Some findings during replication

Liu Chengyuan

July 23, 2020

1 Brief View

1.1 Pre-Processing

In the processing, document texts and their labels are extracted from the *raw muc*. We should keep:

- Indexes of Both **sentences from one 'group of k' in one line** and **list of sentences from one 'group of k'**.
- Indexes of Labels for each word.
- Tokenized words.

Take continuous k sentences in one paragraph for each training batch.

1.2 Batchify data

In this section, the main purpose is to pad the data to maximum length. Apply it to:

- Sentences from one 'group of k' in one line, which means we pad each sentence to the maximum length of **batch**.
- Labels from one 'group of k' in one line, same as the above description.
- Pad to the maximum length of sentences in one group. It is considered as an optimization.

1.3 Model

About word embedding 1.3.1, LSTM 1.3.2, concat 1.3.3 and CRF 1.3.4.

1.3.1 Word Embedding

Both Glove and Bert are used in the embedding layer, which means we have word-self and contextualized meanings. Because of Bert model, better use bert tokenize to get a good result.

1.3.2 LSTM

LSTM requires same length data for each batch.
Pay attention to the usage of *pack_padded_sequence*. Remember to do it on both of the two docs.

1.3.3 Concat

Although simple sum function is mentioned in the paper, the author only apply gate and sigmoid to outputs of linear(lstm_out).

1.3.4 CRF

Lets try *pytorch-crf* package. It is supposed to excel hand-made crf.

2 Details

1. There is a scale factor in the embedding, which equals $\sqrt{\frac{embedding\_dim}{3}}$. It's for the cases when word not find in the pre-trained embedding dict. We apply random vector range in $(-scale, scale)$.

2. Output of the *pad_packed_sequence* is shaped like $(seq\_len, batch\_size, *)$. There is a transpose with dropout layer.

3. If optimizer is set to 'SGD', we apply dynamical learning rate to the model, which is $\frac{lr}{1+epoch \times decacy\_rate}$.

3 Questions

1. What does *feature_prefix* do?
2. There are extra empty data suffixed at the return Tensor of *read_instance*.
3. In *calculate_PZ_* from CRF model, view score to *[batch size, sequence len, tag size, tag size]* then transpose?

4 Optimization

1. We have to match the tokens with roles one by one when using tokenized words. A faster procession is expected when solved by KMP.
2. For sentence level lstm, trying to take the whole paragraph as one batch, instead of send them through LSTM layer sentence by sentence.
3. Remove some redundant arguments or variables.
4. Detailed Annotation.