A Implementation Details

We implemented baseline methods in Python and tuned hyper-parameters of models by grid search.

A.1 SVM

We used the LinearSVC implementation in scikit-learn (v0.20.2). We tested the following hyper-parameter grid on validation splits.

- Regularization: \{L1, L2\}
- Loss function: \{Hinge loss, Squared hinge loss\}
- Penalty parameter: \{2^{-3}, 2^{-2}, 2^{-1}, 0\}

A.2 MLP

We used allennlp (v0.8.2) (Gardner et al., 2017) with pytorch (v1.0.1) for MLP and encoders. For all encoders, we initialized an embedding layer by 100-D pre-trained GloVe embeddings (Pennington et al., 2014) and fixed the values throughout training. We also tried other embedding algorithms, which yielded similar results.

The details of hyper-parameter settings of encoders are the following:

**SWEM:** No tunable parameters

**CNN:**
- The sizes of windows: (3, 4, 5) *fixed
- The number of filters: \{50, 100\}

**BiLSTM:**
- The number of layers: 2 *fixed
- The number of hidden units: \{50, 100\}

The hyper-parameters of MLP is below:

- The number of hidden layers: 1 *fixed
- The number of hidden units: \{50, 100, 200\}
- Activation function: \{ReLU, tanh\}
- Dropout: 0.5 *fixed

We used the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.001 to optimize parameters. The weight decay parameter (L2 regularization) were chosen from \{0, 0.1\}.

Finally, we tuned out-domain discounting factor \(\lambda\) from \{2^{-3}, 2^{-2}, 2^{-1}\}.

References

