A Supplemental Material to accompany *Deep contextualized word representations*

This supplement contains details of the model architectures, training routines and hyper-parameter choices for the state-of-the-art models in Section 4.

All of the individual models share a common architecture in the lowest layers with a context independent token representation below several layers of stacked RNNs – LSTMs in every case except the SQuAD model that uses GRUs.

A.1 Fine tuning biLM

As noted in Sec. 3.4, fine tuning the biLM on task specific data typically resulted in significant drops in perplexity. To fine tune on a given task, the supervised labels were temporarily ignored, the biLM fine tuned for one epoch on the training split and evaluated on the development split. Once fine tuned, the biLM weights were fixed during task training.

Table 8 lists the development set perplexities for the considered tasks. In every case except CoNLL 2012, fine tuning results in a large improvement in perplexity, e.g., from 72.1 to 16.8 for SNLI.

The impact of fine tuning on supervised performance is task dependent. In the case of SNLI, fine tuning the biLM increased development accuracy 0.6% from 88.9% to 89.5% for our single best model. However, for sentiment classification development set accuracy is approximately the same regardless whether a fine tuned biLM was used.

A.2 Importance of γ in Eqn. (1)

The γ parameter in Eqn. (1) was of practical importance to aid optimization, due to the different distributions between the biLM internal representations and the task specific representations. It is especially important in the last-only case in Sec. 5.1. Without this parameter, the last-only case performed poorly (well below the baseline) for SNLI and training failed completely for SRL.

A.3 Textual Entailment

Our baseline SNLI model is the ESIM sequence model from Chen et al. (2017). Following the original implementation, we used 300 dimensions for all LSTM and feed forward layers and pre-trained 300 dimensional GloVe embeddings that were fixed during training. For regularization, we added 50% variational dropout (Gal and Ghahramani, 2016) to the input of each LSTM layer and 50% dropout (Srivastava et al., 2014) at the input to the final two fully connected layers. All feed forward layers use ReLU activations. Parameters were optimized using Adam (Kingma and Ba, 2015) with gradient norms clipped at 5.0 and initial learning rate 0.0004, decreasing by half each time accuracy on the development set did not increase in subsequent epochs. The batch size was 32.

Dataset		Before tuning	After tuning
SNLI		72.1	16.8
CoNLL 2012 (coref/SRL)		92.3	-
CoNLL 2003 (NER)		103.2	46.3
SQuAD	Context	99.1	43.5
	Questions	158.2	52.0
SST		131.5	78.6

Table 8: Development set perplexity before and after fine tuning for one epoch on the training set for various datasets (lower is better). Reported values are the average of the forward and backward perplexities.

The best ELMo configuration added ELMo vectors to both the input and output of the lowest layer LSTM, using (1) with layer normalization and $\lambda = 0.001$. Due to the increased number of parameters in the ELMo model, we added ℓ^2 regularization with regularization coefficient 0.0001 to all recurrent and feed forward weight matrices and 50% dropout after the attention layer.

Table 9 compares test set accuracy of our system to previously published systems. Overall, adding ELMo to the ESIM model improved accuracy by 0.7% establishing a new single model state-of-the-art of 88.7%, and a five member ensemble pushes the overall accuracy to 89.3%.

A.4 Question Answering

Our QA model is a simplified version of the model from Clark and Gardner (2017). It embeds tokens by concatenating each token's case-sensitive 300 dimensional GloVe word vector (Pennington et al., 2014) with a character-derived embedding produced using a convolutional neural network followed by max-pooling on learned character embeddings. The token embeddings are passed through a shared bi-directional GRU, and then the bi-directional attention mechanism from BiDAF (Seo et al., 2017). The augmented con-

Model	Acc.
Feature based (Bowman et al., 2015)	78.2
DIIN (Gong et al., 2018)	88.0
BCN+Char+CoVe (McCann et al., 2017)	88.1
ESIM (Chen et al., 2017)	88.0
ESIM+TreeLSTM (Chen et al., 2017)	88.6
ESIM+ELMo	**88.7** $\pm$ 0.17
DIIN ensemble (Gong et al., 2018)	88.9
ESIM+ELMo ensemble	**89.3**

Table 9: SNLI test set accuracy.[3] Single model results occupy the portion, with ensemble results at the bottom.

text vectors are then passed through a linear layer with ReLU activations, a residual self-attention layer that uses a GRU followed by the same attention mechanism applied context-to-context, and another linear layer with ReLU activations. Finally, the results are fed through linear layers to predict the start and end token of the answer.

Variational dropout is used before the input to the GRUs and the linear layers at a rate of 0.2. A dimensionality of 90 is used for the GRUs, and 180 for the linear layers. We optimize the model using Adadelta with a batch size of 45. At test time we use an exponential moving average of the weights and limit the output span to be of at most size 17. We do not update the word vectors during training.

Performance was highest when adding ELMo without layer normalization to both the input and output of the contextual GRU layer and leaving the ELMo weights unregularized ($\lambda = 0$).

Table 10 compares test set results from the SQuAD leaderboard as of November 17, 2017 when we submitted our system. Overall, our submission had the highest single model and ensemble results, improving the previous single model result (SAN) by 1.4% F_1 and our baseline by 4.2%. A 11 member ensemble pushes F_1 to 87.4%, 1.0% increase over the previous ensemble best.

A.5 Semantic Role Labeling

Our baseline SRL model is an exact reimplementation of (He et al., 2017). Words are represented using a concatenation of 100 dimensional vector representations, initialized using GloVe (Pennington et al., 2014) and a binary, per-word predicate feature, represented using an 100 dimensional embedding. This 200 dimensional token representation is then passed through an 8 layer "interleaved" biLSTM with a 300 dimensional hidden size, in which the directions of the LSTM layers alternate per layer. This deep LSTM uses Highway connections (Srivastava et al., 2015) between layers and variational recurrent dropout (Gal and Ghahramani, 2016). This deep representation is then projected using a final dense layer followed by a softmax activation to form a distribution over all possible tags. Labels consist of semantic roles from PropBank (Palmer et al., 2005) augmented with a BIO labeling scheme to represent argument spans. During training, we minimize the negative log likelihood of the tag sequence using Adadelta with a learning rate of 1.0 and $\rho = 0.95$ (Zeiler, 2012). At test time, we perform Viterbi decoding to enforce valid spans using BIO constraints. Variational dropout of 10% is added to all LSTM hidden layers. Gradients are clipped if their value exceeds 1.0. Models are trained for 500 epochs or until validation F1 does not improve for 200 epochs, whichever is sooner. The pretrained GloVe vectors are fine-tuned during training. The final dense layer and all cells of all LSTMs are initialized to be orthogonal. The forget gate bias is initialized to 1 for all LSTMs, with all other gates initialized to 0, as per (Józefowicz et al., 2015).

Table 11 compares test set F1 scores of our ELMo augmented implementation of (He et al., 2017) with previous results. Our single model score of 84.6 F1 represents a new state-of-the-art result on the CONLL 2012 Semantic Role Labeling task, surpassing the previous single model result by 2.9 F1 and a 5-model ensemble by 1.2 F1.

A.6 Coreference resolution

Our baseline coreference model is the end-to-end neural model from Lee et al. (2017) with all hy-

[3] A comprehensive comparison can be found at https://nlp.stanford.edu/projects/snli/

Model	EM	F_1
BiDAF (Seo et al., 2017)	68.0	77.3
BiDAF + Self Attention	72.1	81.1
DCN+	75.1	83.1
Reg-RaSoR	75.8	83.3
FusionNet	76.0	83.9
r-net (Wang et al., 2017)	76.5	84.3
SAN (Liu et al., 2017)	76.8	84.4
BiDAF + Self Attention + ELMo	**78.6**	**85.8**
DCN+ Ensemble	78.9	86.0
FusionNet Ensemble	79.0	86.0
Interactive AoA Reader+ Ensemble	79.1	86.5
BiDAF + Self Attention + ELMo Ensemble	**81.0**	**87.4**

Table 10: Test set results for SQuAD, showing both Exact Match (EM) and F_1. The top half of the table contains single model results with ensembles at the bottom. References provided where available.

Model	F_1
Pradhan et al. (2013)	77.5
Zhou and Xu (2015)	81.3
He et al. (2017), single	81.7
He et al. (2017), ensemble	83.4
He et al. (2017), our impl.	81.4
He et al. (2017) + ELMo	**84.6**

Table 11: SRL CoNLL 2012 test set F_1.

Model	Average F_1
Durrett and Klein (2013)	60.3
Wiseman et al. (2016)	64.2
Clark and Manning (2016)	65.7
Lee et al. (2017) (single)	67.2
Lee et al. (2017) (ensemble)	68.8
Lee et al. (2017) + ELMo	**70.4**

Table 12: Coreference resolution average F_1 on the test set from the CoNLL 2012 shared task.

perparameters exactly following the original implementation.

The best configuration added ELMo to the input of the lowest layer biLSTM and weighted the biLM layers using (1) without any regularization ($\lambda = 0$) or layer normalization. 50% dropout was added to the ELMo representations.

Table 12 compares our results with previously published results. Overall, we improve the single model state-of-the-art by 3.2% average F_1, and our single model result improves the previous ensemble best by 1.6% F_1. Adding ELMo to the output from the biLSTM in addition to the biLSTM input reduced F_1 by approximately 0.7% (not shown).

A.7 Named Entity Recognition

Our baseline NER model concatenates 50 dimensional pre-trained Senna vectors (Collobert et al., 2011) with a CNN character based representation. The character representation uses 16 dimensional character embeddings and 128 convolutional filters of width three characters, a ReLU activation and by max pooling. The token representation is passed through two biLSTM layers, the first with 200 hidden units and the second with 100 hidden units before a final dense layer and softmax layer. During training, we use a CRF loss and at test time perform decoding using the Viterbi algorithm while ensuring that the output tag sequence is valid.

Variational dropout is added to the input of both biLSTM layers. During training the gradients are rescaled if their ℓ^2 norm exceeds 5.0 and parameters updated using Adam with constant learning rate of 0.001. The pre-trained Senna embeddings are fine tuned during training. We employ early stopping on the development set and report the averaged test set score across five runs with different random seeds.

ELMo was added to the input of the lowest layer task biLSTM. As the CoNLL 2003 NER data set is relatively small, we found the best performance by constraining the trainable layer weights to be effectively constant by setting $\lambda = 0.1$ with (1).

Table 13 compares test set F_1 scores of our ELMo enhanced biLSTM-CRF tagger with previous results. Overall, the 92.22% F_1 from our system establishes a new state-of-the-art. When compared to Peters et al. (2017), using representations

Model	$F_1 \pm$ std.
Collobert et al. (2011)♣	89.59
Lample et al. (2016)	90.94
Ma and Hovy (2016)	91.2
Chiu and Nichols (2016)♣,♢	91.62 ± 0.33
Peters et al. (2017)♢	91.93 ± 0.19
biLSTM-CRF + ELMo	$\mathbf{92.22 \pm 0.10}$

Table 13: Test set F_1 for CoNLL 2003 NER task. Models with ♣ included gazetteers and those with ♢ used both the train and development splits for training.

Model	Acc.
DMN (Kumar et al., 2016)	52.1
LSTM-CNN (Zhou et al., 2016)	52.4
NTI (Munkhdalai and Yu, 2017)	53.1
BCN+Char+CoVe (McCann et al., 2017)	53.7
BCN+ELMo	**54.7**

Table 14: Test set accuracy for SST-5.

from all layers of the biLM provides a modest improvement.

A.8 Sentiment classification

We use almost the same biattention classification network architecture described in McCann et al. (2017), with the exception of replacing the final maxout network with a simpler feedforward network composed of two ReLu layers with dropout. A BCN model with a batch-normalized maxout network reached significantly lower validation accuracies in our experiments, although there may be discrepancies between our implementation and that of McCann et al. (2017). To match the CoVe training setup, we only train on phrases that contain four or more tokens. We use 300-d hidden states for the biLSTM and optimize the model parameters with Adam (Kingma and Ba, 2015) using a learning rate of 0.0001. The trainable biLM layer weights are regularized by $\lambda = 0.001$, and we add ELMo to both the input and output of the biLSTM; the output ELMo vectors are computed with a second biLSTM and concatenated to the input.