Advanced NLP (CS7.501)

Monsoon 2022, IIIT Hyderabad
Assignment 2

Abhinav S Menon (2020114001)

Questions (Theory)

ELMo & CoVe

ELMo and CoVe differ in both architecture and pretraining details, although both provide contextual representations.
ELMo uses a six-layer bidirectional LSTM, while CoVe uses a two-layer bidirectional LSTM. Furthermore, ELMo sums up the hidden states of each LSTM in its stack, while CoVe takes the final outputs only.
Moreover, ELMo is pretrained on a bidirectional language modelling task (next- and previous-word prediction), while CoVe is trained on a machine translation task. Thus CoVe is paired with a two-layer decoder LSTM during its pretraining, while ELMo is simply used with a classification head to predict the next word.

Another point of difference lies in the incorporation of *global embeddings* (typically GloVe in both cases) into the contextual embeddings. ELMo simply adds them to the forward and backward embeddings, while CoVe concatenates its contextual representation with GloVe to create the final representation.

Character Convolutional Layer

A character convolutional layer is simply an application of CNN (convolutional neural networks) to character sequences instead of pixels. They give the model information extracted from the subword level, which is not available to it otherwise.

An alternative to this is any form of subword tokenisation, of which many methods have been identified. One popular method is BPE (byte-pair encoding), under which we start with characters as different tokens and merge them until we reach a certain vocabulary size. Morph analysis is another method that can be followed.

Analysis

The model was trained for one epoch on the language modelling task, with the following hyperparameters:

$$\begin{aligned} \text{hidden size} &= 100 \\ \text{learning rate} &= 10^{-3} \\ \text{optimiser} &= \text{SGD} \end{aligned}$$

It achieved an average loss of approx 9.95 after this pretraining.

It was then trained for 10 epochs on the text classification task with the same hyperparameters. The classification report of the model's performance after this is as follows.

dev set

	precision	recall	f1-score	support
0	0.24	0.47	0.32	500
1	0.00	0.00	0.00	500
2	0.21	0.31	0.25	500
3	0.19	0.21	0.20	500
4	0.29	0.12	0.17	500
accuracy			0.22	2500
macro avg	0.19	0.22	0.19	2500
weighted avg	0.19	0.22	0.19	2500

test set

	precision	recall	f1-score	support
0	0.22	0.46	0.30	1000
1	0.00	0.00	0.00	1000
2	0.20	0.30	0.24	1000
3	0.21	0.22	0.21	1000
4	0.31	0.13	0.18	1000
accuracy			0.22	5000
macro avg	0.19	0.22	0.19	5000
weighted avg	0.19	0.22	0.19	5000

By this stage, loss was not significantly reducing (the change between epochs starts at about 0.2 and reaches 0.005 by the tenth epoch). This means that the model had learnt all it could under the pretraining it had received, which means that the pretraining plays an important role in the model's learning.

The models can be found **here**.