Advanced Topics in Computational Semantics

Lysander de Jong, 11788674

April 13, 2021

1 Introduction

In this short report we attempt to reproduce the paper by Conneau et al. (2017). To this end we implement four different models which attempt to construct sentence embedding from pre-trained word GloVe embeddings (Pennington et al., 2014). We train all the models on the SNLI (Bowman et al., 2015) dataset to obtain a generally applicable sentence encoder. For more detailed information about what has been performed, we refer the reader to Appendix A. The source code for this project can be found on github[1].

2 Analysis

In table 1 we compare our four model directly against the authors choosing two of their results which are most relevant for our comparison. The original BiLSTM-Max and ours a fairly competitive trading blows at various NLP tasks. Another thing to note is seeing how well simply taking the mean of words vectors works, being on par with GloVe BOW. More of a surprise is the performance of our LSTM, which is trailing the others. On the question-type classification task (TREC) it scores particularly bad. This might be due to the model forgetting how the question started, as this tends to be indicated by the first word in the sentence. A suspicion confirmed when we look at the sore for the same task for the BiLSTM, which has no trouble with it.

In sentiment classification (SST), we see that the task relies more on the meaning of words, as both LSTM and BiLSTM are outpaced by GloVe-mean.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	SICK-R	SICK-E	STS14
Orig. GloVe BOW	78.7	78.5	91.6	87.6	79.8	83.6	72.1	0.800	78.6	0.54/0.56
Orig. BiLSTM-Max	**79.9**	**84.6**	92.1	**89.8**	83.3	88.7	**75.1**	0.885	**86.3**	0.68/0.65
GloVe-mean	77.47	78.49	91.45	87.86	81.05	81.6	71.77	0.763	75.22	0.55/0.56
LSTM	77.12	79.52	85.35	87.93	68.04	47.4	69.28	0.793	80.70	0.49/0.47
BiLSTM	79.74	83.05	90.44	88.59	79.02	85.2	71.71	0.857	84.55	0.53/0.52
BiLSTM-Max	79.03	82.73	**92.28**	88.91	**83.75**	**89.8**	74.72	**0.888**	85.95	**0.69/0.67**

Table 1: Results on various NLP tasks.

In table 2 we show how our model perform on the training task (SNLI) and what the averaged are between tasks. Micro is the average of the tasks in table 1 taking into account the number of sample in each task, while macro is the plain average over each tasks.

Here our Bi-LSTM-Max is about half a point worse compared the the authors original. We suspect this is mostly due due differences in seeds and the authors more extensive testing. However, when averaging of the downstream NLP task the trend reverses and our model outperform the original in both micro and macro averages.

It is also here where we see how bad simply taking the mean can be, as SNLI spears to be the only tasks difficult enough to separate the models based on complexity.

Another interesting disparity is that is between our LSTM and the author's, which seems to be performing much better on SNLI, scoring even beyond our BiLSTM, but is worse when we take the averages of the transfer tasks.

		SNLI		Transfer	
Model	Dim	dev	test	micro	macro
Orig. LSTM	2048	81.9	80.7	79.5	78.6
Orig. BiLSTM-Max	4096	**85.0**	**84.5**	85.2	83.7
GloVe-mean	300	65.87	64.71	82.96	80.61
LSTM	2048	72.78	71.14	81.18	74.42
BiLSTM	4096	77.25	75.86	84.84	82.79
BiLSTM-Max	4096	84.39	83.95	**85.66**	**84.65**

Table 2: SNLI and average scores.

[1] `https://github.com/LysanderdeJong/ATCS2021`

In table 3 we use various probes to investigate what type of knowledge the models has learned. In the length task, we observe that that average word embedding unsurprisingly do not know the sentence length. The LSTM suffers the same, as has been discussed when talking about the TREC task. However on the word content task GloVe-mean perform really well, only outperformed by BiLSTM-Max.

The BiLSTM is really good at detecting when unwanted changes have been made to sentences, meaning that the last hidden state hold much information about the sentence which is lost when taking the maximum over all states.

Task	GloVe-mean	LSTM	BiLSTM	BiLSTM-Max
Length	59.70	61.35	73.69	**76.25**
WordContent	79.15	25.34	44.87	**89.09**
Depth	33.35	35.93	**41.44**	38.30
TopConstituents	62.45	41.63	76.38	**76.88**
BigramShift	51.01	61.32	**62.42**	59.20
Tense	84.15	81.70	86.12	**86.67**
SubjNumber	78.51	72.96	**86.37**	83.96
ObjNumber	76.58	74.00	79.88	**79.96**
OddManOut	51.57	**61.83**	60.85	53.86
CoordinationInversion	53.89	65.77	**70.62**	66.71

Table 3: Sentence embeddings property probing.

3 Conclusion

The general takeaway from these experiments is that given fixed size sentence embedding are difficult to construct and must strike a balance between knowledge about each word in the sentence or general knowledge about the ensemble as a whole. However the expressivity of the embeddings increases with size. The BiLSTM-Max strikes a middle ground in this regard.

References

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Conneau, A., & Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Appendices

A Extended introduction

In this short report we attempt to reproduce the paper by Conneau et al. (2017). To this end we implement four different models which attempt to construct sentence embedding from pre-trained word GloVe embeddings (Pennington et al., 2014). We train all the models on the SNLI (Bowman et al., 2015) dataset. This dataset consist of 570k sentence pairs, in which the model has to predict the relation between the pairs. There are three categories: entailment, neutral, contradiction. These complex relations should force the model to learn about language, and hopefully obtain a generally applicable sentence encoder.

As mentioned before, we use four model of increasing complexity to to encode our sentences. Our first model, named 'GloVe-mean', simply takes the mean of the word vectors to create our sentence embedding. The second model, named 'LSTM', is naturally and lstm model. We opt to use 2048 as our hidden dimension, similar to Conneau et al. (2017), but make no use of any normalization layers or dropout to regularize the model. We use the final hidden sate of the LSTM as our sentence embedding. Similar to the LSTM, our 'BiLSTM' model , sharing it configuration, with the exception of running in two directions. Thus we concatenate the last hidden state of each direction to form our sentence embedding. Lastly, the fourth model 'BiLSTM-Max', uses the same configuration as the third mode, however we take the maximum over each hidden state as our sentence embedding.

With the encoder established the can obtain sentence embedding for arbitrary sentences. In order to train it on SNLI we use the encoder on both sentences in a pair, then concatenate their embeddings as well as their differences and their element-wise product. On this we fit a simple multiplayer perceptron, which is slight different than in the paper. The main change is that we have add a ReLu non-linearity in between after the hidden layer. Making our MLP a bit more expressive than in the original paper. We train our model using the same training criteria as in the original paper, and refer individual to it, if they wish greater insight into the training setup.

After training we evaluate using SentEval (Conneau & Kiela, 2018) our models to see how well the sentence embeddings generalize to downstream tasks. Using the same framework we also do a few probing tasks to see what properties the sentence embeddings have learned.