NATURAL LANGUAGE PROCESSING

Assignment 4

Submitted by

Jayakrishna Anvesh Paidy

SBU Id: 112675567

Relation Extraction

1. Baseline Model:

As part of this model, I have implemented the Bi-Directional networks with GRU. The word embeddings are passed to this bidirectional model and the sequences that are obtained are passed into an attention layer to get the effective representation of the sentence. This representation is passed into a multiclass classifier in order to predict the relation between the elements.

For computing the attention weights, we have computed the SoftMax on the product of trained weight matrix and the representations (tanh(reps)) obtained from the Bi-GRU network (As explained in the paper 1)

$$M = \tanh(H)$$
$$\alpha = \mathrm{softmax}(w^T M)$$
$$r = H \alpha^T$$
$$h^* = \tanh(r)$$

While computing the loss for the model, we have computed the L2 regularization for all the model trainable parameters and added it to the loss to improve the model accuracy.

2. GRU Experimentation:

As part of the this, we have performed the following experimentations,

1. Running with only the word embedding features: On passing only the word embedding features as the input to the model, we have attained the following accuracies.

Epochs	F1 score	Validation loss
1	0.3175	1.996
5	0.5244	1.7484
10	0.5303	2.3929

2. Running model with the word + pos features:

Epochs	F1 score	Validation loss
1	0.32	2.1206
5	0.5417	2.4558
10	0.5181	2.3929

3. Running the model with word + dep structure features:

Epochs	F1 score	Validation loss
1	0.32	2.1206
5	0.576	1.8471
10	0.5623	2.1147

Through this experiment it is clearly evident that the word + dependency features clearly outperforms than running on the other two inputs in terms of the f1 score.

4. Advanced Model:

Motivation:

In order to improve the accuracy of the model, we plan to use an ensemble of two models. For this experiment we are computing the ensemble of two models. a) Connectionist bi-directional RNN which is best suitable for the sentence classification tasks by producing taking the decisions based on the intermediate representations produced.

b. CNN with extended contexts (As in https://arxiv.org/pdf/1605.07333.pdf) CNNs perform a discrete convolution on an input matrix with a set of different filters. Using these Convolution Networks, we can capture some lexical ,sentence level features that can be effectively used for the purpose of Relation Extraction.

c. Ensemble: Finally , we will combine these two models using the soft vote ensemble process (each classifier provides a probability value that element pair (e1,e2) has to have particular relation. The predictions are weighted by the classifier's importance and summed up. Then the target label with the greatest sum of weighted probabilities wins the vote) Through this process, our model tend to combine the features that are learned from both neural networks to effectively perform the task of relation extraction.

1. Connectionist bi-directional RNN

As part of this model I have built a model using Connectionist bi-directional RNN, which on taking the word embeddings along with dependency structure as inputs and pass it to GRU in both forward and backward direction.

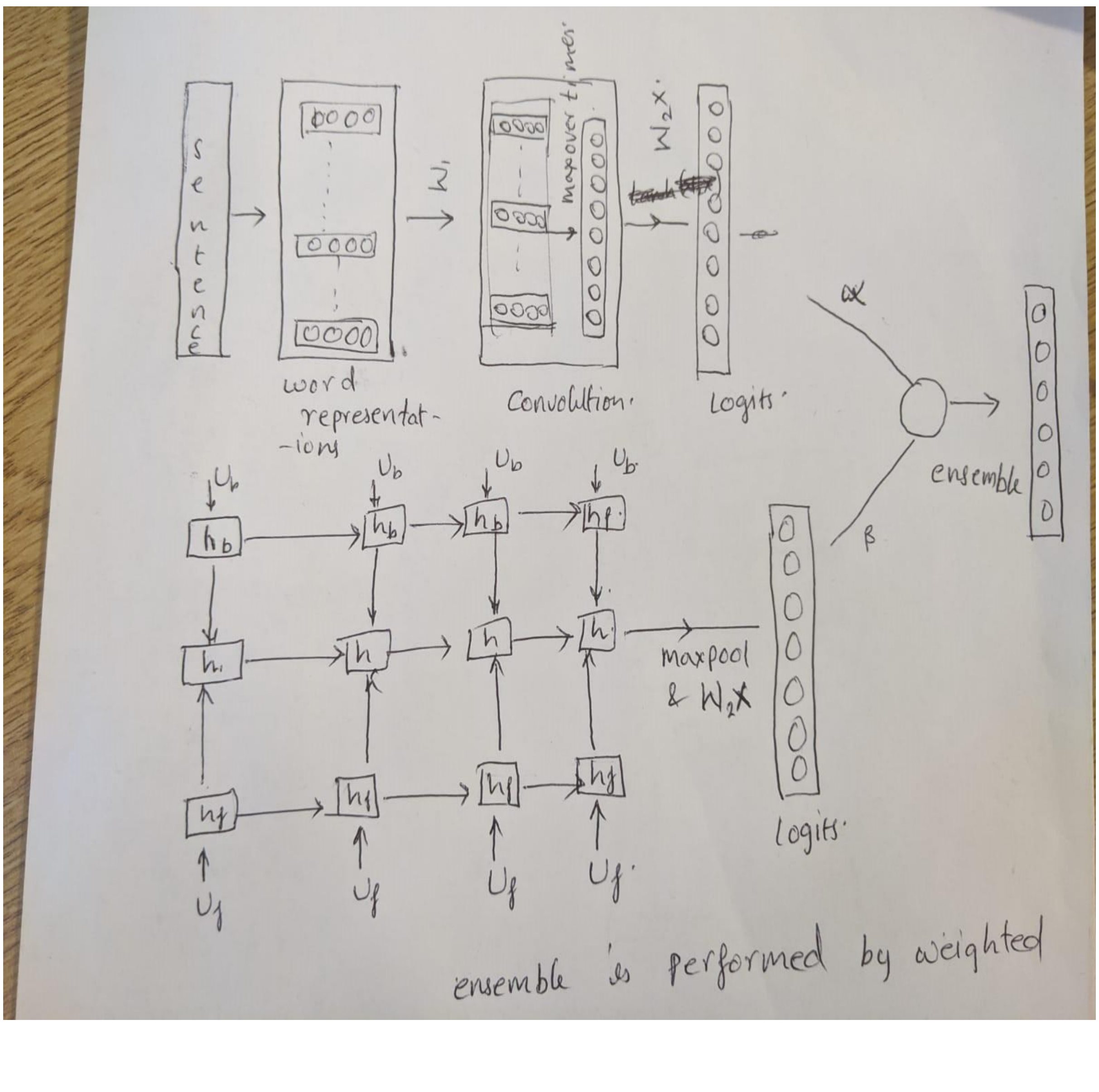
The generated sequences (both in forward $h_{f\,t}$ and backward direction $h_{b\,t}$) are passed back into another GRU layer. The output is generated by max pooling over the sequences generated from the final layer. This representation is passed to classification layer for extracting the relation.

$$
\begin{aligned}
h_{f_t} &= f(U_f \cdot w_t + V \cdot h_{f_{t-1}}) \\
h_{b_t} &= f(U_b \cdot w_{n-t+1} + B \cdot h_{b_{t+1}}) \\
h_t &= f(h_{b_t} + h_{f_t} + H \cdot h_{t-1})
\end{aligned}
$$

These Representations are passed to the attention layer, we will get the final state based on the attention weights similar way to that of the Bi-Directional networks with GRU

2.Convolution neural network: As part of this model we have implemented a convolutional neural network with 1D filter. With tanh activation and maxpool from the representations. Apply dropout with rate of 0.4. This representation is passed into the classification layer to generate the logits.

In the final layer we compute ensemble of these two models output by computing the weighted average of the logits from these models. This ensemble gives us better f1 score in comparison to the baseline model. Here we perform soft voting process



The figure above depicts the neural models used and the outputs of these models are used to perform the classification to predict the relation between the elements (Multiclass classification).

Results.

Using the above specified model the we have obtained the following accuracies.

Epochs	F1 score	Validation loss
1	0.3708	1.9744
5	0.581	1.8089
10	0.5677	2.226

Further the model accuracy can be increased by further tuning the convolution model to get better results.

Best Configuration: "vocab_size": 10000, "batch_size": 10, "embed_dim": 100, "training": true, "hidden_size": 128, "type": "advanced"