1.

(a)

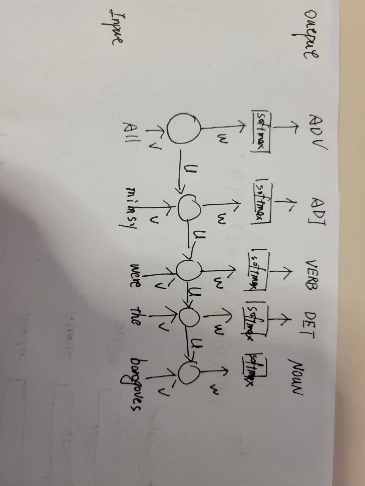
i.

ii. No. because finally the output will go through the softmax function which will convert the vectors into probability distribution.

iii. No. Word2vec model use context text, which contains previous n words and following n words, while here we only use previous words.

(b)

i.



ii.

P(y\_i | x\_<i) = y\_i

Y\_i = softmax(Wh\_i + b\_2)

H\_i = sigmoid(Vx\_i + Uh\_i-1 + b1)

X\_i = onehot(x\_i)

iii.

encoder-decoder structure. Encoder RNN to encode the input sequence into vector embedding, decoder RNN generate tagging sequence based on final hidden state and its current decoder output.

iv.

h\_i = RNN\_enc(x\_i, hi-1)

s\_i = RNN\_dec(y\_i, si-1)

p(y\_i |y<I, x) = softmax(W concat(s\_i, h[x])+b)

(c)

i.

approximate softmax: during training time, vocabulary based on the training partition; at test time: determine likely target words based on source text.

(e)

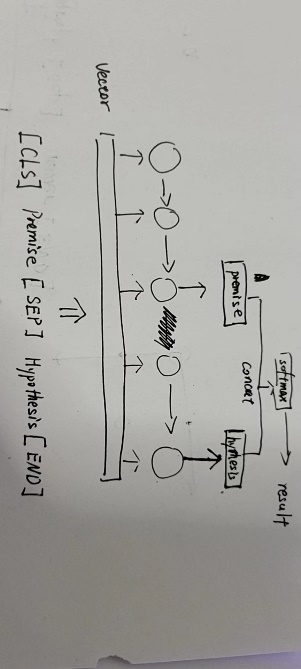
Input is sentence and the target predicate.

RNN takes the whole input and precious tagging result to predict the next tag.

Output is the semantic tagging of the sequence where each word should be associated with a tag.

3.

(a)



Training objective:

Training algorithm:

Training input/output:

(b)

The premise and hypothesis are processed seperatedly athough in the same LSTM model, there should be some mechanism to capture the similarity between premise and hypothesis.

Attention

context

(c)

Bert model and finetuning for classification.

(见google drive)

(d)

Paraphrasing relation.

Mainly relys on Human evaluation. If have reference, could use BLEU to automate the evaluation process.