1. (10 points) Copy your code comments here.

A. The function of following line is to create two LSTM chain for encoder layer. one List for the forward encoder and the other for the backward encoder. The function of add_link() is to register a child link to this chain. When access links we can use traverse link list. The reason why using two loops is that encoder process use bidirectional-LSTM: the forward LSTM captures the feature information above, while the reverse LSTM captures the following feature information
B. Because the encoder process uses bidirectional-LSTM, so after the encoder process, the size of the LSTMs is doubled for decoder process , and the decoder process is unidirectional.So for the embed layer, the input size is the size for English and the output size is doubled size of the LSTMs. For LSTM layers the input size is the output size of the embed layer and the out put size is the same size as the input size. For linear layer, the input size is the last output size of LSTM layers and the output size is the size for English
C. set_decoder_state: sets the internal state of the decoder, joining separately the cell state and the hidden state from the forward encoder's last layer with that from the backward encoder's last layer.
c_state: A new cell states of LSTM units,
h_state: A new output at the previous time step.
D. Because the encoder process uses bidirectional-LSTM, one is encode the word with forward LSTM and the other is encode the word with backward LSTM
E. It Computes softmax cross entropy between logits and labels, measures the probability error indiscrete classification tasks in which the classes are mutually exclusive. The value means the probability error from the machine translation result from the right translation sentence.
F. Replacing words with UNK may cause the translate architecture are not complete.

2. (10 points) Examine the parallel data and answer questions. (Plots may appear in the appendix.)

1. The distribution of sentence lengths in the English and Japanese is shown in figure 1. The length of English and Japanese show a normal distribution, most of English sentence length is about 30 shorter than Japanese sentence length
2. There are 97643 tokens in English and 143581 tokens in Japanese
3. There are 7211 word types in English and 8252 word types in Japanese.
4. There are 3384 UNK in English and 4172 UNK in Japanese
5. NMT system translate sentences in isolation rather than considering the context of the paragraph or page so that if the sentence is too long may not capable of achieving complex structural order and long distance sequencing. And if there are a lot of unknown words, the translation results will not be smooth. So that the longer the sentence is or the more the unknown words are, the poor the translation is .

3. (10 points) What language phenomena might influence what you observed above?

Your answer here. First, there is a phenomenon that when the sentence is long, the subject is always translated well while the object and verb is not very well , which is because the order of Japanese is Subject, Object and Verb while the order of English is Subject, verb and object(Matthew S. Dryer. 2013. Order of Subject, Object and Verb.), and when the sentence is much too long , it's alignment is worse. Then we can notice that most of the English sentences are shorter than Japanese sentence, which may because of the language environment, Japanese is more euphemistic than English. and there are less English tokens, word types and UNK in English than Japanese , which is because a normal Japanese will use about 50000 words while a normal English will only use about 10000 words.

4. (10 points) Answers to questions about sampling, beam search, and dynamic programming.

1. First l choose one sentence to see the difference between using and not using sample, l notice that sample sometimes makes translation worse while sometimes better, for example , one Japanese sentence should be translated as "there is no doubt as to her beauty ." When not using sample it is translated as "she is no beautiful beautiful beautiful beautiful in . . _EOS " and the precision is 0.2727, recall is 0.3333. When using sample it may be translated as "it is not usually . are verge are sweet up . _EOS" the precision is 0.1667, recall is 0.2222, which becomes worse, and it may also be translated as "she is no her beautiful in person . words in _EOS " and the precision is 0.3636, recall is 0.4444, which become

better. Then l try to see the first ten sentence using of not using sample. l find 2 translation become better, 1 translation is not changed and the other three translation become worse. So, all in all, l think using sample will actually improve the a little sentence but it will be worse to most of the sentence.

2. Beam search is a form of greedy search, What we do is instead of computing the most likely first word, we compute the b most likely first words (this set of b most likely candidates is called the "beam"). Then to compute the second word based on the former words. So first l would change the function of selected_word, from max probability to first b probability. Then to compute the second word, we condition in turn on each of those b first words, and score all the sequences of length 2 obtained that way (b*n of them if n is the size of the vocabulary), then we take the top b of those. Now we can compute the b*n highest scoring length-3 sequences whose length-2 prefixes are in our previous beam, and take the highest-scoring b of those, etc. Repeating this process takes only b times as much computation as greedy decoding. So Then l will change the function of encoder_decoder_train with two loop. Loop until the BEAM contains no nodes, while set is not empty and the number of nodes in BEAM is less than b, if state is not in the hash_table, add state to hash_table and Bean
3. No, l can not implement dynamic programming for this RNN model because we can only compute the words one by one, but dynamic programming store the state and skip the words and fill the gap later. That is also NMT system's weakness, the alignment for this architect is not very well

5. (10 points) Experiment with changes to the model, and explain results.

Your answer here. The result for the blue and perplexity and how many translation become better or worse is shown in table1 in appendix, and l use encoder =1and decoder=1 as my baseline. 1. Add the number of layers in both encoder and decoder, l find the perplexity increase when layers=2 and then decrease when layers=3 . and the bleu is decreasing all through, and most of the fist ten sentence become worse . Adding layers on decoder, may cause the perplexity increase and bleu decrease. Most of the fist ten sentence become worse. as well 3. Adding layer on encoder, may cause the perplexity increase and bleu decrease but the fist ten sentence five become worse and five become better.
All in all, There is no exact tendency for the change of layers. and the bleu and perplexity has no close relation. But we know that the smaller the perplexity is and the higher the bleu is , the better the model is. In my experiment when both encoder and decoder has one layer performs best, l think it may because the data is too limit so that it is easy to cause overfitting.

6. (10 points) Implement dropout, and explain the results.

Your answer here. baseline: decoder and encoder layer=1.hidden_units=100. l add L+1 dropout followed to the paper in the function feed_lstm where feed into the embed layer and LSTM layer and l implement after both embed layer and LSTM layer .The ratio means how many percentage of element of a layer's being set to 0. The experiment result is shown in figure2. After adding dropout (0.1), the perplexity is decreased . the bleu is decreased as well. When dropout percentage increase to 0.3 it is better than 0.1 but still worse than baseline.When comparing first ten sentence's translation result, l find that 2 sentence become better while other 8 sentence become worse, and more UNK appears. For example, one sentence should be translated as "i like my steak medium ." baseline is translated as" i like my steak medium . _EOS ", But after dropout(0.3), it is translated as "i like_UNK _UNK _UNK tickets .. _EOS " dropout make many word drop so that the NMT system has to use UNK to replace the word.

7. (20 points) Implement attention. This question will be evaluated on the basis of your code.

8. (10 points) Explain the results of implementing attention.

*Your answer here*baseline:decoder and encoder layer=1.hidden_units=100. After adding attention, the perplexity is decreased and the bleu is increase, all shows that the model become better. When comparing first ten sentence's translation result, l find that 7 sentence become better and only 3 sentence become worse. In a specific translation result, for example, one sentence should be translated as "there is no doubt as to her beauty." baseline is translated as"she is no doubt she her her her . .. _EOS ", But after adding attention, it is translated as "there . is no doubt to as beauty beauty her . _EOS " adding attention makes "beauty" being translated. Using attention will actually improve the translation effect

9. (10 points) Explain what you observe with attention. (Figures may appear in an appendix.)

Your answer here A perceptron formula connects the target language to each word in source language , and then normalizes it by a soft function to get a probability distribution, which is the attention matrix.The heat-map clearly reflect the attention matrix. But The translation is still not very very for long sentence for example, こちら に 着 い たら すぐ 連絡 し て 下さ い 。 is translated as" get me with with with with with touch . . ". While the NMT system really work well on a set of isolated simple sentences, for example 相撲 を 見 た こと が あ り ま す か 。 is translate as "have you ever watched_UNK ? _EOS ", which is almost correct. And most of the Object, Verb, Subject is aligned correctly, for example the first sentence shown in figure5.見 is aligned correct to "watch" . 相撲 is aligned correct to "UNK" . Bur there are still a lot of unreasonable,ステーキ means steak in Google, but it is aligned to "I" .

Optional: you may include an appendix after the line above. Everything above the line must appear in the first three pages of your submission.

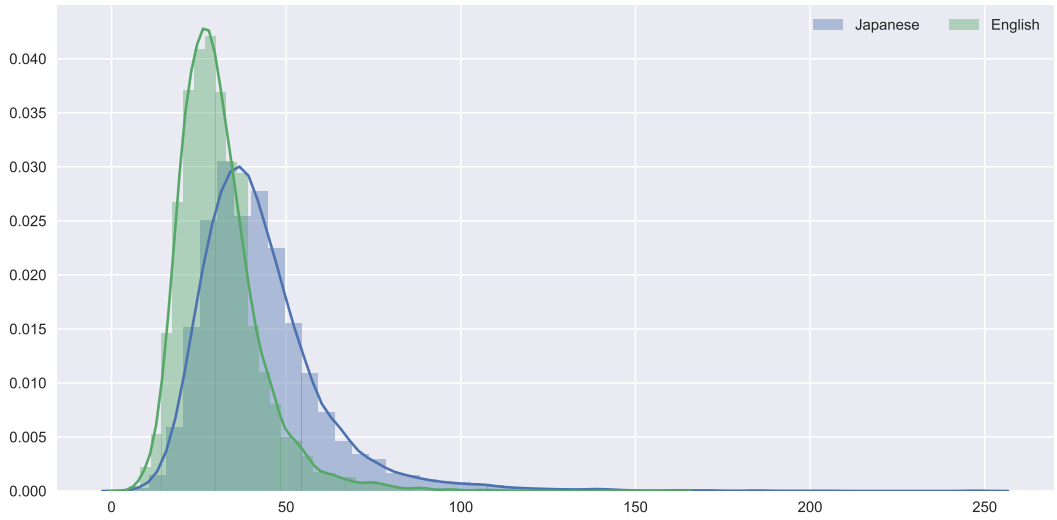

Figure 1: the distribution of sentence lengths in the English and Japanese

layer_enc	layer_dec	epoch	dev perplexity	bleu	loss	better	same	worse
1	1	10	51.1269	19.487	4.24			
2	2	10	56.1992	18.240	4.44	4	0	6
3	3	10	53.8567	15.315	4.80	4	2	4
1	3	10	57.2664	16.929	4.46	4	4	2
3	1	10	58.5886	17.853	4.69	5	0	5

Table 1: the number of layers in the encoder, decoder,

if dropout	dropout rate_dec	epoch	dev perplexity	bleu
no		10	51.1269	19.487
yes	0.1	10	45.9449	17.139
yes	0.3	10	48.7628	17.876

Table 2: dropout

if dropout	dropout rate_dec	epoch	dev perplexity	bleu
1	1	10	51.1269	19.487
yes	0.1	10	45.9449	22.176

Table 3: attention

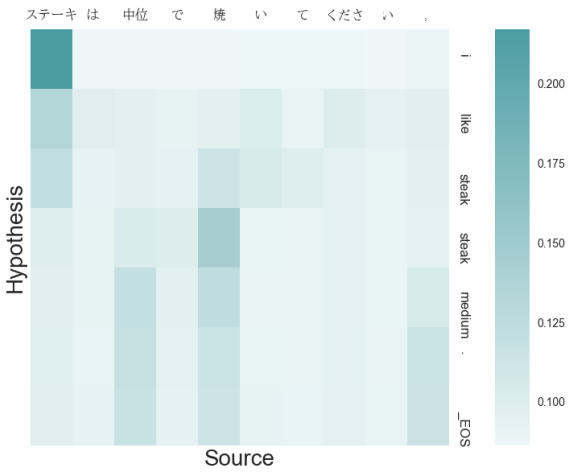

Figure 2: the distribution of sentence lengths in the English and Japanese

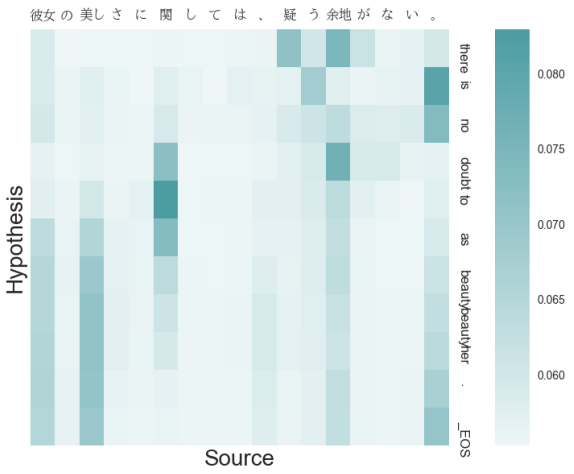

Figure 3: the distribution of sentence lengths in the English and Japanese

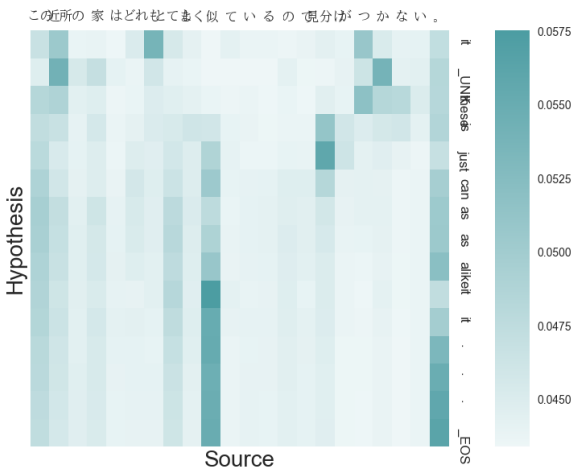

Figure 4: the distribution of sentence lengths in the English and Japanese

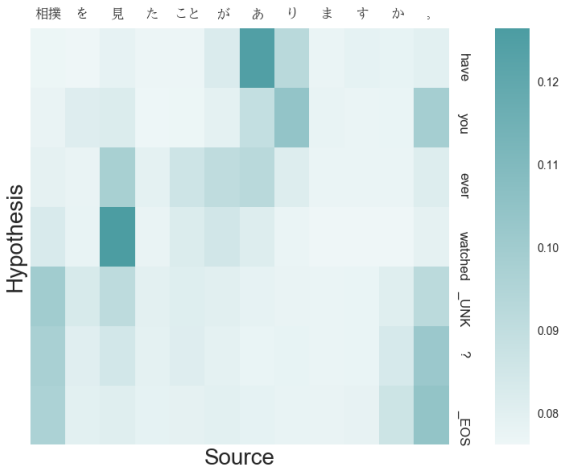

Figure 5: the distribution of sentence lengths in the English and Japanese

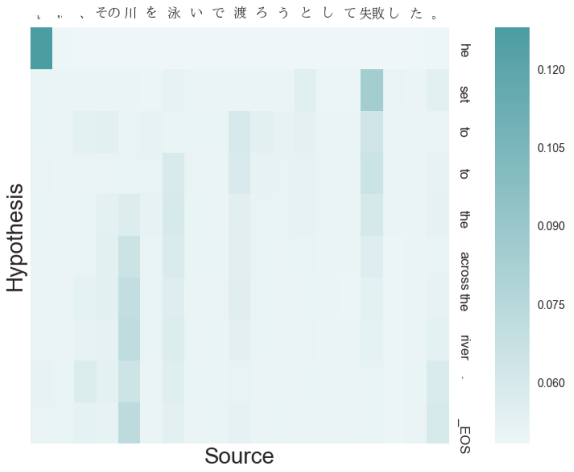

Figure 6: the distribution of sentence lengths in the English and Japanese