Neural-Davidsonian Semantic Proto-role Labeling

Rachel Rudinger
Johns Hopkins University

Adam Teichert
Johns Hopkins Univeristy

Ryan Culkin
Johns Hopkins University

Sheng Zhang
Johns Hopkins University

Benjamin Van Durme
Johns Hopkins University

Abstract

We present a model for semantic proto-role labeling (SPRL) using an adapted bidirectional LSTM encoding strategy that we call *Neural-Davidsonian*: predicate-argument structure is represented as pairs of hidden states corresponding to predicate and argument head tokens of the input sequence. We demonstrate: (1) state-of-the-art results in SPRL, and (2) that our network naturally shares parameters between attributes, allowing for learning new attribute types with limited added supervision.

1 Introduction

Universal Decompositional Semantics (UDS) (White et al., 2016) is a contemporary semantic representation of text (Abend and Rappoport, 2017) that forgoes traditional inventories of semantic categories in favor of bundles of simple, interpretable properties. In particular, UDS includes a practical implementation of Dowty's theory of *thematic proto-roles* (Dowty, 1991): arguments are labeled with properties typical of Dowty's *proto-agent* (AWARENESS, VOLITION ...) and *proto-patient* (CHANGED STATE ...).

Annotated corpora have allowed the exploration of *Semantic Proto-role Labeling* (SPRL) [1] as a natural language processing task (Reisinger et al., 2015; White et al., 2016; Teichert et al., 2017). For example, consider the following sentence, in which a particular pair of predicate and argument heads have been emphasized: "The cat *ate* the rat." An SPRL system must infer from the context of the sentence whether the rat had VOLITION, CHANGED-STATE, and EXISTED-AFTER the *eating* event (see Table 2 for more properties).

We present an intuitive neural model that achieves state-of-the-art performance for SPRL.[2] As depicted in Figure 1, our model's architecture is an extension of the bidirectional LSTM, capturing a Neo-Davidsonian like intuition, wherein select pairs of hidden states are concatenated to yield a dense representation of predicate-argument structure and fed to a prediction layer for end-to-end training. We include a thorough quantitative analysis highlighting the contrasting errors between the proposed model and previous (non-neural) state-of-the-art.

In addition, our network naturally shares a subset of parameters between attributes. We demonstrate how this allows learning to predict new at-

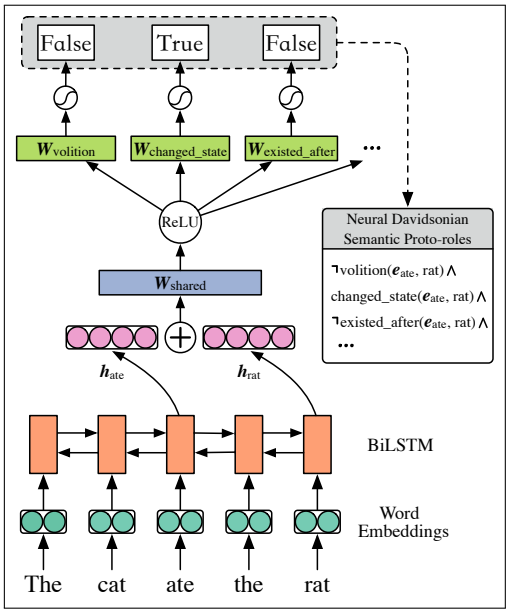

Figure 1: BiLSTM sentence encoder with SPR decoder. Semantic proto-role labeling is with respect to a specific predicate and argument within a sentence, so the decoder receives the two corresponding hidden states.

[1] SPRL and SPR refer to the labeling task and the underlying semantic representation, respectively.

[2] Implementation available at https://github.com/decomp-sem/neural-sprl.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 944–955
Brussels, Belgium, October 31 - November 4, 2018.

SPR Property	Explanation of Property		
INSTIGATION	`Arg` caused the `Pred` to happen?	✓	✗
VOLITIONAL	`Arg` chose to be involved in the `Pred`?	✓	✗
AWARE	`Arg` was/were aware of being involved in the `Pred`?	✓	✓
PHYSICALLY EXISTED	`Arg` existed as a physical object?	✓	✓
EXISTED AFTER	`Arg` existed after the `Pred` stopped?	✓	✗
CHANGED STATE	The `Arg` was/were altered or somehow changed during or by the end of the `Pred`?	✓	✓

Table 1: Example SPR annotations for the toy example "The cat ate the rat," where the `Predicate` in question is "*ate*" and the `Argument` in question is either "cat" or "rat." Note that not all SPR properties are listed, and the binary labels (✓, ✗) are coarsened from a 5-point Likert scale.

tributes with limited supervision: a key finding that could support efficient expansion of new SPR attribute types in the future.

2 Background

Davidson (1967) is credited for representations of meaning involving propositions composed of a fixed arity predicate, all of its core arguments arising from the natural language syntax, and a distinguished event variable. The earlier example could thus be denoted (modulo tense) as $(\exists e)\textbf{eat}[(e, \text{CAT}, \text{RAT})]$, where the variable e is a *reification* of the eating event. The order of the arguments in the predication implies their role, where leaving arguments unspecified (as in "The cat eats") can be handled either by introducing variables for unstated arguments, e.g., $(\exists e)(\exists x)[\textbf{eat}(e, \text{CAT}, x)]$, or by creating new predicates that correspond to different arities, e.g., $(\exists e)\textbf{eat\_intransitive}[(e, \text{CAT})]$.[3] The Neo-Davidsonian approach (Castañeda, 1967; Parsons, 1995), which we follow in this work, allows for variable arity by mapping the argument positions of individual predicates to generalized *semantic roles*, shared across predicates,[4] e.g., AGENT, PATIENT and THEME, in: $(\exists e)[\textbf{eat}(e) \wedge \textbf{Agent}(e, \text{CAT}) \wedge \textbf{Patient}(e, \text{RAT})]$.

Dowty (1991) conjectured that the distinction between the role of a prototypical **Agent** and prototypical **Patient** could be decomposed into a number of semantic properties such as *"Did the argument change state?"*. Here we formulate this as a Neo-Davidsonian representation employing *semantic proto-role* (SPR) attributes:

$$(\exists e)\,[\textbf{eat}(e) \\ \wedge \textbf{volition}(e, \text{CAT}) \wedge \textbf{instigation}(e, \text{CAT})\ldots \\ \wedge \neg\textbf{volition}(e, \text{RAT}) \wedge \textbf{destroyed}(e, \text{RAT})\ldots\,]$$

Dowty's theory was empirically verified by Kako (2006), followed by pilot (Madnani et al., 2010) and large-scale (Reisinger et al., 2015) corpus annotation efforts, the latter introducing a logistic regression baseline for SPRL. Teichert et al. (2017) refined the evaluation protocol,[5] and developed a CRF (Lafferty et al., 2001) for the task, representing existing state-of-the-art.

Full details about the SPR datasets introduced by Reisinger et al. (2015) and White et al. (2016), which we use in this work, are provided in Appendix B. For clarity, Table 1 shows a toy SPRL example, including a few sample SPR properties and explanations.

3 "Neural-Davidsonian" Model

Our proposed SPRL model (Fig. 1) determines the value of each attribute (e.g., VOLITION) on an *argument* (a) with respect to a particular *predication* (e) as a function on the latent states associated with the pair, (e, a), in the context of a full sentence. Our architecture encodes the sentence using a shared, one-layer, bidirectional LSTM (Hochreiter and Schmidhuber, 1997; Graves et al., 2013). We then obtain a continuous, vector representation $\boldsymbol{h}_{ea} = [\boldsymbol{h}_e; \boldsymbol{h}_a]$, for each predicate-argument pair as the concatenation of the hidden BiLSTM

[3] This formalism aligns with that used in PropBank (Palmer et al., 2005), which associated numbered, core arguments with each sense of a verb in their corpus annotation.

[4] For example, as seen in FrameNet (Baker et al., 1998).

[5] Splitting train/dev/test along Penn Treebank boundaries and casting the SPRL task as multi-label binary classification.

states $\boldsymbol{h}_e$ and $\boldsymbol{h}_a$ corresponding to the syntactic head of the predicate of e and argument a respectively. These heads are obtained over gold syntactic parses using the predicate-argument detection tool, PredPatt (White et al., 2016).[6]

For each SPR attribute, a score is predicted by passing $\boldsymbol{h}_{ea}$ through a separate two-layer perceptron, with the weights of the first layer shared across all attributes:

$$\text{Score}(\text{attr}, \boldsymbol{h}_{ea}) = \boldsymbol{W}_{\text{attr}} \left[g \left(\boldsymbol{W}_{\text{shared}} \left[\boldsymbol{h}_{ea} \right] \right) \right]$$

This architecture accomodates the definition of SPRL as multi-label binary classification given by Teichert et al. (2017) by treating the score as the log-odds of the attribute being present (i.e. $\text{P}(\text{attr}|\boldsymbol{h}_{ea}) = \frac{1}{1+\exp[-\text{Score}(\text{attr}, \boldsymbol{h}_{ea})]}$). This architecture also supports SPRL as a *scalar* regression task where the parameters of the network are tuned to directly minimize the discrepancy between the predicted score and a reference scalar label. The loss for the binary and scalar models are negative log-probability and squared error, respectively; the losses are summed over all SPR attributes.

Training with Auxiliary Tasks A benefit of the shared neural-Davidsonian representation is that it offers many levels at which multi-task learning may be leveraged to improve parameter estimation so as to produce semantically rich representations $\boldsymbol{h}_{ea}$, $\boldsymbol{h}_e$, and $\boldsymbol{h}_a$. For example, the sentence encoder might be pre-trained as an encoder for machine translation, the argument representation $\boldsymbol{h}_a$ can be jointly trained to predict word-sense, the predicate representation, $\boldsymbol{h}_e$, could be jointly trained to predict factuality (Saurí and Pustejovsky, 2009; Rudinger et al., 2018), and the predicate-argument representation, $\boldsymbol{h}_{ea}$, could be jointly trained to predict other semantic role formalisms (e.g. PropBank SRL—suggesting a neural-Davidsonian *SRL* model in contrast to recent BIO-style neural models of SRL (He et al., 2017)).

To evaluate this idea empirically, we experimented with a number of multi-task training strategies for SPRL. While all settings outperformed prior work in aggregate, simply initializing the BiLSTM parameters with a pretrained English-to-French machine translation encoder[7] produced the best results,[8] so we simplify discussion by focusing on that model. The efficacy of MT pretraining that we observe here comes as no surprise given prior work demonstrating, e.g., the utility of bitext for paraphrase (Ganitkevitch et al., 2013), that NMT pretraining yields improved contextualized word embeddings[9] (McCann et al., 2017), and that NMT encoders specifically capture useful features for SPRL (Poliak et al., 2018).

Full details about each multi-task experiment, including a full set of ablation results, are reported in Appendix A; details about the corresponding datasets are in Appendix B.

Except in the ablation experiment of Figure 2, our model was trained on only the SPRL data and splits used by Teichert et al. (2017) (learning all properties jointly), using GloVe[10] embeddings and with the MT-initialized BiLSTM. Models were implemented in PyTorch and trained end-to-end with Adam optimization (Kingma and Ba, 2014) and a default learning rate of 10^{-3}. Each model was trained for ten epochs, selecting the best-performing epoch on dev.

Prior Work in SPRL We additionally include results from prior work: "LR" is the logistic-regression model introduced by Reisinger et al. (2015) and "CRF" is the CRF model (specifically SPRL*) from Teichert et al. (2017). Although White et al. (2016) released additional SPR annotations, we are unaware of any benchmark results on that data; however, our multi-task results in Appendix A do use the data and we find (unsurprisingly) that concurrent training on the two SPR datasets can be helpful. Using only data and splits from White et al. (2016), the scalar regression architecture of Table 6 achieves a Pearson's ρ of 0.577 on test.

There are a few noteworthy differences between our neural model and the CRF of prior work. As an adapted BiLSTM, our model easily ex-

[6] Observed to be state-of-the-art by Zhang et al. (2017).

[7] using a modified version of `OpenNMT-py` (Klein et al., 2017) trained on the 10^9 Fr-En corpus (Callison-Burch et al., 2009) (Appendix A).

[8] e.g. this initialization resulted in raising micro-averaged F1 from 82.2 to 83.3

[9] More recent discoveries on the usefulness of language model pretraining (Peters et al., 2018; Howard and Ruder, 2018) for RNN encoders suggest a promising direction for future SPRL experiments.

[10] 300-dimensional, uncased; `glove.42B.300d` from https://nlp.stanford.edu/projects/glove/; 15,533 out-of-vocabulary words across all datasets were assigned a random embedding (uniformly from $[-.01, .01]$). Embeddings remained fixed during training.

	previous work		this work	
	LR	CRF	binary	scalar
instigation	76.7	85.6	**88.6**	0.858
volition	69.8	86.4	**88.1**	0.882
awareness	68.8	87.3	**89.9**	0.897
sentient	42.0	85.6	**90.6**	0.925
physically existed	50.0	76.4	**82.7**	0.834
existed before	79.5	84.8	**85.1**	0.710
existed during	93.1	**95.1**	95.0	0.673
existed after	82.3	**87.5**	85.9	0.619
created	0.0	**44.4**	39.7	0.549
destroyed	17.1	0.0	**24.2**	0.346
changed	54.0	67.8	**70.7**	0.592
changed state	54.6	66.1	**71.0**	0.604
changed possession	0.0	38.8	**58.0**	0.640
changed location	6.6	35.6	**45.7**	0.702
stationary	13.3	21.4	**47.4**	0.711
location	0.0	18.5	**53.8**	0.619
physical contact	21.5	40.7	**47.2**	0.741
manipulated	72.1	86.0	**86.8**	0.737
micro f1	71.0	81.7	**83.3**	
macro f1	55.4*	65.9*	**71.1**	
macro-avg pearson				0.753

Table 2: SPR comparison to Teichert et al. (2017). Bold number indicate best F1 results in each row. Right-most column is pearson correlation coeficient for a model trained and tested on the scalar regression formulation of the same data.

ploits the benefits of large-scale pretraining, in the form of GloVe embeddings and MT pretraining, both absent in the CRF. Ablation experiments (Appendix A) show the advantages conferred by these features. In contrast, the discrete-featured CRF model makes use of gold dependency labels, as well as joint modeling of SPR attribute pairs with explicit joint factors, both absent in our neural model. Future SPRL work could explore the use of models like the LSTM-CRF (Lample et al., 2016; Ma and Hovy, 2016) to combine the advantages of both paradigms.

4 Experiments

Table 2 shows a side-by-side comparison of our model with prior work. The full breakdown of F1 scores over each individual property is provided. For every property except EXISTED DURING, EXISTED AFTER, and CREATED we are able to exceed prior performance. For some properties, the absolute F1 gains are quite large: DESTROYED (+24.2), CHANGED POSSESSION (+19.2.0), CHANGED LOCATION (+10.1), STATIONARY (+26.0) and LOCATION (+35.3). We also report performance with a scalar regression version of the model, evaluated with Pearson correlation. The scalar model is with respect to the original SPR annotations on a 5-point Likert scale, instead of a binary cut-point along that scale (> 3).

		phys. contact			volition		
		# DIFFER	Δ FALSE−	Δ FALSE+	# DIFFER	Δ FALSE−	Δ FALSE+
1	ALL	80	−14	6	80	−14	−10
2	PROPERNOUN	18	−2	−2	21	4	−5
3	ORG.	15	−9	2	31	−6	−1
4	PRONOUN	10	0	8	12	0	0
5	PHRASEVERB	14	−6	0	9	−4	1
6	METAPHOR	11	−5	−2	6	−2	0
7	LIGHTVERB	5	−2	1	5	−1	2

Table 3: Manual error analysis on a sample of instances (80 for each property) where outputs of CRF and the binary model from Table 2 differ. **Negative** Δ FALSE+ and Δ FALSE− indicate the neural model represents a **net *reduction* in type I and type II errors** respectively over CRF. Positive values indicate a net increase in errors. Each row corresponds to one of several (overlapping) subsets of the 80 instances in disagreement: (1) all (sampled) instances; (2) argument is a proper noun; (3) argument is an organization or institution; (4) argument is a pronoun; (5) predicate is phrasal or a particle verb construction; (6) predicate is used metaphorically; (7) predicate is a light-verb construction. #DIFFER is the size of the respective subset.

Manual Analysis We select two properties (VOLITION and MAKES PHYSICAL CONTACT) to perform a manual error analysis with respect to CRF [11] and our binary model from Table 2. For each property, we sample 40 dev instances with gold labels of "True" (> 3) and 40 instances of "False" (≤ 3), restricted to cases where the two system predictions disagree.[12] We manually label each of these instances for the six features shown in Table 3. For example, given the input "He *sits* down at the piano and plays," our neural model correctly predicts that He makes physical contact during the *sitting*, while CRF does not. Since He is a pronoun, and *sits down* is phrasal, this example contributes -1 to Δ FALSE− in rows 1, 4 and 5.

[11] We obtained the CRF dev system predictions of Teichert et al. (2017) via personal communication with the authors.

[12] According to the reference, of the 1071 dev examples, 150 have physical contact and 350 have volition. The two models compared here differed in phy. contact on 62 positive and 44 negative instances and for volition on 43 positive and 54 negative instances.

Figure 2: Effect of using only a fraction of the training data for a property while either ignoring or co-training with the full training data for the other SPR1 properties. Measurements at 1%, 5%, 10%, 25%, 50%, and 100%.

For both properties our model appears more likely to correctly classify the argument in cases where the predicate is a phrasal verb. This is likely a result of the fact that the BiLSTM has stronger language-modeling capabilities than the CRF, particularly with MT pretraining. In general, our model increases the false-positive rate for MAKES PHYSICAL CONTACT, but especially when the argument is pronominal.

Learning New SPR Properties One motivation for the decompositional approach adopted by SPRL is the ability to incrementally build up an inventory of annotated properties according to need and budget. Here we investigate (1) the degree to which having less training data for a single property degrades our F1 for that property on held-out data and (2) the effect on degradation of concurrent training with the other properties. We focus on two properties only: INSTIGATION, a canonical example of a proto-agent property, and MANIPULATED, which is a proto-patient property. For each we consider six training set sizes (1, 5, 10, 25, 50 and 100 percent of the instances). Starting with the same randomly initialized BiLSTM[13], we consider two training scenarios: (1) ignoring the remaining properties or (2) including the model's loss on other properties with a weight of $\lambda = 0.1$ in the training objective.

Results are presented in Figure 2. We see that, in every case, most of the performance is achieved with only 25% of the training data. The curves also suggest that training simultaneously on all SPR properties allows the model to learn the target property more quickly (i.e., with fewer training samples) than if trained on that property in isolation. For example, at 5% of the training training data, the "all properties" models are achieving roughly the same F1 on their respective target property as the "target property only" models achieves at 50% of the data.[14] As the SPR properties currently annotated are by no means semantically exhaustive,[15] this experiment indicates that future annotation efforts may be well served by favoring breadth over depth, collecting smaller numbers of examples for a larger set of attributes.

5 Conclusion

Inspired by: (1) the SPR decomposition of predicate-argument relations into overlapping feature bundles and (2) the neo-Davidsonian formalism for variable-arity predicates, we have proposed a straightforward extension to a BiLSTM classification framework in which the states of pre-identified predicate and argument tokens are pairwise concatenated and used as the target for SPR prediction. We have shown that our *Neural-Davidsonian* model outperforms the prior state of the art in aggregate and showed especially large gains for properties of CHANGED-POSSESSION, STATIONARY, and LOCATION. Our architecture naturally supports discrete or continuous label paradigms, lends itself to multi-task initialization or concurrent training, and allows for parameter sharing across properties. We demonstrated this sharing may be useful when some properties are only sparsely annotated in the training data, which is suggestive of future work in efficiently increasing the range of annotated SPR property types.

Acknowledgments

This research was supported by the JHU HLTCOE, DARPA AIDA, and NSF GRFP (Grant No. DGE-1232825). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA, NSF, or the U.S. Government.

[13] Note that this experiment does not make use of MT pretraining as was used for Table 2, to best highlight the impact of parameter sharing across attributes.

[14] As we observed the same trend more clearly on the dev set, we suspect some over-fitting to the development data which was used for independently select a stopping epoch for each of the plotted points.

[15] E.g., annotations do not include any questions relating to the *origin* or *destination* of an event.

References

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 77–89.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English web treebank. *Linguistic Data Consortium, Philadelphia, PA*.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. Propbank: Semantics of new predicate types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Hector Neri Castañeda. 1967. Comment on d. davidsons "the logical forms of action sentences". In N. Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, Pittsburgh.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Donald Davidson. 1967. The logical forms of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, Pittsburgh.

David Dowty. 1991. Thematic proto-roles and argument selection. *Language*, 67(3):547–619.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE.

Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

N. Ide and J. Pustejovsky. 2017. *Handbook of Linguistic Annotation*. Springer Netherlands.

Edward Kako. 2006. Thematic role properties of subjects and objects. *Cognition*, 101(1):1–42.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1528–1533, San Diego, California. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML

'01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *International Conference on Learning Representations*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Nitin Madnani, Jordan Boyd-Graber, and Philip Resnik. 2010. Measuring transitivity using untrained annotators. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazons Mechanical Turk*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6294–6305. Curran Associates, Inc.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Terence Parsons. 1995. Thematic relations and arguments. *Linguistic Inquiry*, pages 635–662.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Adam Poliak, Yonatan Belinkov, James Glass, and Benjamin Van Durme. 2018. On the evaluation of semantic phenomena in neural machine translation using natural language inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 513–523.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3:475–488.

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Louisiana. Association for Computational Linguistics.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3):227.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Adam Teichert, Adam Poliak, Benjamin Van Durme, and Matthew R Gormley. 2017. Semantic proto-role labeling. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Sheng Zhang, Rachel Rudinger, and Benjamin Van Durme. 2017. An Evaluation of PredPatt and Open IE via Stage 1 Semantic Role Labeling. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*.

Name	#	Description
LR		Logistic Regr. model, Reisinger et al. (2015)
CRF		CRF model, Teichert et al. (2017)
SPR1	0	SPR1 basic model
SPR1-RAND	0	SPR1, random word embeddings
MT:SPR1	1a	SPR1 after MT pretraining
PB:SPR1	1a	SPR1 after PB pretraining
MT:PB:SPR1	1a	SPR1 after MT+PB pretraining
SPR1+2	1b	SPR1 and SPR2 concurrently
SPR1+WSD	1b	SPR1 and WSD concurrently
MT:SPR1+2	1b	SPR1+2 after MT pretraining
MT:SPR1+WSD	1b	SPR1+WSD after MT pretraining
MT:SPR1S	1c	SPR1 scalar after MT pretraining
PB:SPR1S	1c	SPR1 scalar after PB pretraining
PS-MS	1d	SPR1 propty-specific model sel.
SPR2	3	SPR2 basic scalar model
MT:SPR2	3	SPR2 after MT pretraining
PB:SPR2	3	SPR2 after PB pretraining
MT:PB:SPR2	3	SPR2 after MT+PB pretraining

Table 4: Name and short description of each experimental condition reported. MT: indicates pretraining with machine translation; PB: indicates pretraining with PropBank SRL.

A Mult-Task Investigation

Multi-task learning has been found to improve performance on many NLP tasks, particularly for neural models, and is rapidly becoming *de rigueur* in the field. The strategy involves optimizing for multiple training objectives corresponding to different (but usually related) tasks. Collobert and Weston (2008) use multi-task learning to train a convolutional neural network to perform multiple core NLP tasks (POS tagging, named entity recognition, etc.). Multi-task learning has also been used to improve sentence compression (Klerke et al., 2016), chunking and dependency parsing (Hashimoto et al., 2017). Related work on UDS (White et al., 2016) shows improvements on event factuality prediction with multi-task learning on BiLSTM models (Rudinger et al., 2018). To complete the basic experiments reported in the main text, here we include an investigation of the impact of multi-task learning for SPRL.

We borrow insights from Mou et al. (2016) who explore different multi-task strategies for NLP including approach of initializing a network by training it on a related task ("INIT") versus interspersing tasks during training ("MULT"). Here we employ both of these strategies, referring to them as *pretraining* and *concurrent* training. We also use the terminology *target task* and *auxiliary task* to differentiate the primary task(s) we are interested in from those that play only a supporting role in training. In order to tune the impact of auxiliary tasks on the learned representation, Luong et al. (2016) use a *mixing parameter*, α_i, for each task i. Each parameter update consists of selecting a task with probability proportional to its α_i and then performing one update with respect to that task alone. They show that the choice of α has a large impact on the effect of multi-task training, which influences our experiments here.

Please refer to Appendix B for details on the datasets used in this section. In particular, with a few exceptions, White et al. (2016) annotates for the same set of properties as Reisinger et al. (2015), but with slightly different protocol and on a different genre. However, in this section we treat the two datasets as if they were separate tasks. To avoid cluttering the results in the main text, we exclusively present results there on what we call *SPR1* which consists of the data from Reisinger et al. (2015) and the train/dev/test splits of Teichert et al. (2017). We refer to the analogous tasks built on the data and splits of White et al. (2016) using the term SPR2. (We are not aware of any prior published results on property prediction for the SPR2.)

In addition to the binary and scalar SPR architectures outlined in Section 3 of the main paper, we also considered concurrently training the BiLSTM on a fine-grained word-sense disambiguation task or on joint SPR1 and SPR2 prediction. We also experimented with using machine translation and PropBank SRL to initialize the parameters of the BiLSTM. Preliminary experimentation on dev data with other combinations helped prune down the set of interesting experiments to those listed in Table 4 which assigns names to the models explored here. Our ablation study in Section 4 of the main paper uses the model named SPR1 while the other results in the main paper correspond to MT:SPR1 in the case of binary prediction and MT:SPR1S in the case of scalar prediction. After detailing the additional components used for pretraining or concurrent training, we present aggregate results and for the best performing models (according to dev) we present property-level aggregate results.

A.1 Auxiliary Tasks

Each auxiliary task is implemented in the form of a task-specific decoder with access to the hidden

states computed by the shared BiLSTM encoder. In this way, the losses from these tasks backpropagate through the BiLSTM. Here we describe each task-specific decoder.

PropBank Decoder The network architecture for the auxiliary task of predicting abstract role types in PropBank is nearly identical to the architecture for SPRL described in Section 3 of the main paper. The main difference is that the PropBank task is a single-label, categorical classification task.

$$\mathrm{P}(\mathrm{role}_i|\boldsymbol{h}_{ea}) = \mathrm{softmax}_i\left(\boldsymbol{W}_{\text{propbank}}\left[\boldsymbol{h}_{ea}\right]\right)$$

The loss from this decoder is the negative log of the probability assigned to the correct label.

Supersense Decoder The word sense disambiguation decoder computes a probability distribution over 26 WordNet supersenses with a simple single-layer feedforward network:

$$\mathrm{P}(\mathrm{supersense}_i|\boldsymbol{h}_a) = \mathrm{softmax}_i(\boldsymbol{W}\left[\boldsymbol{h_a}\right])$$

where $\boldsymbol{W} \in \mathbb{R}^{1200\times 26}$ and $\boldsymbol{h}_a$ is the RNN hidden state corresponding to the argument head token we wish to disambiguate. Since the gold label in the supersense prediction task is a *distribution* over supersenses, the loss from this decoder is the cross-entropy between its predicted distribution and the gold distribution.

French Translation Decoder Given the encoder hidden states, the goal of translation is to generate the reference sequence of tokens $Y = y_1, \cdots, y_n$ in the target language, i.e., French. We employ the standard decoder architecture for neural machine translation. At each time step i, the probability distribution of the decoded token y_i is defined as:

$$P(y_i) = \mathrm{softmax}\big(\tanh(\boldsymbol{W}_{\text{fr}}\big[\boldsymbol{s_i}; \boldsymbol{c_i}\big] + \boldsymbol{b}_{\text{fr}})\big)$$

where $\boldsymbol{W}_{\text{fr}}$ is a transform matrix, and $\boldsymbol{b}_{\text{fr}}$ is a bias. The inputs are the decoder hidden state $\boldsymbol{s_i}$ and the context vector $\boldsymbol{c_i}$. The decoder hidden state $\boldsymbol{s_i}$ is computed by:

$$\boldsymbol{s_i} = \mathrm{RNN}(\boldsymbol{y_{i-1}}, \boldsymbol{s_{i-1}})$$

where RNN is a recurrent neural network using L-layer stacked LSTM, $\boldsymbol{y_{i-1}}$ is the word embedding of token y_{i-1}, and $\boldsymbol{s_0}$ is initialized by the last encoder left-to-right hidden state.

	micro-F1	macro-F1
LR	71.0	55.4⋆
CRF	81.7	65.9⋆
SPR1-RAND	77.7	57.3
SPR1	**82.2**	69.3
MT:SPR1	**83.3**	71.1
PB:SPR1	**82.3**	67.9
MT:PB:SPR1	**82.8**	70.9
SPR1+2	**83.3**	70.4
SPR1+WSD	**81.9**	67.9
MT:SPR1+2	**83.2**	70.0
MT:SPR1+WSD	**81.8**	67.4
PS-MS	**82.9**	69.5

Table 5: Overall test performance for all settings described in Experiments 1 and 1a-d. The target task is SPR1 as binary classification. Micro- and macro-F1 are computed over all properties. (⋆Baseline macro-F1 scores are computed from property-specific precision and recall values in Teichert et al. (2017) and may introduce rounding errors.)

The context vector $\boldsymbol{c_i}$ is computed by an attention mechanism (Bahdanau et al., 2014; Luong et al., 2015),

$$\boldsymbol{c_i} = \sum_t \alpha_{i,t}\boldsymbol{h_t},$$

$$\alpha_{i,t} = \frac{\exp\left(\boldsymbol{s_i}^\top(\boldsymbol{W_\alpha h_t} + \boldsymbol{b_\alpha})\right)}{\sum_k \exp\left(\boldsymbol{s_i}^\top(\boldsymbol{W_\alpha h_k} + \boldsymbol{b_\alpha})\right)},$$

where $\boldsymbol{W_\alpha}$ is a transform matrix and $\boldsymbol{b_\alpha}$ is a bias. The loss is the negative log-probability of the decoded sequence.

A.2 Results

In this section, we present a series of experiments using different components of the neural architecture described in Section 3, with various training regimes. Each experimental setting is given a name (in SMALLCAPS) and summarized in Table 4. Unless otherwise stated, the target task is SPR1 (classification). To ease comparison, we include results from the main paper as well as additional results.

Experiment 0: Embeddings By default, all models reported in this paper employ pretrained word embeddings (GloVe). In this experiment we replaced the pretrained embeddings in the vanilla

	CRF	SPR1	MT:SPR1	SPR1+2
instigation	85.6	84.6	**88.6**	85.6
volition	86.4	**87.9**	**88.1**	**88.0**
awareness	87.3	**88.3**	**89.9**	**88.4**
sentient	85.6	**89.6**	**90.6**	**90.0**
physically existed	76.4	**82.3**	**82.7**	**80.2**
existed before	84.8	**86.0**	**85.1**	**86.8**
existed during	95.1	94.2	95.0	94.8
existed after	87.5	86.9	85.9	87.5
created	44.4	**46.6**	39.7	**51.6**
destroyed	0.0	**11.1**	**24.2**	**6.1**
changed	67.8	67.4	**70.7**	**68.1**
changed state	66.1	**66.8**	**71.0**	**67.1**
changed possession	38.8	**57.1**	**58.0**	**63.7**
changed location	35.6	**60.0**	**45.7**	**52.9**
stationary	21.4	**43.2**	**47.4**	**53.1**
location	18.5	**46.9**	**53.8**	**53.6**
physical contact	40.7	**52.7**	**47.2**	**54.7**
manipulated	86.0	82.2	**86.8**	**86.7**
micro f1	81.7	**82.2**	**83.3**	**83.3**
macro f1	65.9	**69.3**	**71.1**	**70.4**

Table 6: Breakdown by property of binary classification F1 on SPR1. All new results outperforming prior work (CRF) in bold.

SPR1 model (SPR1) with randomly initialized word embeddings (SPR1-RAND). The results (Table 5) reveal substantial gains from the use of pretrained embeddings; this is likely due to the comparatively small size of the SPR1 training data.

Experiment 1a: Multi-task Pretraining We pretrained the BiLSTM encoder with two separate auxiliary tasks: **French Translation** and **PropBank Role Labeling**. There are three settings: (1) Translation pretraining only (MT:SPR1), (2) PropBank pretraining only (PB:SPR1), and (3) Translation pretraining followed by PropBank pretraining (MT:PB:SPR1). In each case, after pretraining, the SPRL decoder is trained end-to-end, as in Experiment 0 (on SPR1 data).

Experiment 1b: Multi-task Concurrent One auxiliary task (**Supersense** or **SPR2**) is trained concurrently with SPR1 training. In one epoch of training, a training example is sampled at random (without replacement) from either task until all training instances have been sampled. The loss from the auxiliary task (which, in both cases, has more training instances than the target SPRL task) is down-weighted in proportion to ratio of the dataset sizes:

$$\alpha = \frac{|\text{target task}|}{|\text{auxiliary task}|}$$

SPR property	SPR1S	MT:SPR1S	SPR2
instigation	0.835	0.858	0.590
volition	0.869	0.882	0.837
awareness	0.873	0.897	0.879
sentient	0.917	0.925	0.880
physically existed	0.820	0.834	-
existed before	0.696	0.710	0.618
existed during	0.666	0.673	0.358
existed after	0.612	0.619	0.478
created	0.540	0.549	-
destroyed	0.268	0.346	-
changed	0.619	0.592	-
changed state	0.616	0.604	0.352
changed possession	0.359	0.640	0.488
change of location	0.778	0.702	0.492
changed state continuous	-	-	0.373
was for benefit	-	-	0.578
stationary	0.705	0.711	-
location	0.627	0.619	-
physical contact	0.731	0.741	-
manipulated	0.715	0.737	-
was used	-	-	0.203
partitive	-	-	0.359
macro-avg pearson	0.743	0.753	0.591

Table 7: SPR1 and SPR2 as scalar prediction tasks. Pearson correlation between predicted and gold values.

The auxiliary task loss is further down-weighted by a hyperparameter $\lambda \in \{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ which is chosen based on dev results. We apply this training regime with the auxiliary task of Supersense prediction (SPR1+WSD) and the scalar SPR2 prediction task (SPR1+SPR2), described in Experiment 2.

Experiment 1c: Multi-task Combination This setting is identical to Experiment 1b, but includes MT pretraining (the best-performing pretraining setting on dev), as described in 1a. Accordingly, the two experiments are MT:SPR1+WSD and MT:SPR1+SPR2.

Experiment 1d: Property-Specific Model Selection (PS-MS) Experiments 1a–1c consider a variety of pretraining tasks, co-training tasks, and weight values, λ, in an effort to improve aggregate F1 for SPR1. However, the SPR properties are diverse, and we expect to find gains by choosing training settings on a property-specific basis. Here, for each property, we select from the set of models considered in experiments 1a–1c the one that achieves the highest dev F1 for the target property. We report the results of applying those property-specific models to the test data.

SPR1S	0.743	SPR2	**0.591**
MT:SPR1S	**0.753**	MT:SPR2	0.577
PB:SPR1S	0.731	PB:SPR2	0.568
MT:PB:SPR1S	0.720	MT:PB:SPR2	0.564

Table 8: SPR1 and SPR2 as scalar prediction tasks. The overall performance for each experimental setting is reported as the average Pearson correlation over all properties. Highest SPR1 and SPR2 results are in bold.

Experiment 2: SPR as a scalar task In Experiment 2, we trained the SPR decoder to predict properties as scalar instead of binary values. Performance is measured by Pearson correlation and reported in Tables 8 and 7. In this case, we treat SPR1 and SPR2 both as target tasks (separately). By including SPR1 as a target task, we are able to compare (1) SPR as a binary task and a scalar task, as well as (2) SPR1 and SPR2 as scalar tasks. These results constitute the first reported numbers on SPR2.

We observe a few trends. First, it is generally the case that properties with high F1 on the SPR1 binary task also have high Pearson correlation on the SPR1 scalar task. The higher scoring properties in SPR1 scalar are also generally the higher scoring properties in SPR2 (where the SPR1 and SPR2 properties overlap), with a few notable exceptions, like INSTIGATION. Overall, correlation values are lower in SPR2 than SPR1. This may be the case for a few reasons. (1) The underlying data in SPR1 and SPR2 are quite different. The former consists of sentences from the Wall Street Journal via PropBank (Palmer et al., 2005), while the latter consists of sentences from the English Web Treebank (Bies et al., 2012) via the Universal Dependencies; (2) certain filters were applied in the construction of the SPR1 dataset to remove instances where, e.g., predicates were embedded in a clause, possibly resulting in an easier task; (3) SPR1 labels came from a single annotator (after determining in pilot studies that annotations from this annotator correlated well with other annotators), where SPR2 labels came from 24 different annotators with scalar labels averaged over two-way redundancy.

Discussion With SPR1 binary classification as the target task, we see overall improvements from various multi-task training regimes (Experiments 1a-d, Tables 5 and 6), using four different auxiliary tasks: machine translation into French, PropBank abstract role prediction, word sense disambiguation (WordNet supersenses), and SPR2.[16] These auxiliary tasks exhibit a loose trade-off in terms of the quantity of available data and the semantic relatedness of the task: MT is the least related task with the most available (parallel) data, while SPR2 is the most related task with the smallest quantity of data. While we hypothesized that the relatedness of PropBank role labeling and word sense disambiguation tasks might lead to gains in SPR performance, we did not see substantial gains in our experiments (PB:SPR1, SPR1+WSD). We did, however, see improvements over the target-task only model (SPR1) in the cases where we added MT pretraining (MT:SPR1) or SPR2 concurrent training (SPR1+2). Interestingly, combining MT pretraining with SPR2 concurrent training yielded no further gains (MT:SPR1+2).

B Data

SPR1 The SPR1.0 ("SPR1") dataset introduced by Reisinger et al. (2015) contains proto-role annotations on 4,912 Wall Street Journal sentences from PropBank (Palmer et al., 2005) corresponding to 9,738 predicate-argument pairs with 18 properties each, in total 175,284 property annotations. All annotations were performed by a single, trusted annotator. Each annotation is a rating from 1 to 5 indicating the likelihood that the property applies, with an additional "N/A" option if the question of whether the property holds is nonsensical in the context.

To compare with prior work (Teichert et al., 2017), we treat the SPR1 data as a binary prediction task: the values 4 and 5 are mapped to **True** (property holds), while the values 1, 2, 3, and "N/A" are mapped to **False** (property does not hold). In additional experiments, we move to treating SPR1 as a scalar prediction task; in this case, "N/A" is mapped to 1, and all other annotation values remain unchanged.

SPR2 The second SPR release (White et al., 2016) contains annotations on 2,758 sentences from the English Web Treebank (EWT) (Bies et al., 2012) portion of the Universal Dependencies (v1.2) (Silveira et al., 2014)[17], corresponding

[16] Note that in some cases we treat SPR2 as an auxiliary task, and in others, the target task.

[17] We exclude the SPR2 pilot data; if included, the SPR2

to 6,091 predicate-argument pairs. With 14 proto-role properties each, there are a total of 85,274 annotations, with two-way redundancy. As in SPR1, the value of each annotation is an integral value 1-5 or "N/A." We treat SPR2 as a scalar prediction task, first mapping "N/A" to 1, and then averaging the two-way redundant annotation values to a single value.

Word Sense Disambiguation Aligned with proto-role property annotations in the SPR2 release are word sense disambiguation judgments for the head tokens of arguments. Candidate word senses (fine-grained) from WordNet (Fellbaum, 1998) were presented to Mechanical Turk workers (at least three annotators per instance), who selected every applicable sense of the word in the given context. In this work, we map the fine-grained word senses to one of 26 coarse-grained WordNet noun supersenses (e.g., `noun.animal`, `noun.event`, `noun.quantity`, etc.). In many cases, a word may be mapped to more than one supersense. We treat the supersense label on a word as a distribution over supersenses, where the probability assigned to one supersense is proportional to the number of annotators that (indirectly) selected that supersense. In practice, the entropy of these resulting supersense distributions is low, with an average perplexity of 1.42.

PropBank The PropBank project consists of predicate-argument annotations over corpora for which gold Penn TreeBank-style constituency parses are available. We use the Unified PropBank release (Bonial et al., 2014; Ide and Pustejovsky, 2017), which contains annotations over OntoNotes as well as the English Web TreeBank (EWT). Each predicate in each corpus is annotated for word sense, and each argument of each predicate is given a label such as ARG0, ARG1, etc., where the interpretation of the label is defined relative to the word sense. We use PropBank Frames to map these sense-specific labels to 16 sense-independent labels such as PAG (proto-agent), PPT (proto-patient), etc., and then formulate a task to predict the abstracted labels. Because our model requires knowledge of predicate and argument head words, we ran the Stanford Universal Dependencies converter (Schuster and Manning, 2016) over the gold constituency parses to obtain Universal Dependency parses, which were then processed by the PredPatt framework (Zhang et al., 2017; White et al., 2016) to identify head words.

English-French Data The 10^9 French-English parallel corpus (Callison-Burch et al., 2009) contains 22,520,376 French-English sentence pairs, made up of 811,203,407 French words and 668,412,817 English words. The corpus was constructed by crawling the websites of international organizations such as the Canadian government, the European Union, and the United Nations.

release contains annotations for 2,793 sentences.