Write up - Decoder

Partners:

Lionel Eisenberg **leisenb5**

Sanat Deshpande **sdeshpa4**

## Base decoder

Here are the results that we observed when running the stack decoder that was given to us on the data on stacks of size 1 – 10000 and of k translations per phrase of 1 – 10.

|  |  |  |
| --- | --- | --- |
| Stack Size / number translations per phrase | 1 | 10 |
| 1 | -1439.873990 | -1375.922818 |
| 10 | -1436.360138 | -1354.642177 |
| 100 | -1436.360138 | -1354.642177 |
| 1000 | -1436.360138 | -1354.642177 |
| 10000 | -1436.360138 | -1354.642177 |

We observe that as the size of the stack and the amount of translations per phrase increases, we get better and better scores. This is completely what we would expect as, as the number of stacks increases, we can computer more hypotheses, and as the number of translations increases we have more possibilities in our hypotheses. Now of course that can only help us so far as there comes a point where those parameters are maxed out.

## Beam Decoder with swapping

Here we implemented an addition to the baseline decoder that was given to us. To do this we just added an extra algorithmic layer to our passthrough. Essentially, in addition to checking every possible substring to add to a hypothesis and stack we made sure to add the adjacent sentences and swap them. This increases our search space, and therefore bettered our scores. Indeed, if we look at the following results:

|  |  |  |
| --- | --- | --- |
| Stack Size / number translations per phrase | 1 | 10 |
| 1 | -1404.153329 | -1374.174516 |
| 10 | -1384.609459 | -1350.207840 |
| 100 | -1384.609459 | -1350.207840 |
| 1000 | -1384.609459 | -1350.207840 |
| 10000 | -1384.609459 | -1350.207840 |

It is clear that this model will perform better as the base decoder since it adds more possibilities and hypotheses to our final stack.