Predicting the Argumenthood of English Prepositional Phrases

Najoung Kim,[†] **Kyle Rawlins,**[†] **Benjamin Van Durme,**[†,∂] **Paul Smolensky**[†,△]

[†]Department of Cognitive Science, Johns Hopkins University, Baltimore, MD, USA
[∂]Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA
[△]Microsoft Research AI, Redmond, WA, USA
{n.kim,kgr,vandurme,smolensky}@jhu.edu

Abstract

Distinguishing between arguments and adjuncts of a verb is a longstanding, nontrivial problem. In natural language processing, argumenthood information is important in tasks such as semantic role labeling (SRL) and prepositional 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) $acc. = 0.955$, $F_1 = 0.954$ (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 *arguments* and *adjuncts* of a verb. For example, in the following example, *the window* is an argument of the verb *open*, whereas *this morning* and *with Mary* are adjuncts.

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

What distinguishes arguments (or *complements*) from adjuncts (or *modifiers*)[1], 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

[1]Various combinations of the terminology are found in the literature, with subtle domain-specific preferences. We use *arguments* and *adjuncts*, with a rough definition of *arguments* as elements specifically selected or subcategorized by the verb. We use the umbrella term *(verbal) dependents* to refer to both.

parsing (Carroll, Minnen, and Briscoe 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. Prop-Bank (Palmer, Gildea, and Kingsbury 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 to a systematic approach for predicting argumenthood.

We focus on PPs 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 better PP argumenthood prediction is indeed useful for NLP. Through a controlled evaluation setup, we demonstrate that pretraining 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 verbal 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 (Rissman, Rawlins, and Landau 2015)). This raises many interesting theoretical

questions such as what kinds of lexical and contextual information affect these judgments, and whether the judgments would be predictable in a principled way given that information. By building prediction models for gradient judgments using lexical features, we hope to gain insights about what factors explain gradience and to what degree they do so.

Utility in NLP   Automatic parsing will likely benefit from argumenthood information. For instance, the issue of PP attachment negatively affects the accuracy of a competitive dependency parser (Dasigi et al. 2017). It has been shown that reducing PP attachment errors leads to higher parsing accuracy (Agirre, Baldwin, and Martinez 2008; Belinkov et al. 2014), and also that argument-adjunct distinction is useful for PP attachment disambiguation (Merlo and Ferrer 2006).

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 attachment errors. They also observe that errors in widely-used automatic parsers pose challenges to improving performance in syntax-aware neural models (Marcheggiani and Titov 2017). This suggests that improving parsers with better argumenthood distinction would lead to better SRL performance.

Przepiórkowski and Patejuk (2018) discuss the issue of core-noncore distinction being confounded with argument-adjunct distinction in the annotation protocol of Universal Dependencies (UD), which leads to internal inconsistencies. They point out the flaws of the argument-adjunct distinction and suggest a solution that disentangles it better from the core-noncore distinction that UD advocates. This does improve within-UD consistency, but argument-adjunct status is still explicitly encoded in many formal grammars including Combinatory Categorial Grammar (CCG) (Steedman 2000). Thus, being able to predict argumenthood would still be important in improving the quality of resources being ported between different grammars, such as CCGBank.

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 features. However, we do not use logical forms or explicit formal grammar in our models, although the use of distributional word representations may capture some syntactic information. 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 classfication 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, Rawlins, and Landau (2015).

Our evaluation setup to measure downstream task performance gains from PP argumenthood information (Section 5)

is inspired by a recent line of efforts on *probing* for evaluating linguistic representations encoded by neural networks (Gulordava et al. 2018; Ettinger et al. 2018). In order to investigate whether PP argument-adjunct distinction tasks have a practical application, we attempt to improve performances on existing tasks such as SRL with an ultimate goal of making sentence representations better and more generalizable (as opposed to representations optimized for one specific task). We use the setup of *pre*training a sentence encoder with a linguistic task of interest, fixing the encoder weights and then training a classifier for tasks other than the pretraining task, using the representations from the frozen encoder (Bowman et al. 2018). 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   Exp. 1: Binary Classification
3.1   Task Formulation

Class Labels   We use VerbNet subcategorization frames to define the argument-adjunct status of a verb-PP combination. 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 subcategorization frame, it is considered an adjunct of the verb. This way of defining argumenthood has been proposed and studied by McConville and Dzikovska (2008). Their evaluation suggests that even though VerbNet is not an exhaustive list of subcategorization frames and not all frames listed are strictly arguments, VerbNet membership is a reasonable proxy of PP argumenthood. We chose VerbNet over PropBank ARG-N and AM labels, which are also a viable proxy for argumenthood (Abend and Rappoport 2010), since VerbNet's design goal of exhaustively listing frames for each verb better matches our task that requires broad type-level coverage of V-PP constructions.

Figure 1: Examples of PP.X frames in VerbNet.

Verbs and Prepositions   2714 unique verbs (V) and 60 unique prepositions (P) are used to generate all possible combinations of $\{V, P\}$. 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 $\{+\text{SPATIAL}\}$, without specifying which specific prepositions these features correspond to (see Figure 1). For such featurally-marked frames, a manual mapping was made to preposition sets constructed ap-

	w2v		Glove		fastText		ELMo	
Classification model	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
BiLSTM + MLP	94.0	94.0	94.5	94.4	94.6	94.6	**95.5**	**95.4**
Concatenation + MLP	92.4	92.5	93.3	93.3	93.6	93.5	94.4	94.4
BoW + MLP	91.9	91.9	92.4	92.4	92.7	92.5	93.7	93.7
Majority class (== chance)	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0

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

proximately based on PrepWiki sense annotations (Schneider et al. 2015). As a result, we have $2714 * 60 = 162,480$ different {V, P} tuples that either correspond to ($\mapsto$ARG) or do not correspond to ($\mapsto$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 {V, P} 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 and F_1 on the test set.

Full-sentence Tasks   The meaning of the complement noun phrase (NP) of the preposition can also be a clue to determining argumenthood. We did not include the NP as a part of the input in our main task dataset because the argumenthood labels we are using are type-level and not labels for individual instantiation of the types (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, Nangia, and Bowman 2018)) datasets using the syntactic parse trees provided, to find sentences that contain a particular {V, P} 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 {V, P} were unchanged regardless of what the NP may be. Duplicate entries for the same {V, P} 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 UNOBSERVED label to such cases. This helps filter out overgenerated adjunct labels in the original dataset, where the {V, P} is not listed as a frame because it is an impossible or an infelicitous construction. It also eliminates the need for subsampling, since the three 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 Màrquez 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-occuring sentences (MNLI) from datasets of a consistent structure.

3.2   Model

Input Representation   We report results using 4 different types of word embeddings (word2vec (Mikolov et al. 2013), GloVe (Pennington, Socher, and Manning 2014), fastText (Bojanowski et al. 2016), ELMo (Peters et al. 2018)) to represent the input tuples. Publicly available pretrained embeddings provided by the respective authors[2] are used.

Classifiers   Our current best model uses a combination of bidirectional LSTM (BiLSTM) and multi-layer perceptron (MLP) for the binary classification task. We first obtain a representation of the given {V, P} sequence using an encoder and then train an MLP classifier (Eq. 1) on top to perform the actual classification. The BiLSTM encoder is implemented using the AllenNLP toolkit (Gardner et al. 2018), and the MLP classifier is a simple feedforward neural network with a single hidden layer that consists of 512 units (Eq. 1). We also test models that use the same MLP classifier with concatenated input vectors (Concatenation + MLP) or bag-of-words encodings of the input (BoW + MLP).

$$l = argmax(\sigma(W_o[\tanh(W_h[V_{vp}] + b_h)] + b_o)) \quad (1)$$

V_{vp} is a linear projection ($d = 512$) of the output of an encoder (encoder states followed by max-pooling or a concate-

---

[2] code.google.com/archive/p/word2vec/
nlp.stanford.edu/projects/glove/
github.com/facebookresearch/fastText
github.com/allenai/allennlp

	w2v		Glove		fastText		ELMo	
Classification model	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
BiLSTM + MLP	95.6	94.3	97.0	96.1	96.9	96.0	**97.4**	**96.6**
BoW + MLP	93.8	92.0	93.7	91.8	94.2	92.5	95.9	94.7
Majority class	38.8	38.8	38.8	38.8	38.8	38.8	38.8	38.8
Chance	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3

Table 2: Model performances on **full-sentence, unobserved label included** variant of the original classification task (now ternary classification) on the test set ($n = 18,764$).

nated word vector), and σ is the softmax activation function. The output of Eq. 1 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 models that use 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 out of all models tested ELMo yielded the best performance. Using concatenated inputs that preserve the linear order of the inputs increases both accuracy and F_1 by around 1.5%p over bag-of-words, and using a BiLSTM to encode the V+P representation adds 1.5%p 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%p by replacing BoW with BiLSTM as in the main task.

4 Exp. 2: Gradient Argumenthood Prediction

As discussed in Section 2, there is much work in theoretical linguistics literature that suggests argument-adjuncthood 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).
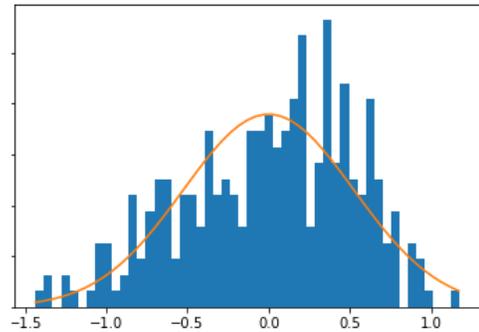

Figure 2: Distribution of argumenthood scores in our gradient argumenthood dataset.

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 or sentences containing ARG-N PPs in PropBank. To capture a full range of the argumenthood spectrum from fully argument-like to fully adjunct-like, we manually augmented the dataset by adding more strongly adjunct-like examples. These examples are generated by substituting the PP of the original sentence with a felicitous adjunct. This step is necessary since PPs listed as a subcategorization frame or ARG-N are more argument-like, as discussed in Section 3.1.

25 participants were recruited to produce argumenthood judgments about these sentences on a 7-point Likert scale, using protocols adapted from a prior work for collecting similar judgments (Rissman, Rawlins, and Landau 2015). Larger numbers are given an interpretation of being more argument-like and smaller numbers, more adjunct-like. The

all $d = 300$	w2v-googlenews			GloVe			fastText		
Model	Pearson's r	R^2	R^2_{adj}	Pearson's r	R^2	R^2_{adj}	Pearson's r	R^2	R^2_{adj}
Simple MLP	0.554	0.255	0.231	0.568	0.268	0.245	0.582	0.281	0.257
Linear	0.579	0.280	0.257	0.560	0.243	0.219	0.591	0.291	0.268
SVM	0.561	0.243	0.218	0.463	0.110	0.082	0.580	0.267	0.243

$d \geq 1000$	w2v-wiki[4] ($d = 1000$)			ELMo ($d = 1024$)		
Model	Pearson's r	R^2	R^2_{adj}	Pearson's r	R^2	R^2_{adj}
Simple MLP	**0.624**	**0.330**	**0.309**	0.609	0.304	0.281
Linear	0.609	0.311	0.289	0.586	0.293	0.270
SVM	0.549	0.237	0.213	0.337	0.052	0.022

Table 3: 10-fold cross-validation results on the gradient argumenthood prediction task.

results were z-normalized within-subject to adjust for individual differences in the use of the scale, and then the final argumenthood scores were computed by averaging the normalized scores across all participants. The result is a set of values on a continuum, each value representing each sentence in the dataset. The actual numbers range between $[-1.435, 1.172]$, roughly centered around zero ($\mu = 1.967e\text{-}11, \sigma = 0.526$). 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:

*I **whipped** the sugar [with cream].* (0.35)
*The witch **turned** him [into a frog].* (0.57)
*The children **hid** [in a hurry].* (−0.41)
*It **clamped** [on his ankle].* (0.66)
*Amanda **shuttled** the children [from home].* (−0.1)

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 which is equivalent to Eq. 1 in Experiment 1 except that it outputs a continuous value. This regressor consists of an input layer with n units (corresponding to n features), m hidden units and a single output unit. Smooth L1 loss is used in order to reduce sensitivity to outliers, and m, the activation function (ReLU, Tanh or Sigmoid), optimizers and learning rates are all tuned using the development set for each individual model. 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

Model	w2v-wiki			ELMo			No embeddings		
	Pearson's r	R^2	R^2_{adj}	Pearson's r	R^2	R^2_{adj}	Pearson's r	R^2	R^2_{adj}
Embeddings	0.430	0.064	0.046	0.404	0.063	0.044	-	-	-
+MI	0.464	0.158	0.138	0.458	0.153	0.133	0.376	0.083	0.079
+D.O.	0.575	0.245	0.227	0.530	0.202	0.183	0.301	0.029	0.025
+diag.	0.449	0.125	0.104	0.440	0.138	0.117	0.268	0.027	0.023
+MI. +D.O.	0.586	0.278	0.258	0.607	0.297	0.277	0.466	0.165	0.158
+MI. +diag.	0.515	0.205	0.182	0.512	0.193	0.170	0.436	0.141	0.134
+diag. +D.O.	0.572	0.265	0.245	0.535	0.224	0.203	0.392	0.114	0.107
+all	0.624	0.330	0.309	0.609	0.304	0.281	0.516	0.215	0.206

Table 4: Ablation results from the two best models and a non-embedding features only-model (10-fold cross validation).

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 (although our ablation study reveals that using only word embeddings as predictors is not sufficient to obtain satisfactory results). We further speculated that the dimensionality of the embeddings may have impacted the results, and ran additional experiments using higher-dimensional embeddings that were publicly available. 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 inital 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 feature 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 existing NLP 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 useful 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 Wall Street Journal (WSJ) data (CoNLL 2005; Carreras and Màrquez 2005), SRL on OntoNotes Corpus (CoNLL 2012 data; Pradhan et al. 2012)[5], and PP attachment disambiguation on WSJ (Belinkov et al. 2014).

We follow Bowman et al. (2018)'s setup to pretrain and evaluate sentence encoders[6]. 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 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 initalization model in both SRL tasks, which supports our initial claim that argumenthood tasks can be useful for SRL. Although not all errors made by the models were interpretable, we found interesting improvements such as the model trained on the PP argumenthood task being slightly more accurate than the random initialization model on AM-DIR, AM-LOC, and AM-MNR labels.

---

[5]Tasks are labeling only, as described in Tenney et al. (2019).
[6]github.com/jsalt18-sentence-repl/jiant

Test tasks	metric	Sentence encoder pretraining tasks			
		Random	Arg	Arg fullsent	Arg fullsent 3-way
SRL-CoNLL2005 (WSJ)	F_1	81.7	**83.9*****	**84.7*****	**84.5*****
SRL-CoNLL2012 (OntoNotes)	F_1	77.3	**80.2*****	**80.4*****	**80.7*****
PP attachment (Belinkov et al. 2014)	*acc.*	87.5	87.6	88.2	87.0

Table 5: Gains over random initialization from pretraining sentence encoders on PP argumenthood tasks. (*** : $p < .001$)

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 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.

	PP att.	Arg	Arg full	Arg full 3-way
Size	32k	19k	58k	87k
SRL2005	80.2	**83.9*****	**84.7*****	**84.5*****
SRL2012	79.8	**80.2*****	**80.3*****	**80.7*****

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

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 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 se-

lection of lexical features; second, we justified the utility of our binary PP argumenthood classification as a standalone task by reporting performance gains on multiple end-tasks through encoder pretraining. Finally, we have conducted a proof-of-concept study with a novel gradient argumenthood prediction task, paired with a new public dataset[7].

6.1 Future Work

The pretraining approach holds promise in understanding and improving neural network models of language. Especially for end-to-end models, this method has an 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 directly 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

Abend, O., and Rappoport, A. 2010. Fully unsupervised core-adjunct argument classification. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*, 226–236.

Agirre, E.; Baldwin, T.; and Martinez, D. 2008. Improving parsing and PP attachment performance with sense information. In *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics*, 317–325.

Aldezabal, I.; Aranzabe, M.; Gojenola, K.; Sarasola, K.; and Atutxa, A. 2002. Learning argument/adjunct distinction for basque. In *Proc. of the ACL-02 Workshop on Unsupervised lexical acquisition, Vol. 9*, 42–50.

[7]To be released at: decomp.io

Belinkov, Y.; Lei, T.; Barzilay, R.; and Globerson, A. 2014. Exploring compositional architectures and word vector representations for prepositional phrase attachment. *Transactions of the Association for Computational Linguistics* 2:561–572.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv:1607.04606*.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*, 632–642.

Bowman, S. R.; Pavlick, E.; Grave, E.; Durme, B. V.; Wang, A.; Hula, J.; Xia, P.; Pappagari, R.; McCoy, R. T.; Patel, R.; Kim, N.; Tenney, I.; Huang, Y.; Yu, K.; Jin, S.; and Chen, B. 2018. Looking for elmo's friends: Sentence-level pretraining beyond language modeling. *arXiv:1812.10860*.

Carreras, X., and Màrquez, L. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proc. of the 9th Conference on Computational Natural Language Learning*, 152–164.

Carroll, J.; Minnen, G.; and Briscoe, T. 1998. Can subcategorisation probabilities help a statistical parser? In *Proc. of the 6th ACL/SIGDAT Workshop in Very Large Corpora*. http://www.aclweb.org/anthology/W98-1114.

Chomsky, N. 1993. *Lectures on government and binding: The Pisa lectures*. Walter de Gruyter.

Dasigi, P.; Ammar, W.; Dyer, C.; and Hovy, E. 2017. Ontology-aware token embeddings for prepositional phrase attachment. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Vol 1: Long Papers)*, 2089–2098.

Ettinger, A.; Elgohary, A.; Phillips, C.; and Resnik, P. 2018. Assessing composition in sentence vector representations. In *Proc. of the 27th International Conference on Computational Linguistics*, 1790–1801.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N. F.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 1–6.

Gulordava, K.; Bojanowski, P.; Grave, E.; Linzen, T.; and Baroni, M. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, 1195–1205.

He, L.; Lee, K.; Lewis, M.; and Zettlemoyer, L. 2017. Deep semantic role labeling: What works and what's next. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers)*, 473–483.

Hockenmaier, J., and Steedman, M. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proc. of the 3rd International Conference on Language Resources and Evaluation*.

Kipper-Schuler, K. 2005. Verbnet: A broad-coverage, comprehensive verb lexicon. *Ph. D. diss., University of Pennsylvania*.

Marcheggiani, D., and Titov, I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1506–1515.

McConville, M., and Dzikovska, M. O. 2008. Evaluating complement-modifier distinctions in a semantically annotated corpus. In *Proc. of the International Conference on Language Resources and Evaluation*.

Merlo, P., and Ferrer, E. E. 2006. The notion of argument in prepositional phrase attachment. *Computational Linguistics* 32(3):341–378.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics* 31(1):71–106.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (Vol. 1: Long Papers)*, 2227–2237.

Pollard, C., and Sag, I. 1987. *Information-Based Syntax and Semantics, Vol. 1*. CSLI Publications.

Pradhan, S.; Moschitti, A.; Xue, N.; Uryupina, O.; and Zhang, Y. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, 1–40.

Przepiórkowski, A., and Patejuk, A. 2018. Arguments and adjuncts in universal dependencies. In *Proc. of the 27th International Conference on Computational Linguistics*, 3837–3852.

Reisinger, D.; Rudinger, R.; Ferraro, F.; Harman, C.; Rawlins, K.; and Van Durme, B. 2015. Semantic proto-roles. *Transactions of the Association for Computational Linguistics* 3:475–488.

Rissman, L.; Rawlins, K.; and Landau, B. 2015. Using instruments to understand argument structure: Evidence for gradient representation. *Cognition* 142:266–290.

Schneider, N.; Srikumar, V.; Hwang, J. D.; and Palmer, M. 2015. A hierarchy with, of, and for preposition supersenses. In *Proc. of The 9th Linguistic Annotation Workshop*, 112–123.

Schütze, C. T. 1995. PP attachment and argumenthood. *MIT working papers in linguistics* 26(95):151.

Steedman, M. 2000. *The syntactic process*, volume 24. MIT Press.

Tenney, I.; Xia, P.; Chen, B.; Wang, A.; Poliak, A.; McCoy, R. T.; Kim, N.; Durme, B. V.; Bowman, S.; Das, D.; and Pavlick, E. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *ICLR*.

Tutunjian, D., and Boland, J. E. 2008. Do we need a distinction between arguments and adjuncts? evidence from psycholinguistic studies of comprehension. *Language and Linguistics Compass* 2(4):631–646.

Villavicencio, A. 2002. Learning to distinguish pp arguments from adjuncts. In *Proc. of the 6th Conference on Natural language learning, Vol. 20*, 1–7. Association for Computational Linguistics.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (Vol. 1: Long Papers)*, 1112–1122.

Yeh, A. 2000. More accurate tests for the statistical significance of result differences. In *Proc. of the 18th Conference on Computational linguistics, Vol. 2*, 947–953.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv:1212.5701*.