Polyglot Contextual Representations Improve Crosslingual Transfer

Phoebe Mulcaire[♡] **Jungo Kasai**[♡] **Noah A. Smith**[♡♢]
[♡]Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA, USA
[♢]Allen Institute for Artificial Intelligence, Seattle, WA, USA
{pmulc,jkasai,nasmith}@cs.washington.edu

Abstract

We introduce Rosita, a method to produce multilingual contextual word representations by training a single language model on text from multiple languages. Our method combines the advantages of contextual word representations with those of multilingual representation learning. We produce language models from dissimilar language pairs (English/Arabic and English/Chinese) and use them in dependency parsing, semantic role labeling, and named entity recognition, with comparisons to monolingual and non-contextual variants. Our results provide further evidence for the benefits of polyglot learning, in which representations are shared across multiple languages.

1 Introduction

State-of-the-art methods for crosslingual transfer make use of multilingual word embeddings, and much research has explored methods that align vector spaces for words in different languages (Faruqui and Dyer, 2014; Upadhyay et al., 2016; Ruder et al., 2017). On the other hand, *contextual* word representations (CWR) extracted from language models (LMs) have advanced the state of the art beyond what was achieved with word type representations on many monolingual NLP tasks (Peters et al., 2018). Thus, the question arises: can *contextual* word representations benefit from *multilinguality*?

We introduce a method to produce multilingual CWR by training a single "polyglot" language model on text in multiple languages. As our work is a multilingual extension of ELMo (Peters et al., 2018), we call it Rosita (after a bilingual character from *Sesame Street*). Our hypothesis is that, although each language is unique, different languages manifest similar characteristics (e.g., morphological, lexical, syntactic) which can be exploited by training a single model with data from multiple languages (Ammar, 2016). Previous work has shown this to be true to some degree in the context of syntactic dependency parsing (Ammar et al., 2016), semantic role labeling (Mulcaire et al., 2018), named entity recognition (Xie et al., 2018), and language modeling for phonetic sequences (Tsvetkov et al., 2016) and for speech recognition (Ragni et al., 2016). Recently, de Lhoneux et al. (2018) showed that parameter sharing between languages can improve performance in dependency parsing, but the effect is variable, depending on the language pair and the parameter sharing strategy. Other recent work also reported that concatenating data from different languages can hurt performance in dependency parsing (Che et al., 2018). These mixed results suggest that while crosslingual transfer in neural network models is a promising direction, the best blend of polyglot and language-specific elements may depend on the task and architecture. However, we find overall contextual representations from polyglot language models succeed in a range of settings, even where multilingual word type embeddings do not, and are a useful technique for crosslingual transfer.

We explore crosslingual transfer between highly dissimilar languages (English→Chinese and English→Arabic) for three core tasks: Universal Dependency (UD) parsing, semantic role labeling (SRL), and named entity recognition (NER). We provide some of the first work using polyglot LMs to produce contextual representations,[1] and the first analysis comparing them to monolingual LMs for this purpose. We also introduce an LM variant which takes multilingual word embedding input as well as character input, and explore its

[1]Contemporaneous work uses polyglot LMs for natural language inference and machine translation (Lample and Conneau, 2019).

Proceedings of NAACL-HLT 2019, pages 3912–3918
Minneapolis, Minnesota, June 2 - June 7, 2019.

applicability for producing contextual word representations. Our experiments focus on comparisons in three dimensions: monolingual vs. polyglot representations, contextual vs. word type embeddings, and, within the contextual representation paradigm, purely character-based language models vs. ones that include word-level input.

Previous work has shown that contextual representations offer a significant advantage over traditional word embeddings (word type representations). In this work, we show that, on these tasks, polyglot character-based language models can provide benefits on top of those offered by contextualization. Specifically, even when crosslingual transfer with word type embeddings hurts target language performance relative to monolingual models, polyglot *contextual* representations can improve target language performance relative to monolingual versions, suggesting that polyglot language models tie dissimilar languages in an effective way.

In this paper, we use the following terms: *crosslingual transfer* and *polyglot learning*. While crosslingual transfer is often used in situations where target data are absent or scarce, we use it broadly to mean any method which uses one or more source languages to help process another target language. We also draw a sharp distinction between multilingual and polyglot models. Multilingual learning can happen independently for different languages, but a polyglot solution provides a single model for multiple languages, e.g., by parameter sharing between languages in networks during training.

2 Polyglot Language Models

We first describe the language models we use to construct multilingual (and monolingual) CWR.

2.1 Data and Preprocessing

Because the Universal Dependencies treebanks we use for the parsing task predominantly use Traditional Chinese characters and the Ontonotes data for SRL and NER consist of Simplified Chinese, we train separate language models for the two variants. For English we use text from the Billion Word Benchmark (Chelba et al., 2013), for Traditional Chinese, wiki and web data provided for the CoNLL 2017 Shared Task (Ginter et al., 2017), for Simplified Chinese, newswire text from Xinhua,[2] and for Arabic, newswire text from AFP.[3] We use approximately 60 million tokens of news and web text for each language.

We tokenized the language model training data for English and Simplified Chinese using Stanford CoreNLP (Manning et al., 2014). The Traditional Chinese corpus was already pre-segmented by UDPipe (Ginter et al., 2017; Straka et al., 2016). We found that the Arabic vocabulary from AFP matched both the UD and Ontonotes data reasonably well without additional tokenization. We also processed all corpora to normalize punctuation and remove non-text.

2.2 Models and Training

We base our language models on the ELMo method (Peters et al., 2018), which encodes each word with a character CNN, then processes the word in context with a word-level LSTM.[4] Following Che et al. (2018), who used 20 million words per language to train monolingual language models for many languages, we use the same hyperparameters used to train the monolingual English language model from Peters et al. (2018), except that we reduce the internal LSTM dimension from 4096 to 2048.

For each target language dataset (Traditional Chinese, Simplified Chinese, and Arabic), we produce:

- a monolingual language model with character CNN (MONOCHAR) trained on that language's data;
- a polyglot LM (ROSITACHAR) trained with the same code, on that language's data with an additional, equal amount of English data;
- a modified polyglot LM (ROSITAWORD), described below.

The ROSITAWORD model concatenates a 300 dimensional word type embedding, initialized with multilingual word embeddings, to the character CNN encoding of the word, before passing this combined vector to the bidirectional LSTM.

[2] catalog.ldc.upenn.edu/LDC95T13

[3] catalog.ldc.upenn.edu/LDC2001T55

[4] A possible alternative is BERT (Devlin et al., 2018), which uses a bidirectional objective and a transformer architecture in place of the LSTM. Notably, one of the provided BERT models was trained on several languages in combination, in a simple polyglot approach (see https://github.com/google-research/bert/blob/master/multilingual.md). Our initial exploration of multilingual BERT models raised sufficient questions about preprocessing that we defer exploration to future work.

The idea of this word-level initialization is to bias the model toward crosslingual sharing; because words with similar meanings have similar representations, the features that the model learns are expected to be at least partially language-agnostic. The word type embeddings used for these models, as well as elsewhere in the paper, are trained on our language model training set using the fastText method (Bojanowski et al., 2017), and target language vectors are aligned with the English ones using supervised MUSE[5] (Conneau et al., 2018). See appendix for more LM training details.

3 Experiments

All of our task models (UD, SRL, and NER) are implemented in AllenNLP, version 0.7.2 (Gardner et al., 2018).[6] We generally follow the default hyperparameters and training schemes provided in the AllenNLP library regardless of language. See appendix for the complete list of our hyperparameters. For each task, we experiment with five types of word representations: in addition to the three language model types (MONOCHAR, ROSITACHAR, and ROSITAWORD) described above, we show results for the task models trained with monolingual and polyglot non-contextual word embeddings.

After pretraining, the word representations are fine-tuned to the specific task during task training. In non-contextual cases, we fine-tune by updating word embeddings directly, while in contextual cases, we only update coefficients for a linear combination of the internal representation layers for efficiency (Peters et al., 2018). In order to properly evaluate our models' generalization ability, we ensure that sentences in the test data are excluded from the data used to train the language models.

3.1 Universal Dependency Parsing

We use a state-of-the-art graph-based dependency parser with BiLSTM and biaffine attention (Dozat and Manning, 2017). Specifically, the parser takes as input word representations and 100-dimensional fine-grained POS embeddings following Dozat and Manning (2017). We use the same UD treebanks and train/dev./test splits as the CoNLL 2018 shared task on multilingual dependency parsing (Zeman et al., 2018). In particular, we use the GUM treebank for English,[7] GSD for Chinese, and PADT for Arabic. For training and validation, we use the provided gold POS tags and word segmentation.

For each configuration, we run experiments five times with random initializations and report the mean and standard deviation. For testing, we use the CoNLL 2018 evaluation script and consider two scenarios: (1) gold POS tags and word segmentations and (2) predicted POS tags and word segmentations from the system outputs of Che et al. (2018) and Qi et al. (2018).[8] The former scenario enables us to purely assess parsing performance; see column 3 in Table 1 for these results on Chinese and Arabic. The latter allows for a direct comparison to the best previously reported parsers (Chinese, Che et al., 2018; Arabic, Qi et al., 2018). See Table 2 for these results.

As seen in Table 1, the Universal Dependencies results generally show a significant improvement from the use of CWR. The best results for both languages come from the ROSITACHAR LM and polyglot task models, showing that polyglot training helps, but that the word-embedding initialization of the ROSITAWORD model does not necessarily lead to a better final model. The results also suggest that combining ROSITACHAR LM and polyglot task training is key to improve parsing performance. Table 2 shows that we outperform the state-of-the-art systems from the shared task competition. In particular, our LMs even outperform the Harbin system, which uses monolingual CWR and an ensemble of three biaffine parsers.

3.2 Semantic Role Labeling

We use a strong existing model based on BIO tagging on top of a deep interleaving BiLSTM with highway connections (He et al., 2017). The SRL model takes as input word representations and 100-dimensional predicate indicator embeddings following He et al. (2017). We use a standard PropBank-style, span-based SRL dataset for English, Chinese, and Arabic: Ontonotes (Pradhan et al., 2013). Note that Ontonotes provides annotations using a single shared annotation scheme for

[5] For our English/Chinese and English/Arabic data, their unsupervised method yielded substantially worse results in word translation.

[6] We make our multilingual fork available at https://github.com/pmulcaire/rosita

[7] While there are several UD English corpora, we choose the GUM corpus to minimize domain mismatch.

[8] System outputs for all systems are available at https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2885

vectors (lang.)	task lang.	UD LAS	SRL F_1	NER F_1
fastText (CMN)	CMN	$85.15_{\pm 0.12}$	69.79	76.31
fastText (CMN+ENG)	CMN+ENG	$84.92_{\pm 0.28}$	70.82	76.05
MONOCHAR (CMN)	CMN	$87.55_{\pm 0.25}$	74.14	78.18
ROSITACHAR (CMN+ENG)	CMN	$87.16_{\pm 0.08}$	74.24	**78.29**
ROSITACHAR (CMN+ENG)	CMN+ENG	$\mathbf{87.75}_{\pm 0.16}$	74.69	77.68
ROSITAWORD (CMN+ENG)	CMN	$86.50_{\pm 0.17}$	**74.84**	77.19
ROSITAWORD (CMN+ENG)	CMN+ENG	$86.37_{\pm 0.35}$	74.69	77.16
Best prior work	CMN	–	62.83	75.63
fastText (ARA)	ARA	$82.58_{\pm 0.51}$	50.50	71.60
fastText (ARA+ENG)	ARA+ENG	$82.67_{\pm 0.46}$	54.82	71.45
MONOCHAR (ARA)	ARA	$84.98_{\pm 0.18}$	**59.55**	75.02
ROSITACHAR (ARA+ENG)	ARA	$84.98_{\pm 0.12}$	58.69	75.56
ROSITACHAR (ARA+ENG)	ARA+ENG	$\mathbf{85.24}_{\pm 0.13}$	59.29	**76.19**
ROSITAWORD (ARA+ENG)	ARA	$84.34_{\pm 0.20}$	58.34	74.02
ROSITAWORD (ARA+ENG)	ARA+ENG	$84.24_{\pm 0.13}$	59.47	72.79
Best prior work	ARA	–	48.68	68.02

Table 1: LAS for UD parsing, F_1 for SRL, and F_1 for NER, with different input representations. For UD, each number is an average over five runs with different initialization, with standard deviation. SRL/NER results are from one run. The "task lang." column indicates whether the UD/SRL/NER model was trained on annotated text in the target language alone, or a blend of English and the target language data. ROSITAWORD LMs use as word-level input the same multilingual word vectors as fastText models. The best prior result for Ontonotes Chinese NER is in Shen et al. (2018); the others are from Pradhan et al. (2013).

LM type	task lang.	LAS
Harbin (Che et al., 2018)	CMN	76.77
Harbin (non-ensemble)	CMN	75.55
ROSITACHAR	CMN	77.40
ROSITACHAR	CMN+ENG	**77.63**
Stanford (Qi et al., 2018)	ARA	77.06
ROSITACHAR	ARA	77.79
ROSITACHAR	ARA+ENG	**78.02**

Table 2: LAS (F_1) comparison to the winning systems for each language in the CoNLL 2018 shared task for UD. We use predicted POS and the segmentation of the winning system for that language. The ROSITACHAR LM variant was selected based on development performance in the gold-segmentation condition.

English, Chinese, and Arabic, which can facilitate crosslingual transfer. For Chinese and English we simply use the provided surface form of the words. The Arabic text in Ontonotes has diacritics to indicate vocalization which do not appear (or only infrequently) in the original source or in our language modeling data. We remove these for better consistency with the language model vocabulary. We use gold predicates and the CoNLL 2005 evaluation script for the experiments below to ensure our results are comparable to prior work. See column 4 in Table 1 for results on the CoNLL-2012 Chinese and Arabic test sets.

The SRL results confirm the advantage of CWR. Unlike the other two tasks, multilingual word type embeddings are better than monolingual versions in SRL. Perhaps relatedly, models using ROSITAWORD are more successful here, providing the highest performance on Chinese. One unusual result is that the model using the MONOCHAR LM is most successful for Arabic. This may be linked to the poor results on Arabic SRL overall, which are likely due to the much smaller size of the corpus compared to Chinese (less than 20% as many annotated predicates) and higher proportion of language-specific tags. Such language-specific tags in Arabic could limit the effectiveness of shared English-Arabic representations. Still, polyglot methods' performance is only slightly behind.

3.3 Named Entity Recognition

We use the state-of-the-art BiLSTM-CRF NER model with the BIO tagging scheme (Peters et al., 2017). The network takes as input word representations and 128-dimensional character-level embeddings from a character LSTM. We again use the Ontonotes dataset with the standard data

splits. See the last column in Table 1 for results on the CoNLL-2012 Chinese and Arabic test sets. As with most other experiments, the NER results show a strong advantage from the use of contextual representations and a smaller additional advantage from those produced by polyglot LMs.

4 Discussion

Overall, our results show that polyglot language models produce very useful representations. While Universal Dependency parsing, Arabic SRL, and Chinese NER show models using contextual representations outperform those using word type representations, the advantage from polyglot training in some cases is minor. However, Chinese SRL and Arabic NER show strong improvement both from contextual word representations and from polyglot training. Thus, while the benefit of crosslingual transfer appears to be somewhat variable and task dependent, polyglot training is helpful overall for contextual word representations. Notably, the RositaChar LM does not involve any direct supervision of tying two languages together, such as bilingual dictionaries or parallel corpora, yet is still most often able to learn the most effective representations. One explanation is that it automatically learns crosslingual connections from unlabeled data alone. Another possibility, though, is that the additional data provided in polyglot training produces a useful regularization effect, improving the target language representations without crosslingual sharing (except that induced by shared vocabulary, e.g., borrowings, numbers, or punctuation). Nevertheless, the success of polyglot language models is worth further study.

5 Conclusion

We presented a method for using polyglot language models to produce multilingual, contextual word representations, and demonstrated their benefits, producing state-of-the-art results in multiple tasks. These results provide a foundation for further study of polyglot language models and their use as unsupervised components of multilingual models.

Acknowledgments

The authors thank Mark Neumann for assistance with the AllenNLP library and the anonymous reviewers for their helpful feedback. This research was funded in part by NSF grant IIS-1562364, the Funai Overseas Scholarship to JK, and the NVIDIA Corporation through the donation of a GeForce GPU.

References

Waleed Ammar. 2016. *Towards a Universal Analyzer of Natural Languages*. Ph.D. thesis, Carnegie Mellon University.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *TACL*, 4:431–444.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proc. of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. arXiv:1312.3005.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proc. of ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Timothy Dozat and Christopher Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proc. of ICLR*.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proc. of EACL*, pages 462–471.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. arXiv:1803.07640.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and

word embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proc. of ACL*, pages 473–483.

Diederik P. Kingma and Jimmy Lei Ba. 2015. ADAM: A method for stochastic optimization. In *Proc. of ICLR*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. arxiv:1901.07291.

Miryam de Lhoneux, Johannes Bjerva, Isabelle Augenstein, and Anders Sgaard. 2018. Parameter sharing between dependency parsers for related languages. In *Proc. of EMNLP*, pages 4992–4997.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proc. of ACL, System Demonstrations*, pages 55–60.

Phoebe Mulcaire, Swabha Swayamdipta, and Noah A Smith. 2018. Polyglot semantic role labeling. In *Proc. of ACL*, volume 2, pages 667–672.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proc. of ACL*, pages 1756–1765.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL-HLT*, pages 2227–2237.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proc. of CoNLL*, pages 143–152.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proc. of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170.

Anton Ragni, Edgar Dakin, Xie Chen, Mark J. F. Gales, and Kate Knill. 2016. Multi-language neural network language models. In *Proc. of INTERSPEECH*, volume 08-12-September-2016, pages 3042–3046.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2017. A survey of cross-lingual word embedding models. arXiv:1706.04902.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2018. Deep active learning for named entity recognition. In *Proc. ICLR*.

Milan Straka, Jan Hajic, and Jana Strakov. 2016. UDPipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proc. of LREC*, pages 4290–4297.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proc. of NAACL-HLT*, pages 1357–1366.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proc. of ACL*, pages 1661–1670.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proc. of EMNLP*, pages 369–379.

Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. arxiv:1212.5701.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proc. of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21.

A Supplementary Material

In this supplementary material, we provide hyperparameters used in our models for easy replication of our results.

A.1 Language Models

Character CNNs	
Char embedding size	16
(# Window Size, # Filters)	(1, 32), (2, 32), (3, 68), (4, 128), (5, 256), 6, 512), (7, 1024)
Activation	Relu
Word-level LSTM	
LSTM size	2048
# LSTM layers	2
LSTM projection size	256
Use skip connections	Yes
Inter-layer dropout rate	0.1
Training	
Batch size	128
Unroll steps (Window Size)	20
# Negative samples	64
# Epochs	10
Adagrad (Duchi et al., 2011) lrate	0.2
Adagrad initial accumulator value	1.0

Table 3: Language Model Hyperparameters.

Seen in Table 3 is a list of hyperparameters for our language models. We generally follow Peters et al. (2018) and use their publicly available code for training.[9] For character only models, we halve the LSTM and projection sizes to expedite training and to compensate for the greatly reduced training data—their hyperparameters were tuned on around 30M sentences, while we used less than 3M sentences (60-70M tokens) per language.

A.2 UD Parsing

For UD parsing, we generally follow the hyperparameters provided in AllenNLP (Gardner et al., 2018). See a list of hyperparameters in Table 4.

A.3 Semantic Role Labeling

For SRL, we again follow the hyperparameters given in AllenNLP (Table 5). The one exception is that we used 4 layers of alternating BiLSTMs instead of 8 layers to expedite the training process.

A.4 Named Entity Recognition

We again use the hyperparameter configurations provided in AllenNLP. See Table 6 for details.

[9] https://github.com/allenai/bilm-tf

Input	
POS embedding size	100
Input dropout rate	0.3
Word-level BiLSTM	
LSTM size	400
# LSTM layers	3
Recurrent dropout rate	0.3
Inter-layer dropout rate	0.3
Use Highway Connection	Yes
Multilayer Perceptron, Attention	
Arc MLP size	500
Label MLP size	100
# MLP layers	1
Activation	Relu
Training	
Batch size	80
# Epochs	80
Early stopping	50
Adam (Kingma and Ba, 2015) lrate	0.001
Adam β_1	0.9
Adam β_2	0.999

Table 4: UD Parsing Hyperparameters.

Input	
Predicate indicator embedding size	100
Word-level Alternating BiLSTM	
LSTM size	300
# LSTM layers	4
Recurrent dropout rate	0.1
Use Highway Connection	Yes
Training	
Batch size	80
# Epochs	80
Early stopping	20
Adadelta (Zeiler, 2012) lrate	0.1
Adadelta ρ	0.95
Gradient clipping	1.0

Table 5: SRL Hyperparameters.

Char-level LSTM	
Char embedding size	25
Input dropout rate	0.5
LSTM size	128
# LSTM layers	1
Word-level BiLSTM	
LSTM size	200
# LSTM layers	3
Inter-layer dropout rate	0.5
Recurrent dropout rate	0.5
Use Highway Connection	Yes
Multilayer Perceptron	
MLP size	400
Activation	tanh
Training	
Batch size	64
# Epochs	50
Early stopping	25
Adam (Kingma and Ba, 2015) lrate	0.001
Adam β_1	0.9
Adam β_2	0.999
L2 regularization coefficient	0.001

Table 6: NER Hyperparameters.