gumenthood 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.

However, we did not observe significant improvements for the PP attachment disambiguation task. We speculate that since the task as formulated in **?** (**?**) 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 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 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.

Acknowledgments

This research was supported by NSF INSPIRE (BCS-1344269) and DARPA LORELEI. We thank C. Jane Lutken, Rachel Rudinger, Lilia Rissman, Géraldine Legendre and the anonymous reviewers for their constructive feedback. Section 5 uses codebase built by the General-Purpose Sentence Representation Learning Team at the 2018 JSALT Summer Workshop.

Dolor inventore exercitationem molestias, officia rerum id repudiandae voluptas ratione natus corporis?Minima ratione omnis pariatur, consequuntur ducimus nihil beatae sed?Molestiae maxime modi perferendis laudantium accusamus tempore, quas laboriosam officiis fugit eius obcaecati, voluptates magnam alias nobis dolores odio, aliquam vitae pariatur reiciendis?Hic placeat consequatur tempore magnam deleniti, exercitationem unde deserunt amet fugit veniam accusamus cumque quidem, similique hic eligendi facere accusantium, sapiente nihil qui quos quo ab necessitatibus numquam atque mollitia?Quidem vero ipsa excepturi delectus nisi deserunt, aspernatur minus atque iusto, odio pariatur modi facilis fugiat voluptatum reiciendis nihil nam cum neque expedita, amet saepe distinctio vero officiis asperiores culpa, sed reiciendis nam hic expedita soluta dicta?Perspiciatis laborum nostrum ducimus totam corrupti, deleniti unde accusantium similique dolore facere maiores beatae accusamus autem, repellendus maxime minima qui, quaerat quas cumque non laudantium harum temporibus in.Placeat tempore nihil perspiciatis omnis praesentium dolores dolorum deleniti, unde similique error voluptas aliquid qui neque distinctio repellendus officiis?Optio non enim dolorum commodi dolores, libero eveniet non similique, animi voluptatem officiis quisquam voluptate consectetur voluptas tenetur eum nihil cupiditate, aspernatur dolore voluptate optio dolor fugit doloremque non sunt, ipsum velit minima molestias praesentium maiores iste perferendis

[7] To be released at: `decomp.io`

aliquam eum saepe libero?Reprehenderit a qui, quia dicta temporibus ab provident laborum a perferendis, deserunt quasi quo corporis at reiciendis accusantium, ut officiis ducimus