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

a. log-probabilities:
When s remains unchanged, log-probability becomes bigger with the increase of k ; When k remains unchanged, the bigger the s is, the higher the log-probability is; When k and s arrive a certain number(eg, s=20,k=20), log-probability will not increase again.
b. speed:
When s remains unchanged, the bigger the k is , the slower the speed is; When k remains unchanged, the bigger the k is, The slower the speed is; The sys speed is influenced slightly, while the real and user speed is more sensitive., and s has a greater impact on speed than k.
c. translations:
As we known, lower absolute value log-probabilities means higher scores-translations. So that we should increase k and s. when s=1,k=1"i remember as the sessions that held the to the committee of veterans in February" when s=20,k=20"sessions is instead of "meetings", some of the translation really become more precise while others not.But on the whole, the translation accuracy is improved.
d. maximum log-probability:
The maximum log-probability is -1353.673782. when s=20, k=20.

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

1. $h(j,e) = arg\max_{h(i,e')e_1\cdots e_k e} \log P(h(i,e')) + \log P_{TM}(f_{i+1}.\cdots f_j|e_1\cdots e_k e) +$
$\log P_{LM}(e_1|e') + \sum_{k'=1}^{k-1} \log P_{LM}(e_{k'+1}|e_{k'}) + \log P_{LM}(e|e_k)$
where $h(j,e)$means means the best possible translation of translating French word from j and ending in e. $h(i,e')$ means that maximizing probability of the first i French words ending in e' being translated into English. P_{TM} is the probability of $i+1$to j French word being translated into English. P_{LM} is the probability of the word influenced by the prior word.

2. $s(k,j,i,e) = arg\max_{h(k,e')e_1\cdots e_n e} \log P(h(k,e')) + \log P_{TM}(f_{j+1}.\cdots f_i|e_1\cdots e_n e) +$
$\log P_{LM}(e_1|e') + \sum_{k'=1}^{n-1} \log P_{LM}(e_{k'+1}|e_{k'}) + logp_{LM}(e|e_n)$
where a partial translation that covers the first i words of the French, expect those from k to j . $s(k,j,i,e)$ means the best possible translation of k to j ending in e. $h(k,e')$ means that the best possible translation of the first k French words ending in e' , and it has already been translated. P_{TM} is the probability of $j+1$to i French word being translated into English. P_{LM} is the probability of the word influenced by the prior word.

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

$O(s*k*l*n)$
where, s is the stack size, k is translation options,and l is the length of the English sentence,and n is the maximum number of one phrase.

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

Stack[i] means translating i words of the French. For $h(i,e)$, it should be assign to the stack[s.i]. For $s(k,j,i,e)$, it should be assign to the stack[s.k+s.i-s.j].

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

a. log-probabilities:When s remains unchanged, log-probability becomes bigger with the increase of k ; When k remains unchanged, the bigger the s is, the higher the log-probability is; When k and s arrive a certain number(eg, s=60,k=20), log-probability will not increase again, and k has a greater impact on log-probabilities than s.s and k have greater influence on part2 than part1.
b. speed:
When s remains unchanged, the bigger the k is , the slower the speed is; When k remains unchanged, the bigger the k is, The slower the speed is; The sys speed is influenced slightly, while the real and user speed is more sensitive. and s has a greater impact on speed than k.

c. translations: How do they affect the translations? Do the translations change? Do they make more sense? Are they more fluent? Give examples.
As we known, lower absolute value log-probabilities means higher scores-translations. After testing, l found that increase the value k and s can increase the log-probabilities. when s=60,k=20, in part 1, the French is translated as "a committee selection was achievement", while in part 2, it become "a selection committee was achievement", we can notice that the word order is reversed. Most of the translation really become more precise and fluent than part1.Certainly, it will cause a little of them being worse.

d. maximum log-probability:
The maximum log-probability is -1300.526262, when s=60,k=20.

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

we have three state in part three, h(j, e), s(k, j, i, e) and s(k, j, m, e).

1. $h(j,e) = arg\max_{h(i,e')e_1\cdots e_k e} \log P(h(i,e')) + \log P_{TM}(f_{i+1}.\cdots f_j|e_1\cdots e_k e) +$
$\log P_{LM}(e_1|e') + \sum_{k'=1}^{k-1} \log P_{LM}(e_{k'+1}|e_{k'}) + \log P_{LM}(e|e_k)$When j=0 :

$$h(0,e) = \begin{cases} \epsilon & if\ e = START \\ undefined & otherwise \end{cases}$$

$h(j,e)$means means the best possible translation of translating French word from j and ending in e. $h(i,e')$ means that maximizing probability of the first i French words ending in e' being translated into English. P_{TM} is the probability of $i+1$to j French word being translated into English. P_{LM} is the probability of the word influenced by the prior word.

2. $s(k,j,i,e) = arg\max_{h(k,e')e_1\cdots e_n e} \log P(h(k,e')) + \log P_{TM}(f_{j+1}.\cdots f_i|e_1\cdots e_n e) +$
$\log P_{LM}(e_1|e') + \sum_{k'=1}^{n-1} \log P_{LM}(e_{k'+1}|e_{k'}) + logp_{LM}(e|e_n)$
where $s(k,j,i,e)$ means that the decoder can skip one phrase, and it must return back to translate the skipped j+1 to i French words after translate only one phrase after it. $h(k,e')$ means that the best possible translation of the first k French words ending in e' , and it has already been translated. P_{TM} is the probability of $j+1$to i French word being translated into English. P_{LM} is the probability of the word influenced by the prior word.

3. $s(k,j,m,e) = arg\max_{s(k,j,i,e')e_1\cdots e_n e} \log P(s(k,j,i,e)') + \log P_{TM}(f_{j+1}.\cdots f_i|e_1\cdots e_n e) +$
$\log P_{LM}(e_1|e') + \sum_{k'=1}^{n-1} \log P_{LM}(e_{k'+1}|e_{k'}) + logp_{LM}(e|e_n)$
$s(k,j,m,e)$ stand for the state that first i words of the French has been translated, after skipping the word from k to j, it can translate more than one words, then back to translate the skipped one.

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

$O(s*k*l*n)$
where, s is the stack size, k is translation options,and l is the length of the English sentence,and n is the maximum number of one phrase.

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

Stack[i] means translating i words of the French. For $h(i,e)$, it should be assign to the stack[s.i]. For $s(k,j,i,e) and s(k,j,m,e)$, it should be assign to the stack[s.k+s.m-s.j] and stack[s.k+s.m-s.j].

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

The maximum log-probability is -1278.680068 in part 3 when s=800, k=20. Similarly with part 1 and part 2,with the increase of k and s, The log-probability will increase and the decoder's speed will be slower. and after arrived at a certain number, log-probability won't increase again. Although the final log-probability of part 3 is better than part2, But l find that, sentence in part3 become less fluent. For example, in part 2" honorable senators, what happened here ,last Tuesday ?" but in part3"honorable senators here,last Tuesday , what happened ?". l guess that the system is over fitting in order to get the larger log-probability.

Table 1: different parameter setting in part1

s	k	real	user	sys	log-probability
1	1	0m1.511s	0m1.337s	0m0.101s	-1439.873990
1	10	0m1.552s	0m1.390s	0m0.101s	-1375.922818
1	20	0m1.595s	0m1.482s	0m0.084s	-1374.622653
1	30	0m1.754s	0m1.577s	0m0.088s	-1374.622653
1	50	0m1.727s	0m1.615s	0m0.089s	-1374.622653
10	20	0m2.790s	0m2.629s	0m0.102s	-1354.269108
20	20	0m4.395s	0m4.127s	0m0.137s	-1353.673782
50	20	0m7.436s	0m7.018s	0m0.154s	-1353.673782
100	20	0m10.212s	0m9.934s	0m0.152s	-1353.673782

Table 2: different parameter setting in part2

s	k	real	user	sys	log-probability
1	1	0m1.236s	0m1.172s	0m0.057s	-1651.641211
1	10	0m1.525s	0m1.431s	0m0.080s	-1593.042204
1	20	0m1.949s	0m1.672s	0m0.127s	-1590.923218
1	50	0m1.894s	0m1.894s	0m0.111s	-1590.923218
1	1000	0m2.056s	0m1.921s	0m0.098s	-1590.923218
20	20	0m4.998s	0m4.854s	0m0.092s	-1319.811548
50	20	0m11.366s	0m11.366s	0m0.163s	-1304.580618
60	20	0m13.096s	0m12.568s	0m0.181s	-1300.526262
100	20	0m20.953s	0m20.953s	0m0.274s	-1300.526262
200	20	0m37.725s	0m36.882s	0m0.318s	-1300.526262

Table 3: different parameter setting in part3

s	k	real	user	sys	log-probability
1	1	0m1.830s	0m1.342s	0m0.107s	-1638.406854
1	10	0m1.614s	0m1.459s	0m0.097s	-1583.330253
1	20	0m1.658s	0m1.490s	0m0.095s	-1583.236256
1	50	0m1.856s	0m1.686s	0m0.097s	-1583.236256
20	20	0m6.302s	0m5.955s	0m0.144s	-1328.439987
100	20	0m25.055s	0m24.046s	0m0.292s	-1284.625401
600	20	2m4.709s	2m3.277s	0m0.608s	-1279.951427
800	20	2m59.057s	2m52.373s	0m1.807s	-1278.680068
1000	20	3m38.160s	3m31.973s	0m1.933s	-1278.680068