Learning Natural Language Inference with LSTM

1. Data Used

The author used Stanford Natural Language Inference(SNLI) Corpus as training data. It has a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the labels entailment, contradiction, and neutral, supporting the task of natural language inference (NLI), also known as recognizing textual entailment (RTE). The columns of the dataset are gold_label, sentence1_binary_parse, sentence2_binary_parse, sentence1_parse, sentence2_parse, sentence1, sentence2, captionID, pairID, label1, label2, label3, label4, label5

2. Methodology

The author came up with a new LSTM-based architecture for learning natural language inference. Compared to previous models, the author used an LSTM to perform word-by-word matching of the hypothesis with premise. The author also used attention-weighted representation of the premise to help push the accuracy to 83.5% on the SNLI corpus. The author's LSTM sequentially processes the hypothesis, and at each position, it tries to match the current word in the hypothesis with an attention-weighted representation of the premise. Matching results that are critical for the final prediction will be "remembered" by the LSTM while less important matching results will be "forgotten."

3. Model Used

The First limitation of the LSTM model was still using a single vector representation of the premise, namely h a N , to match the entire hypothesis. The second limitation is that the model does not explicitly allow us to place more emphasis on the more important matching results between the premise and the hypothesis and down-weight the less critical ones. In order to solve two limitations above, the author proposed to use an LSTM to sequentially match the two sentences. First, the author process the premise and the hypothesis using two LSTMs, but the author did not feed the last cell state of the premise to the LSTM of the hypothesis. Next, we generate the attention vectors ak similarly to Eqn (2). Our h is the hidden state at position k generated from our mLSTM. This LSTM models the matching between the premise and the hypothesis.

4. Key Conclusions

In this paper, the author proposed a special LSTM architecture for the task of natural language inference. The author first used neural attention models to de- rive attention-weighted vector representations of the premise. And then the author designed a match-LSTM that processes the hypothesis word by word while trying to match the hypothesis with the premise. The last hid- den state of this mLSTM can be used for predicting the relationship between the premise and the hypothesis.