Label Definitions Improve Semantic Role Labeling

Li Zhang[*]
University of Pennsylvania
zharry@seas.upenn.edu

Ishan Jindal
IBM Research
ishan.jindal@ibm.com

Yunyao Li[†]
Apple Inc.
yunyaoli@apple.com

Abstract

Argument classification is at the core of Semantic Role Labeling. Given a sentence and the predicate, a semantic role label is assigned to each argument of the predicate. While semantic roles come with meaningful definitions, existing work has treated them as symbolic. Learning symbolic labels usually requires ample training data, which is frequently unavailable due to the cost of annotation. We instead propose to retrieve and leverage the definitions of these labels from the annotation guidelines. For example, the verb predicate "work" has arguments defined as "worker", "job", "employer", etc. Our model achieves state-of-the-art performance on the CoNLL09 English SRL dataset injected with label definitions given the predicate senses. The performance improvement is even more pronounced in low-resource settings when training data is scarce.[1]

1 Introduction

Semantic role labeling (SRL) is an essential NLP task of answering the question of "who did what to whom, when, where and how." Formally, a semantic role is assigned to each argument of a predicate in a sentence. SRL has been shown to help a wide range of NLP applications such as natural language inference (Zhang et al., 2020c), question answering (Zhang et al., 2020c; Maqsud et al., 2014; Yih et al., 2016) and machine translation (Shi et al., 2016). It can also be used as a pre-processing step for tasks such as information extraction (Niklaus et al., 2018; Zhang et al., 2020a).

Learning from ample labeled examples is the predominant paradigm in many NLP tasks (Schick and Schütze, 2021), including SRL. However, labeled data is costly and often lacking in many tasks, domains, and languages. One attempt at this issue, made possible by recent advancement of language models, is to inject task descriptions into the data so that models become "aware" of the task requirements and the meaning of the labels. This technique has successfully been used in sentiment analysis (Schick and Schütze, 2021), event extraction (Du and Cardie, 2020; Zhang et al., 2021), intent detection (Zhang et al., 2020b), word sense disambiguation (Kumar et al., 2019), and many other tasks (Brown et al., 2020). In SRL, while the label space is particularly sparse as each predicate has different semantic roles, definitions are readily available for all possible arguments of supported predicates. While early work has used label definitions for frame generalization specific to FrameNet (Baker et al., 1998; Baldewein et al., 2004; Matsubayashi et al., 2009; Johansson, 2012; Kshirsagar et al., 2015), there has been no work that targets general SRL in such a label-aware fashion.

In SRL, the semantic roles are defined specifically for each predicate sense. In Figure 1, given the predicate "work" and its sense, the definitions of its arguments can be found in *frames* provided

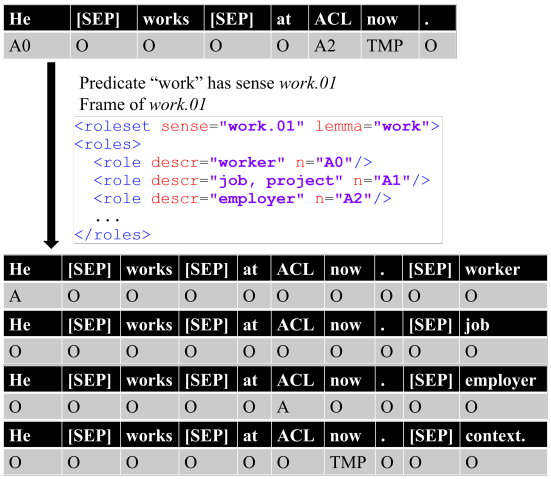

Figure 1: An illustration of our procedure of constructing SRL examples with label definitions. The sense is used to get possible arguments of a predicate.

[*] Work done during an internship at IBM Research
[†] Work done while at IBM Research
[1] Our data and code can be found at https://github.com/System-T/LabelAwareSRL.

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5613 - 5620

	Num. sentences	Num. predicates	Num. arguments
Train	39,280	179,014	393,699
Dev	1,335	6,390	13,865
Test (in)	2,400	10,498	23,286
Test (out)	426	1,259	2,859

Table 1: Some statistics of the CoNLL09 SRL dataset.

by corpora such as PropBank (Palmer et al., 2005). While previous work has treated argument labels as symbolic, we propose to inject their textual descriptions into the data, with the hypothesis that pre-trained language models can leverage them, analogous to answering questions like "what is the employer of 'work' in the sentence". We show that injecting textual descriptions helps language models (1) to outperform the previous state-of-the-art by more than 1 F1 on the CoNLL09 out-domain test set, (2) to improve the model's ability to generalize to unseen or low-frequency predicates, and (3) to better adapt to unfamiliar domains.

2 Task, Dataset, and Baseline Models

The CoNLL 2009 shared task (Hajič et al., 2009) proposed a dependency-based SRL dataset, henceforth referred to as CoNLL09, where arguments are represented as head words instead of spans. It is one of the most commonly used SRL training dataset and benchmark (statistics shown in Table 1). Notably, it includes two test sets: an in-domain one (relative to training and development set) sampled from the Wall Street Journal and an out-domain one sampled from the Brown corpus. Using the same formulation as Shi and Lin (2019) which proposed the current state-of-the-art SRL model[2], we view SRL as two sub-tasks: predicate sense disambiguation and argument classification.

The **predicate sense disambiguation** (PSD) task is to identify the word sense of a predicate in a sentence. In the sentence "She went to Shenzhen", the predicate "went" has sense *motion* and has sense label 01. The task is thus formulated as sequence classification. For simplicity, we finetune an off-the-shelf pre-trained RoBERTa-base model (Liu et al., 2019) as a baseline with performance on par with the current state-of-the-art (Table 2).

The **argument classification** (AC) task is to label each token in a sentence as either non-argument or otherwise a semantic role, given a predicate. The task is thus formulated as token classification. We enclose the predicate with separator tokens. An example is shown in the topmost of Figure 1. As before, using a simple RoBERTa-base model, we achieve performance on par with the current state-of-the-art (Table 3).[3]

[2]The model is based on a BERT+LSTM+classifier architecture, and is the default SRL model of the widely used AllenNLP toolkit (Gardner et al., 2017) as of August 2021.

	In-domain	Out-domain
Shi and Lin (2019)	96.9	90.6
(ours) RoBERTa-base	96.7	88.5

Table 2: Accuracy of our baseline and current state-of-the-art models on the PSD task.

Most previous models including our baseline perform PSD and AC independently (Shi and Lin, 2019; Marcheggiani and Titov, 2020). Next, we provide additional information to the AC data that relies on PSD, a synergy of the two sub-tasks.

3 Argument Label Definitions

We propose to expand argument classification data by injecting argument label definitions (ACD) with semantic meanings to the arguments, which are readily available. Our approach does not focus on and is agnostic to model architecture.

Source of Definitions. The CoNLL09 dataset provides frame files, one for each predicate, which contain possible word senses of each predicate, and for each of the senses, the set of possible semantic roles (i.e. argument labels) with definitions. While previous models neglected this information and only relied on symbolic argument labels such as `A0` , `A1` , we propose to expand the AC data using definitions (ACD) (Figure 1).

Argument labels are specific to predicate senses. In the training and development set, we always use the gold senses to find the corresponding argument label definitions. For the test set, we consider both the gold senses as a performance upper-bound and those predicted by our PSD model.

Adding Definitions to Examples. For each example with a predicate p of some sense, where the frame file of p has N arguments for its sense, we construct N examples, one for each of the arguments, with its definition appended, delimited by a separator token.[4] A definition may have one or

[3]Despite our best efforts, we cannot replicate the models of Shi and Lin (2019). We thus copy performance numbers from their paper and focus on our own AC baseline as a competitive model without access to definitions.

[4]The exploration of other formats is shown in Appendix C.

	In-domain				Out-domain			
	P	R	F1	Arg. F1	P	R	F1	Arg. F1
Shi and Lin (2019)	92.4	92.3	92.4	90.3	85.7	85.8	85.7	83.5
Jindal et al. (2020)	90.0	91.5	90.8	87.5	83.5	86.5	85.0	81.7
Conia et al. (2021)	-	-	91.6	-	-	-	84.6	-
Fei et al. (2021)	92.8	92.0	92.3	-	81.3	79.2	80.6	-
Munir et al. (2021)	91.2	90.6	90.9	-	83.1	82.6	82.8	-
ours								
AC	92.5	91.7	92.1	90.1	85.5	84.3	84.9	83.0
ACD (pred. sense)	93.0	91.0	92.0	90.0	86.7	84.4	85.5	83.5
ACD (gold sense)	93.2	91.3	92.2	90.2	87.3	84.8	86.0	84.6
ACD (symbolic)	-	-	-	90.1	-	-	-	83.1

Table 3: Performance on the full CoNLL09 dataset. The precision, recall and F1 are calculated based on the official scoring script which considers both sense and argument predictions. The micro F1 considers only arguments. ACD (symbolic) replaces definitions as symbolic labels (e.g., `A1`) as a control.

more tokens and is tokenized. In each constructed example, only the labels corresponding to the current argument are kept as 'A', while the rest are labeled as 'O'. Markers for discontinuous role spans (e.g. 'C-A0') and references (e.g. 'R-A0') are reduced to 'C-A' and 'R-A'. For example, in Figure 1, the first constructed example can be interpreted as asking for the *worker* of predicate "work" in the sentence.[5] In inference time, the predicted labels are converted to the numbered labels. Note that it is possible that a token is labeled as several arguments. In such scenarios, we rank the arguments according to the location they appear in corresponding frame file and choose the first argument.[6] While our experiments are based on a dependency-based SRL dataset based on PropBank, our method can be identically applied to span-based ones with other frame dictionaries such as FrameNet.

The arguments discussed in this section so far are *core* arguments which are specific to predicate senses with provided definitions in the frame files. Another type is the *contextual* arguments, such as 'TMP' for "time", 'MNR' for "manner", etc. These arguments can be applied to any predicate sense, and do not have clear definitions[7] from the frames. For each predicate, we group all of its contextual arguments and construct only one additional example, in which all original labels remain (e.g. 'TMP', 'MNR'). This is unlike how we handle core arguments, whose labels (e.g. 'A0', 'A1') are reduced to 'A'. This example has the definition text "contextual". Hence, all contextual arguments of a predicate are predicted within one pass.

Missing Frames. Our ACD data format is contingent on each predicate sense having a corresponding frame file which contains argument label definitions. However, in CoNLL09, some frame files are missing for predicates present in the data. To account for this, we perform additional lookup in PropBank (Palmer et al., 2005) for verb predicates and NomBank (Meyers et al., 2004) and noun predicates. Even so, some predicate senses still do not have frames. For this, we default the possible argument set to be `A0` , `A1` , `A2` and `A3` , each with definitions "unknown". Since these missing frames are dataset noise and disadvantages models based on ACD data, we also report performance on a purged dataset (**p-CoNLL09**) where we remove the examples whose predicate senses do not have frames even after additional look-ups.[8]

4 Experiments and Results

We experiment with 2 settings, all using the RoBERTa-base model mentioned before. First, we consider a model trained and tested on the ordinary AC data and another on the ACD data. For the ACD model, we report performance both using gold predicate senses and using predicted senses in test time. Each model uses the default hyperparameters from HuggingFace Transformers (Wolf et al., 2019) without tuning. The best model on the development set is evaluated on the test set. For reproducibility details, see Appendix A.

[5] We have also tried explicitly encoding the predicate senses (e.g., work.01), and found it works worse empirically.

[6] Empirically, clashes are rare, and other resolution strategies make little difference to performance.

[7] We attempted to use approximate definitions from the annotation guidelines such as "time" or "manner", but found such data empirically led to worse performance.

[8] In the in-domain test set, 8 out of 10,498 predicate senses are removed. In out-domain, 67 out of 1259.

	In-domain	Out-domain
AC	89.9	83.5
ACD (pred. sense)	90.2	83.6
ACD (gold sense)	90.5	84.6

Table 4: Argument F1 on the p-CoNLL09 dataset.

Percentile	N/A	10%	20%	30%	40%
% examples	1.1%	2.9%	3.2%	3.5%	3.8%
AC	77.6	82.5	82.6	86.2	87.0
ACD (gold)	82.0	86.5	85.0	86.9	87.3
Δ	4.4	4.0	2.4	0.7	0.3
ACD (pred)	80.8	85.0	83.5	86.1	87.0
Δ	3.2	2.5	0.9	-0.1	0

Table 5: Argument F1 on subsets of CoNLL09 in-domain test set, bucketed by predicate sense frequency in the training set. The "N/A" column refers to test examples with predicates absent in the training set.

Model performances are shown in Table 3. Following Shi and Lin (2019), we report both combined performance of PSD and AC using the CoNLL09 official scoring script, and the micro-F1 of arguments only to disentangle the two tasks.

On the out-domain test set, our ACD model with gold sense outperforms current state-of-the-art by 1.1 argument F1, and the AC model by 1.6. With predicted sense of accuracy of 88.5% (see Table 2), our ACD model is on par with state-of-the-art while outperforming the AC model by 0.5 F1. On in-domain, the difference in argument F1 is less pronounced, but our ACD models have higher precision. On p-CoNLL09, where all predicate senses have definitions from corresponding frames, the advantage of ACD over AC is more pronounced in-domain and similar out-domain (Table 4). These observations show that label definitions help models transfer to another domain with different data and label distribution.With the definitions in ACD replaced by symbolic labels (e.g., `A1`), its performance drops to no better than AC, showing the benefit of the ACD model can be attributed to the definitions. In PropBank, each predicate has on average 2.1 core arguments, rendering our ACD model 2.1+1=3.1 times slower than the AC model.

5 Low-Resource Settings

We show that inductive biases such as label definitions benefit performance the most when training data is scarce. In SRL, such scenario is common in domains with jargon and in applications that require SRL with minimal supervision.

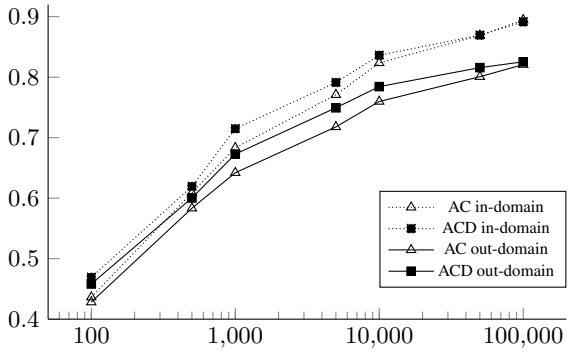

Figure 2: Argument F1 of AC and ACD models trained on varying amount of examples.

Low-Frequency Predicates. Previous work has found that SRL suffers from the long-tail phenomenon, where most predicates are rare words (Jindal et al., 2020). We experiment with disjoint subsets of the test data with predicate senses of different frequencies. In Table 5, ACD outperforms AC by up to 4.4 argument F1 for unseen predicates, notably helping with low-frequency predicates.

Few-Shot Learning. To simulate low-data scenarios, we train the AC and the ACD model with gold sense on varying amount of examples, randomly sampled for 5 runs. The average F1 is reported in Figure 2. Given up to 1,000 training examples, ACD outperforms AC by up to 3.2 F1 in- and out-domain, while the performance gap diminishes as training size approaches 100,000.

Distant Domain Adaptation. To see if definitions benefits adaptation to distant domains, we directly evaluate models trained on CoNLL09 (news articles) on the Biology PropBank (Chou et al., 2006), removing examples whose predicates do not have a frame. Our ACD model achieves 55.5 argument F1, outperforming AC which achieves 54.6, in line with our observation that definitions help with domain adaptation.

6 Conclusion and Future Work

We show that definitions of arguments advances state-of-the-art of semantic role labeling on CoNLL09, and even more notably in low-resource settings. The observed performance gap between ACD with gold and predicted sense suggests that a more competent PSD model is needed. Future work may also expand our approach to span-based SRL datasets, or multilingual settings.

References

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

Ulrike Baldewein, Katrin Erk, Sebastian Padó, and Detlef Prescher. 2004. Semantic role labelling with similarity-based generalization using EM-based clustering. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 64–68, Barcelona, Spain. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wen-Chi Chou, Richard Tzong-Han Tsai, Ying-Shan Su, Wei Ku, Ting-Yi Sung, and Wen-Lian Hsu. 2006. A semi-automatic method for annotating a biomedical Proposition Bank. In *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*, pages 5–12, Sydney, Australia. Association for Computational Linguistics.

Simone Conia, Andrea Bacciu, and Roberto Navigli. 2021. Unifying cross-lingual semantic role labeling with heterogeneous linguistic resources. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–351, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Hao Fei, Shengqiong Wu, Yafeng Ren, and Donghong Ji. 2021. Second-order semantic role labeling with global structural refinement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1966–1976.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson H S Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2017. A deep semantic natural language processing platform.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Ishan Jindal, Ranit Aharonov, Siddhartha Brahma, Huaiyu Zhu, and Yunyao Li. 2020. Improved semantic role labeling using parameterized neighborhood memory adaptation. *arXiv preprint arXiv:2011.14459*.

Richard Johansson. 2012. Non-atomic classification to improve a semantic role labeler for a low-resource language. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 95–99, Montréal, Canada. Association for Computational Linguistics.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime Carbonell, Noah A. Smith, and Chris Dyer. 2015. Frame-semantic role labeling with heterogeneous annotations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 218–224, Beijing, China. Association for Computational Linguistics.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Umar Maqsud, Sebastian Arnold, Michael Hülfenhaus, and Alan Akbik. 2014. Nerdle: Topic-specific question answering using wikia seeds. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 81–85.

Diego Marcheggiani and Ivan Titov. 2020. Graph convolutions over constituent trees for syntax-aware semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3915–3928, Online. Association for Computational Linguistics.

Yuichiroh Matsubayashi, Naoaki Okazaki, and Jun'ichi Tsujii. 2009. A comparative study on generalization of semantic roles in FrameNet. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 19–27, Suntec, Singapore. Association for Computational Linguistics.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Kashif Munir, Hai Zhao, and Zuchao Li. 2021. Adaptive convolution for semantic role labeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:782–791.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Chen Shi, Shujie Liu, Shuo Ren, Shi Feng, Mu Li, Ming Zhou, Xu Sun, and Houfeng Wang. 2016. Knowledge-based semantic embedding for machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2245–2254.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2020a. Unsupervised label-aware event trigger and argument classification. *CoRR*, abs/2012.15243.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1331–1340, Online. Association for Computational Linguistics.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. Intent detection with WikiHow. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 328–333, Suzhou, China. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020c. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.

A Modeling Details

All our models are implemented upon the HuggingFace Transformers library, which provides streamlined model sharing functions. We will upload our models as "model cards" upon paper acceptance.

Both our predicate sense disambiguation (PSD) model and our argument classification (AC) model are based on RoBERTa-base with default RoBERTa tokenizers. Our PSD model is implemented as a *RobertaForSequenceClassification* object[9], while

[9] https://huggingface.co/transformers/model_doc/roberta.html#robertaforsequenceclassification

our AC and ACD models are implemented as a *RobertaForTokenClassification* object[10]. Details of configurations and hyperparameters can be found in their documentations.

We run all experiments on NVIDIA V100 GPUs with 16G memory. During training, we save the model with the best argument F1 on the development set each 100 training steps. We run each experiment (e.g. training ACD on CoNLL09) for up to 48 hours, or when the best model is not updated for 5,000 training steps, whichever is sooner, before evaluating the best model on the test set.

B Qualitative Example

While we have not found convincing patterns of examples that benefit from ACD, we showcase one such example with unconventional syntax.

In the CoNLL09 out-domain test set, one sentence (abridged) is

> Any good decorator these days can [make] you a tasteful home.

with the word "make" as the predicate. Here, "make" has sense *create*, and thus "you" serves as an indirect object in a colloquial use, and the sentence is roughly equivalent to "...can make a tasteful home for you." This use case can easily be confused with the other sense of "make": *cause to be* (e.g. He makes me do all the work).

Predicate "make" with the correct sense "create" has 4 arguments from the frame file:

1. `A0` *creator*, annotated as "decorator";
2. `A1` *creation*, annotated as "home";
3. `A2` *created from*, annotated as none;
4. `A3` *benefactive*, annotated as "you".

With these definitions, the ACD model with gold sense correctly predicts all arguments, except missing `A3` . In contrast, given an incorrectly predicted of sense "cause to be" with the arguments from the frame file, the ACD model predicts:

1. `A0` *impeller to action* correctly as "decorator";
2. `A1` *impelled agent* incorrectly as "you";
3. `A2` *impelled action* incorrectly as "home";
4. `A3` , which is non-existent, incorrectly.

Identically, the AC model correctly predicts `A0` , which in most cases is the subject of the sentence, without much surprise. However, it incorrectly predicts `A1` as "you" and `A2` as "home". This example qualitatively provides evidence that with with definitions of arguments for the correct predicate sense, the model is better at performing SRL on underrepresented or complex examples.

C Other Formats of Injecting Definitions

We demonstrate previously that we append the definitions to the end of a sentence, maintaining the token classification format. We have also attempted other formats which empirically perform worse. For example, the sentence

> He [drills] three holes into the wall.

with the predicate "drill" with arguments `A0` *driller* as "He" and `A1` *thing drilled, gaining holes* as "wall", can be converted to the following formats.

Question answering. Similar to Du and Cardie (2020), in a classical SQuAD-style (Rajpurkar et al., 2016) format, the passage is the sentence. There are two questions: "What is the *driller* for 'drill'? with answer "He", and "What is the *thing drilled, gaining holes* for 'drill'? with answer "wall". In our experiments, we tried a variety of models such as RoBERTa, XLNet, etc. and were not able to have any of these converge on the training set.

Sentence completion. Similar to Schick and Schütze (2021), the input sentence with masked tokens is

> In the sentence "He drills three holes into the wall," the driller for "drill" is...

with the answer "He". The example for "thing drilled" is omitted. In our experiments, we tried a variety of models such as RoBERTa, XLNet, etc. and were not able to have any of these converge on the training set.

Prompting. Similar to Brown et al. (2020), we were also able to convert our examples to a task description and some examples, using the two formats above, as input to models such as GPT-3. While this maneuver has potential, we do not access to the closed beta of GPT-3, and were not able to perform the experiments.

[10] https://huggingface.co/transformers/model_doc/roberta.html#robertafortokenclassification

Percentile	N/A	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
% examples	1.1%	2.9%	3.2%	3.5%	3.8%	4.0%	5.2%	6.9%	9.2%	14.4%	45.7%
AC	77.6	82.5	82.6	86.2	87.0	87.5	90.4	89.2	88.9	90.6	91.2
ACD (gold)	82.0	86.5	85.0	86.9	87.3	89.9	90.1	89.2	88.8	89.7	90.9
ACD (pred)	80.8	85.0	83.5	86.1	87.0	89.7	89.8	89.1	88.5	89.5	90.8

Table 6: Argument F1 on subsets of CoNLL09 in-domain test set, bucketed by the percentile of predicate sense frequency in the training set. The "N/A" column refers to test examples with predicates absent in the training set.

Num. training examples	10^2	$10^{2.5}$	10^3	$10^{3.5}$	10^4	$10^{4.5}$	10^5
AC (in) mean	43.64	60.9	68.32	77.1	82.36	86.9	89.46
AC (in) SE	0.721526	0.258844	0.397995	0.246982	0.169115	0.070711	0.08124
ACD (in) mean	46.92	61.94	71.48	79.12	83.62	86.94	89.14
ACD (in) SE	0.921629	0.894763	0.349857	0.185472	0.115758	0.06	0.11225
AC (out) mean	42.82	58.34	64.2	71.78	75.98	80.08	82.08
AC (out) SE	0.938296	0.465403	0.564801	0.295635	0.558032	0.424735	0.243721
ACD (out) mean	45.82	60.04	67.28	74.94	78.44	81.58	82.54
ACD (out) SE	0.686586	0.612862	0.431741	0.304302	0.314006	0.341174	0.261916

Table 7: Mean and standard error of argument F1 over 5 runs of AC and ACD models trained on varying amount of randomly sampled examples, reported on CoNLL09 in- and out-domain test set.

D Multilingual Settings

We have also attempted leveraging argument definitions in non-English languages. Among the 6 languages present in CoNLL09, the frame files across them are structured very differently. We process those for Chinese whose frame files are formatted similar to those for English. Using a multilingual cased BERT (Devlin et al., 2019), we train AC and ACD models in the same fashion as English, and find that ACD underperforms AC on CoNLL09. Upon inspection, we find that argument labels for the Chinese frames are more terse and uninformative. For example, the definition "agent" and "entity" occupy more than 50% of all definition occurrences, corresponding to `A0` and `A1` most of the time. We hypothesize that these homogeneous definitions renders ACD performance lackluster.

We have also attempted a cross-lingual few-shot transfer setting, where a model is trained on the English training data with or without definition, and then continues to be trained on the Chinese training data without definition, before it is evaluated on the Chinese test set. We find that ACD "pre-training" also underperforms the AC counterpart.

E Risks and Biases

The potential risks and Biases of our work are minimal. Since we leverage the CoNLL09 the PropBank datasets, though unlikely to exist, unsafe and unfair texts or those containing person-identifying information in these human-curated datasets may propagate to the use of models trained on them.

F Licenses of Datasets Used

CoNLL09's licensing information cannot be found.
PropBank is licensed under CC BY-SA 4.0.
The domain-specific PropBanks by IBM are licensed under CDLA-Sharing-1.0.