Appendices

A Experimental Setup

We implemented all neural models in PyTorch,[20] and the **Hard** baseline in scikit-learn (Pedregosa et al., 2011).[21] We train using Adam (Kingma and Ba, 2015) with a batch size of 150. We use 300-dimensional GloVe 840B embeddings (Pennington et al., 2014) normalized to unit length. We randomly initialize all other parameters. Our MLP has two layers. For regularization, we use dropout.[22]

In all cases, we tune the hyperparameters of our model on the development set by running 30 iterations of random search. The full list of hyperparameters explored for each model can be found in Table 4. Finally, we train all models for 250 epochs, stopping early if development loss does not improve for 30 epochs.

Type	Values	Models
Patterns	{5:10,4:10,3:10,2:10}, {6:10,5:10,4:10}, {6:10,5:10,4:10,3:10,2:10}, {6:20,5:20,4:10,3:10,2:10}, {7:10,6:10,5:10,4:10,3:10,2:10}	**SoPa**
Learning rate	0.01, 0.05, 0.001, 0.005	**SoPa, DAN, BiLSTM, CNN**
Dropout	0, 0.05, 0.1, 0.2	**SoPa, BiLSTM, CNN**
MLP hid. dim.	10, 25, 50, 100, 300	**SoPa, DAN, BiLSTM, CNN**
Hid. layer dim.	100, 200, 300	**BiLSTM**
Out. layer dim.	50, 100, 200	**CNN**
Window size	4, 5, 6	**CNN**
Word dropout	0.1, 0.2, 0.3, 0.4	**DAN**
Log. reg. param	1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001	**Hard**
Min. pattern freq.	2–10, 0.1%	**Hard**

Table 4: The hyperparameters explored in our experiments. **Patterns**: the number of patterns of each length. For example, {5:20,4:10} means 20 patterns of length 5 and 10 patterns of length 4. **MLP hid. dim.**: the dimension of the hidden layer of the MLP. **Hid. layer dim.**: the BiLSTM hidden layer dimension. **Out. layer dim.**: the CNN output layer dimension. **Window size**: the CNN window size. **Log. reg. param**: the logistic regression regularization parameter. **Min. pattern freq.**: minimum frequency for a pattern to be included as a logistic regression feature, expressed either as absolute count or as relative frequency in the train set. **Models**: the models to which each hyperparameter applies (see Section 5).

[20] https://pytorch.org/

[21] http://scikit-learn.org/

[22] **DAN** uses word dropout instead of regular dropout as its only learnable parameters are the MLP layer weights.