MODEL:CRF+LSTM

Contents

Date: 2021-01-15.

2020 *Mathematics Subject Classification.* NLP.

Key words and phrases. Machine Learning, Data Mining, ...

1. Introduction

Neural network has become a model that can deal with many NLP tasks effectively. This kind of method is similar for sequence labeling tasks (such as CWS, POS, NER). The token is mapped from discrete one hot representation to dense embedding in low dimensional space, and then the embedding sequence of sentences is input into RNN. Neural network is used to extract features automatically, and softmax is used to predict the label of each token.
The disadvantage is that the token is classified independently when it is labeled, and the label predicted above cannot be used directly.
In order to solve this problem, we propose LSTM + CRF model for sequence annotation, and join CRF layer after LSTM layer to do sentence level label prediction, so that the annotation process is no longer independent classification of each token.

2. Learning stage

Stage one: forget stage: Choose the most important from historical data.
Stage two: choose the memory stage: Select the most important from the input data.
Stage three: output stage: Control output.

3. Model Building

Main ideas:
1.The token is mapped from a discrete one hot representation to a low dimensional space to become a dense embedding.
2.The embedding sequence of sentences is input into LSTM, and the feature is extracted automatically by neural network.
3.Using CRF to predict tag at sentence level.

4. Embedding layer

Input: the ID list corresponding to the word and the characteristics of word segmentation information
Objective: to transform words into low dimensional dense vectors
Operation: train the n-dimensional word vector model in advance, and get the n-dimensional vector of each word through query. The n-dimensional vector is output to dropout layer together with the feature vector of segmentation information. The pre trained embedding can capture the global word similarity.

5. Dropout layer

Input: the n-dimensional vector corresponding to the word and the feature vector of segmentation information
Objective: to alleviate over fitting

6. Bi LSTM layer

Input: the n-dimensional vector corresponding to the word
Objective: to extract sentence features automatically
Operation: the char embedding sequence of each word is taken as the input of each time step of Bi LSTM, and then the hidden state sequence of forward LSTM output and the hidden state sequence of reverse LSTM output in each position are spliced

according to the position to get the complete hidden state sequence. Here you can capture some rules inside the word. Combining with pre training embedding, we can get better word embedding.

7. Linear layer.

Input: hidden state sequence of Bi LSTM output
Objective: to get the sentence features extracted automatically
Operation: map the hidden state sequence to the dimension of the number of tags, so that each dimension can be regarded as the score value of word to tag. At this time, if softmax is used, the classification results can be obtained directly, but the information that has been marked in other locations can not be applied, so a CRF layer is connected next.

8. Loss layer

Embedded CRF layer.
Input: word for the tag sequence rating
Objective: sequence annotation at sentence level
Operation: score again; the score is divided into two parts, LSTM output score plus CRF transfer matrix (i.e. transfer score from one label to another label). After scoring, softmax is used to get the normalized probability.

References

List of Todos