

1. (10 points) Experiment with stack pruning parameters. What is their affect on...

- log-probabilities: As the values of the maximum stack size (-s) and translations-per-phrase (-k) grow from 1, the value of log-probabilities grows larger. However, the value of log-probabilities is maximized and does not change anymore after the value of -s and the value of -k grow to a certain value.
- speed: As the maximum stack size (-s) and translations-per-phrase (-k) start to increase from 1, the speed is slowing down.
- translations: As the maximum stack size (-s) and translations-per-phrase (-k) grow from 1, the translation results get better. The results of translations(using larger values of -s, -k) have changed, these results make more sense, also become more fluent. For example, the worse translation result: they stand the same when their leaders stand to them; and the better one: they stand out when their leaders stand to them.(s=6, k=5 and s=100, k =100)
- maximum log-probability: -1353.247828.

2. (15 points) Define a new dynamic program for Part 2.

To get the best partial translation of the first i French words ending in English word e . There are two ways to this situation:

1.By extending a similar partial translation covering a shorter prefix.

$$h(i, e) = \arg \max_{h(j, e')_{e_1 \dots e_n e}} \log p(h(j, e')) + \log p_{TM}(f_{j+1} \dots f_i | e_1 \dots e_n e) + \quad (1)$$

$$\log p_{LM}(e_1 | e') + \sum_{n'=1}^{n-1} \log p_{LM}(e_{n'+1} | e_{n'}) + \log p_{LM}(e | e_n) \quad (2)$$

2.By filling in a gap in the set of translated words that was created by translating the second in pair of phrases first.

$$h(i, e) = \arg \max_{s(k, j, i, e')_{e_1 \dots e_n e}} \log p(s(k, j, i, e')) + \log p_{TM}(f_k \dots f_j | e_1 \dots e_n e) + \quad (3)$$

$$\log p_{LM}(e_1 | e') + \sum_{n'=1}^{n-1} \log p_{LM}(e_{n'+1} | e_{n'}) + \log p_{LM}(e | e_n) \quad (4)$$

There is a partial translation that covers the first i words of the French, except those from k to j . This case is the antecedent of 2. above. So the best possible translation of these words, ending in e , as follows:

$$s(k, j, i, e) = \arg \max_{h(k-1, e')_{e_1 \dots e_n e}} \log p(h(k-1, e')) + \log p_{TM}(f_{j+1} \dots f_i | e_1 \dots e_n e) + \quad (5)$$

$$\log p_{LM}(e_1 | e') + \sum_{n'=1}^{n-1} \log p_{LM}(e_{n'+1} | e_{n'}) + \log p_{LM}(e | e_n) \quad (6)$$

The dynamic program operation flow is as follows: 1.If there is not a gap, program will extend a similar partial translation or chose to translate a phrase from $j+1$ to i , skipping the phrase from k to j . 2.If there is a gap, program will fill in a gap in the set of translated k to j words. 3.The program repeat the operation 1,2 above.

3. (5 points) What is the complexity of your Part 2 decoder? Explain your reasoning.

Its time complexity is calculated as follows:

$$\mathcal{O}(\text{max stack size} * \text{translation options} * \text{phrases length}) \quad (7)$$

We can get the value of the max stack size is s , and the sentence length is I . And we can get the translation option through a double loop. So the time complexity of translation option is I^2 .

1.If phrases can be arbitrarily long, the complexity is shown as $\mathcal{O}(sI^2kI)$

2.So if phrases have a maximum length K , the complexity is shown as $\mathcal{O}(sK^2kI)$

4. (5 points) What is the mapping from hypothesis objects to stacks for Part 2?

There are two situations of mapping these.

- 1.If there is a gap in the hypothesis object, there are translations of exactly $i - (j - k)$ words(There are i words have been already translated except the $(j - k)$ words in gap). So they will be placed on the stack $i - (j - k)$.
2. If there is no gap. the translation of exactly i words will be placed on the stack i .

6. (15 points) Experiment with stack pruning parameters for Part 2. What is their affect on...

- a. log-probabilities: As the values of the maximum stack size (-s) and translations-per-phrase (-k) grow from 1, the value of log-probabilities grows larger. And there is an maximized log-probabilities after after the value of -s and the value of -k grow to a certain value. However, there is a special case, when the value of-s is small, and the value of k is very large, it may make the program shut down or error.
- b. speed: As the maximum stack size (-s) and translations-per-phrase (-k) start to increase from 1, the speed is slowing down.
- c. translations: As the maximum stack size (-s) and translations-per-phrase (-k) grow from 1, the translation results get better. The results of translations(using larger values of -s, -k) have changed, these results make more sense, also become more fluent. For example, yesterday I attended the first to the meeting in committee; yesterday I attended on the first meeting of the committee.(s=10,k=50 and s=100,k=100)
- d. maximum log-probability: -1300.526262

7. (10 points) Define a new dynamic program for Part 3.

To get the best partial translation of the first i French words ending in English word e . There are three ways to this situation:

The first two methods are the same as dynamic program of part2. The third method is shown below.

$$s(k, j, i, e) = \arg \max_{s(k, j, i', e')_{e_1 \dots e_n e}} \log p(s(k, j, i', e')) + \log p_{TM}(f_{i'} \dots f_i | e_1 \dots e_n e) + \quad (8)$$

$$\log p_{LM}(e_1 | e') + \sum_{n'=1}^{n-1} \log p_{LM}(e_{n'+1} | e_{n'}) + \log p_{LM}(e | e_n) \quad (9)$$

The dynamic program operation flow is almost the same as part2, except there is a difference in operation
2. If there is a gap, program will fill in a gap in the set of translated k to j words or program will extend a similar partial translation.

8. (5 points) What is the computational complexity of your Part 3 decoder?

Its time complexity is calculated as follows: (O)(max stack size * translation options * phrases length). We can get the value of the max stack size is s , and the sentence length is I . And we can get the translation option through a double loop. So the time complexity of translation option is I^2 .

- 1.If phrases can be arbitrarily long, the complexity is shown as $O(sI^2kI)$
- 2.So if phrases have a maximum length K , the complexity is shown as $O(sK^2kI)$

9. (5 points) What is the mapping from hypothesis objects to stacks for Part 3?

There are two situations of mapping these.

- 1.If there is a gap in the hypothesis object, there are translations of exactly $i - (j - k)$ words(There are i words have been already translated except the $(j - k)$ words in gap). So they will be placed on the stack $i - (j - k)$.
2. If there is no gap. the translation of exactly i words will be placed on the stack i .

11. (5 points) What is the maximum log-probability your Part 3 decoder can obtain? What do you conclude?

The maximum log-probability of my part3 decoder I can obtain is -1278.680068.

Reordering can improve the maximum log-probability and the quality of the translation.Beyond local reordering performance better than local reordering, but the speed is slower too.