Algorithm Description

HIT WI

Abstract. For this ccks named entity recognition task, we mainly use two methods.The single CRF and the LSTM-CRF neural network model.We use the gold label data set to do the unsupervised learning for both model,and the unlabel data set to do the embedding and supervised learning.Finally get the ...

Keywords: Name entity recognition,CRF,LSTM

1 LSTM-CRF method

1.1 Neural Network Architecture

In this section, we describe the components (layers) of our neural network architecture. We introduce the neural layers in our neural network one by-one from bottom to top.

Char Embedding We use the char level vector to represent the Chinese words,by using the word2vec in the unlabeled dataset,the embedding dimension is 100.

Bi-directional-LSTM For many sequence labeling tasks it is beneficial to have access to both past (left) and future (right) contexts.The solution whose effectiveness has been proven by previous work (Dyer et al., 2015) is bi-directional LSTM (BLSTM). The basic idea is to present each sequence forwards and backwards to two separate hidden states to capture past and future information, respectively. Then the two hidden states are concatenated to form the final output.

CRF For sequence labeling (or general structured prediction) tasks, it is beneficial to consider the correlations between labels in neighborhoods and jointly decode the best chain of labels for a given input sentence.Therefore, we model label sequence jointly using a conditional random field (CRF) (Lafferty et al., 2001), instead of decoding each label independently. In our code,we try to use the negative log likelihood function and the labelwise function to get the loss of the CRF layer.Using the marginal decode and viterbi decode.

BLSTM-CRF The basic train step is as fellow:

Algorithm 1 CRF loss Negative log likelihood

input: feature from BiLSTM,Tags
output: Loss
function LIKELIHOOD LOSS($feature, self, tags$)
    $forwardscore \leftarrow CRFlayer.forward(feature)$
    # Do the forward algorithm to compute the partition function
    $goldscore \leftarrow CRFlayer.score_sentence(feature, tags)$
    # Gives the score of a provided tag sequence
    return $forwardscore - goldscore$
end function

Algorithm 2 Network train step

input: Sequences of context C,Sequences of BIEO tags T
output: Model
Initialize the word-embedding layer← Separating text by char and do Word2vec
model←LstmCrfModel.BiLSTM CRF(Model parameters)
optimizer ← optim.SGD(model.parameters(), lr←0.01, weight decay←1e-4)
for in range(epcho) **do**
    for all *sentence* ∈ *context* **do**
        model.LSTMtrain(sentence)
        Input to the CRFlayer
    end for
    Get loss(T,predict tags)
    SGD Optimize the loss
end for
return $Model$

Optimizer	Embedding	Dropout	Learning rate	Weight dercay	CRF loss
SGD	Word2Vec	0.5	0.01	1e-4	lablewise

2 Parameter Initialization

The Parameter Initialization is as fellow:

3 Single CRF method

3.1 feature chose

Chose the pos tag to be the extra feature,to do the crf in char level.

3.2 Parameter Initialization

The Parameter Initialization is as fellow:

Optimizer	Embedding	Dropout	Learning rate	Weight dercay	CRF loss
SGD	Word2Vec	0.5	0.01	1e-4	lablewise

References

1. Schlkopf B, Platt J, Hofmann T. Training Conditional Random Fields for Maximum Labelwise Accuracy[C]// Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December. DBLP, 2006:529-536.
2. Ma X, Hovy E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF[J]. 2016.
3. Goldberg Y, Levy O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method[J]. Eprint Arxiv, 2014.