors compared to the small size of our dataset \((n = 305)\). Various features in addition to the embeddings of verbs and prepositions were also tested. The features we experimented with include semantic proto-role property scores (Reisinger et al., 2015) of the target PP (normalized mean across 5 annotators), mutual information (MI) (Aldezabal et al., 2002), word embeddings of the nominal head token of the NP under the PP in question, existence of a direct object, and various interaction terms between the features (e.g., additive, subtractive, inner/outer products).

The following lexical features were selected for the final models based on dev set performance: embeddings of the verb, preposition, nominal head, mutual information and existence of a direct object (D.O.). The intuition behind including D.O. is that if there exists a direct object in the given sentence, the syntactically oblique PP dependent would seem comparatively less argument-like compared to the direct object. This feature is expected to reduce the noise introduced by different argument structures of the main verbs.

We also include a diagnostics feature which is a weighted combination of two different traditional diagnostic test results (omissibility and pseudo-cleftability) produced by a linguist with expertise in theoretical syntax. Unlike all other features in our feature set, this diagnostics feature is not straightforwardly computable from corpus data. We add this feature in order to examine how powerful traditional linguistic diagnostics are in capturing gradient argumenthood.

**Regression model.** The selected features are given as inputs to an MLP regressor which is equivalent to Eq. 2 in Experiment 1 except that it outputs a continuous value. The regressor consists of an input layer with \(n\) features, \(m\) hidden units and a single output unit. Smooth L1 loss is used in order to reduce sensitivity to out-

<table>
<thead>
<tr>
<th>Model</th>
<th>Pearson’s (r)</th>
<th>(R^2)</th>
<th>(R_{adj}^2)</th>
<th>Pearson’s (r)</th>
<th>(R^2)</th>
<th>(R_{adj}^2)</th>
<th>Pearson’s (r)</th>
<th>(R^2)</th>
<th>(R_{adj}^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple MLP</td>
<td>0.554</td>
<td>0.255</td>
<td>0.231</td>
<td>0.568</td>
<td>0.268</td>
<td>0.245</td>
<td>0.582</td>
<td>0.281</td>
<td>0.257</td>
</tr>
<tr>
<td>Linear</td>
<td>0.579</td>
<td>0.280</td>
<td>0.257</td>
<td>0.560</td>
<td>0.243</td>
<td>0.219</td>
<td>0.591</td>
<td>0.291</td>
<td>0.268</td>
</tr>
<tr>
<td>SVM</td>
<td>0.561</td>
<td>0.243</td>
<td>0.218</td>
<td>0.463</td>
<td>0.110</td>
<td>0.082</td>
<td>0.580</td>
<td>0.267</td>
<td>0.243</td>
</tr>
</tbody>
</table>

Table 3: 10-fold cross-validation results on the gradient argumenthood prediction task.
we limit ourselves to a simpler MLP-only model for this experiment; the BiLSTM encoder model suffered from overfitting.

**Evaluation metrics.** We use 15% of the dataset as development set, and train/test using 10-fold cross-validation on the remaining 85% rather than reporting performance on a fixed test split. This is because the credibility of performance on one test split may be questioned due to the small sample size. Pearson’s r averaged over the 10 folds using Fisher z-transformation is the main metric. Mean $R^2$ and Adjusted $R^2$ ($R^2_{adj}$) are also reported to account for the potentially differing number of predictors in the ablation experiment.

### 4.3 Results and discussion

Table 3 reports performances on the regression task. Results from several off-the-shelf regressors are reported for comparison. ELMo embeddings again produced the best results among the embeddings used in Experiment 1. We speculate that the dimensionality of the embeddings may have impacted the results, and ran additional experiments using higher-dimensional embeddings. Higher-dimensional embeddings did indeed lead to performance improvements, even though the actual inputs given to the models were all PCA-reduced to $d = 5$. From this observation, we could further improve upon the initial ELMo results. Results from the best model (w2v-wiki) are given in addition to the set of results using the same embeddings as the models in Experiment 1. This model uses 1000-$d$ word2vec features with additional interaction features (multiplicative, subtractive) that improved dev set performance.

**Ablation.** We conducted ablation experiments with the two best-performing models to examine the contribution of non-embedding features discussed in Section 4.2. Table 4 indicates that any linguistic feature contributes positively towards performance, with the direct object feature helping both word2vec and ELMo models the most. This supports our initial hypothesis that adding the direct object feature would help reduce noise in the data. When only the linguistic features are used without embeddings as base features, mutual information is the most informative. This suggests that there is some (but not complete) redundancy in information captured by word embeddings and mutual information. The diagnostics features is informative but is a comparatively weak predictor, which aligns with the current state of diagnostic acceptability tests—they are sometimes useful but not always, especially with respect to syntactically oblique items such as PPs. This behavior of the diagnostics predictor adds credibility to our data collection protocol.

### 5 Why is this a useful standalone task?

In motivating our tasks, we suggested that PP argumenthood information could improve downstream task performance such as SRL and parsing. We investigate whether this is a grounded claim by testing two separate hypotheses: (1) whether the task is indeed useful, and if so, (2) whether it is worthy as a standalone task. We leave the issue of gradient argumenthood to future work for now, since the dataset is currently small and the notion of gradient argumenthood is not yet compatible with formulations of many NLP tasks.

#### 5.1 Improving representations with pretraining

We first test the utility of the binary argumenthood task in improving performances on existing NLP tasks. We selected three tasks that may benefit from PP argumenthood information: SRL on labeled Wall Street Journal data (WSJ) (CoNLL 2005 shared task; Carreras and Marquez 2005), SRL on OntoNotes Corpus (CoNLL 2012 data; Pradhan et al. 2012), and PP attachment disambiguation on WSJ data (Belinkov et al., 2014).

Our test setup is as follows. We first train a BiLSTM sentence encoder with an objective to maximize dev set performance on our PP argumenthood tasks. Then we freeze the encoder weights and train an MLP classifier to perform new tasks only using the representations produced by the frozen-weight sentence encoder. If learning to make correct PP argumenthood distinction teaches models knowledge that is generalizable to the new tasks, the classifier trained on top of the fixed-weights encoder will perform better on those tasks compared to a classifier trained on top of an encoder with randomly initialized weights. Improvements over the

<table>
<thead>
<tr>
<th></th>
<th>w2v-wiki</th>
<th>ELMo</th>
<th>No embeddings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Pearson’s $r$</td>
<td>$R^2$</td>
<td>$R^2_{adj}$</td>
</tr>
<tr>
<td>Embeddings</td>
<td>0.430</td>
<td>0.064</td>
<td>0.046</td>
</tr>
<tr>
<td>+ML</td>
<td>0.464</td>
<td>0.158</td>
<td>0.138</td>
</tr>
<tr>
<td>+D.O.</td>
<td>0.575</td>
<td>0.245</td>
<td>0.227</td>
</tr>
<tr>
<td>+diag.</td>
<td>0.449</td>
<td>0.125</td>
<td>0.104</td>
</tr>
<tr>
<td>+ML. +D.O.</td>
<td>0.586</td>
<td>0.278</td>
<td>0.258</td>
</tr>
<tr>
<td>+ML. +diag.</td>
<td>0.515</td>
<td>0.205</td>
<td>0.182</td>
</tr>
<tr>
<td>+diag. +D.O.</td>
<td>0.572</td>
<td>0.265</td>
<td>0.245</td>
</tr>
<tr>
<td>+all</td>
<td>0.624</td>
<td>0.330</td>
<td>0.309</td>
</tr>
</tbody>
</table>

Table 4: Ablation results from the two best models and a non-embedding features only-model (10-fold cross validation).
randomly initialized setup from pretraining on our main PP argumenthood task (Arg) and its full-sentence variants (Arg fullsent and Arg fullsent 3-way; see Section 3.1 for details) are shown in Table 5. Only statistically significant ($p < .05$) improvements over the random encoder model are bolded, with significance levels calculated via Approximate Randomization (Yeh, 2000) ($R = 1000$). The models trained on PP argumenthood tasks perform significantly better than the random initialization model in both SRL tasks, which supports our initial claim that argumenthood tasks can be useful for SRL. However, we did not observe significant improvements for the PP attachment disambiguation task. We speculate that since the task as formulated in Belinkov et al. (2014) requires the model to understand PP dependents of NPs as well as VPs, our tasks that focus on verbal dependents may not provide the full set of linguistic knowledge necessary to solve this task. Nevertheless, our models are not significantly worse than the baseline, and the accuracy of the Arg fullsent model (88.2%) was comparable to a model that uses an encoder directly trained on PP attachment (88.7%).

Secondly, we discuss whether it is indeed useful to formulate PP argumenthood prediction as a separate task. The questions that need to be answered are (1) whether it would be the same or better to use a different pretraining task that would provide similar information (e.g., PP attachment disambiguation), and (2) whether the performance gain can be attributed to simply seeing more datapoints at train time rather than to the regularities we hope the models would learn through our task. Table 6 addresses both questions; we compare models pretrained on argumenthood tasks to a model pretrained directly on the PP attachment task listed in Table 5. All models trained on PP argumenthood prediction outperform the model trained on PP attachment, despite the fact that the latter has unfair advantage for SRL2005 since the tasks share the same source text (WSJ). Furthermore, the variance in the sizes of the datasets indicates that the reported performance gains cannot solely be due to the increased number of datapoints seen during training.

### Table 5: Performance gains over random initialization from pretraining sentence encoders on PP argumenthood task variants. ($** : p < .001$)

<table>
<thead>
<tr>
<th>Test tasks</th>
<th>metric</th>
<th>Random</th>
<th>Arg</th>
<th>Arg fullsent</th>
<th>Arg fullsent 3-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRL-CoNLL2005 (WSJ)</td>
<td>$F_1$</td>
<td>81.7</td>
<td><strong>83.9</strong>*</td>
<td><strong>84.7</strong>*</td>
<td><strong>84.5</strong>*</td>
</tr>
<tr>
<td>SRL-CoNLL2012 (OntoNotes)</td>
<td>$F_1$</td>
<td>77.3</td>
<td><strong>80.2</strong>*</td>
<td><strong>80.4</strong>*</td>
<td><strong>80.7</strong>*</td>
</tr>
<tr>
<td>PP attachment (Belinkov et al., 2014)</td>
<td>acc.</td>
<td>87.5</td>
<td>87.6</td>
<td>88.2</td>
<td>87.0</td>
</tr>
</tbody>
</table>

### Table 6: Comparison against using PP attachment directly as a pretraining task ($*** : p < .001$).

<table>
<thead>
<tr>
<th></th>
<th>Size</th>
<th>PP att.</th>
<th>Arg</th>
<th>Arg full</th>
<th>Arg full 3-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRL2005</td>
<td>32k</td>
<td>80.2</td>
<td><strong>83.9</strong>*</td>
<td><strong>84.7</strong>*</td>
<td><strong>84.5</strong>*</td>
</tr>
<tr>
<td>SRL2012</td>
<td>19k</td>
<td>79.8</td>
<td><strong>80.2</strong>*</td>
<td><strong>80.3</strong>*</td>
<td><strong>80.7</strong>*</td>
</tr>
</tbody>
</table>

## 6 Conclusion

We have proposed two different tasks—binary and gradient—for predicting PP argumenthood, and reported results on each using four different types of word embeddings as base predictors. We obtain 95.5 accuracy and 95.4 $F_1$ in the binary classification task with BiLSTM and ELMo, and $r = 0.624$ for the gradient human argumenthood judgment prediction task. Our overall contribution is threefold: first, we have demonstrated that a principled prediction of both binary and gradient argumenthood judgments is possible with informed selection of lexical features; second, we justified the utility of our binary PP argumenthood classification as a standalone task by reporting performance gains on multiple SRL tasks through encoder pretraining. Finally, we have conducted a small-scale, proof-of-concept study with a novel gradient argumenthood prediction task, paired with a new dataset that we plan to release.

### 6.1 Future work

The pretraining approach holds much promise in understanding and improving neural network models of language. Especially for end-to-end models, this method has substantial advantage over architecture engineering or hyperparameter tuning in terms of interpretability. That is, we can attribute the source of the performance gain on end tasks to the knowledge necessary to do well on the pretraining task. For instance, in Section 5 we can infer that that knowing how to make correct PP argumenthood distinction helps models encode representations that are more useful for SRL. Furthermore, we believe it is important to contribute to the recent efforts for designing better probing tasks to understand what machines really know about natural language (as opposed to taking downstream performances as metrics of ‘better’ models). We hope to scale up our preliminary experiments and will continue to work on developing a set of linguistically informed probing and pretraining tasks for higher-quality, better-generalizable sentence representations.

### References


