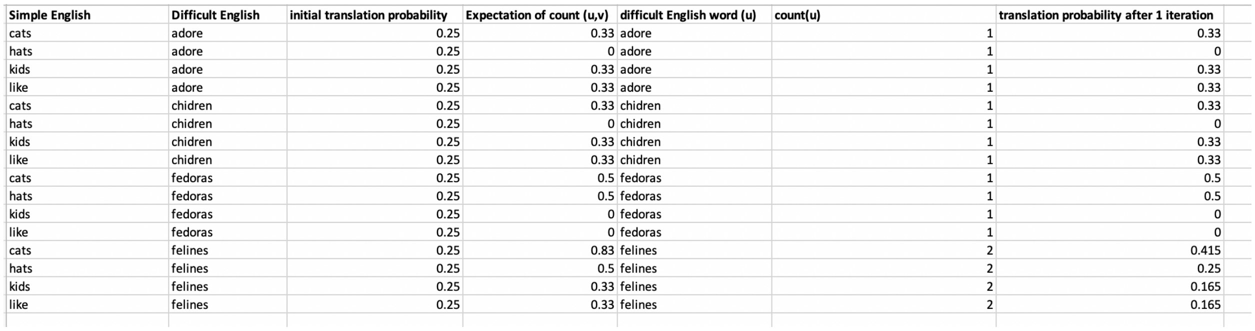
Answer to the question 1

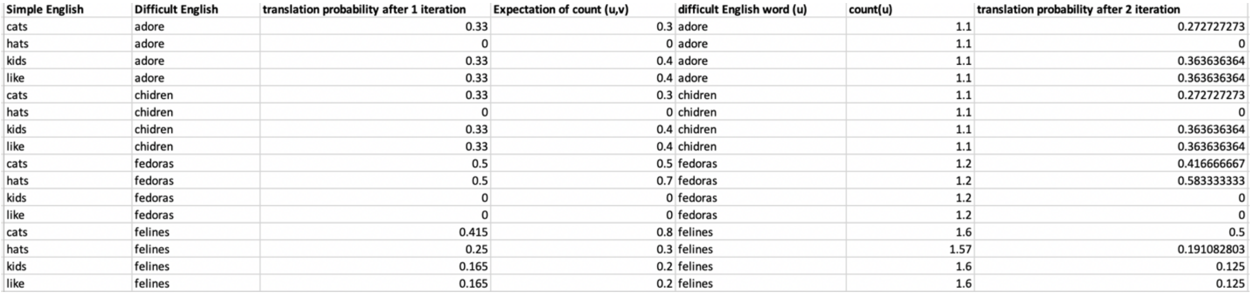
Following the textbook (Einstein) notation. I computed the 2 iterations of the algorithm below. By u I denote the target word and by v I denote the source word. One note about the column *difficult English word (u)* is that here I included duplicate word just to make the microsoft excel computation easier.

First iteration:



Simple English	Difficult English	initial translation probability	Expectation of count (u,v)	difficult English word (u)	count(u)	translation probability after 1 iteration
cats	adore	0.25	0.33	adore	1	0.33
hats	adore	0.25	0	adore	1	0
kids	adore	0.25	0.33	adore	1	0.33
like	adore	0.25	0.33	adore	1	0.33
cats	chidren	0.25	0.33	chidren	1	0.33
hats	chidren	0.25	0	chidren	1	0
kids	chidren	0.25	0.33	chidren	1	0.33
like	chidren	0.25	0.33	chidren	1	0.33
cats	fedoras	0.25	0.5	fedoras	1	0.5
hats	fedoras	0.25	0.5	fedoras	1	0.5
kids	fedoras	0.25	0	fedoras	1	0
like	fedoras	0.25	0	fedoras	1	0
cats	felines	0.25	0.83	felines	2	0.415
hats	felines	0.25	0.5	felines	2	0.25
kids	felines	0.25	0.33	felines	2	0.165
like	felines	0.25	0.33	felines	2	0.165

Second iteration:



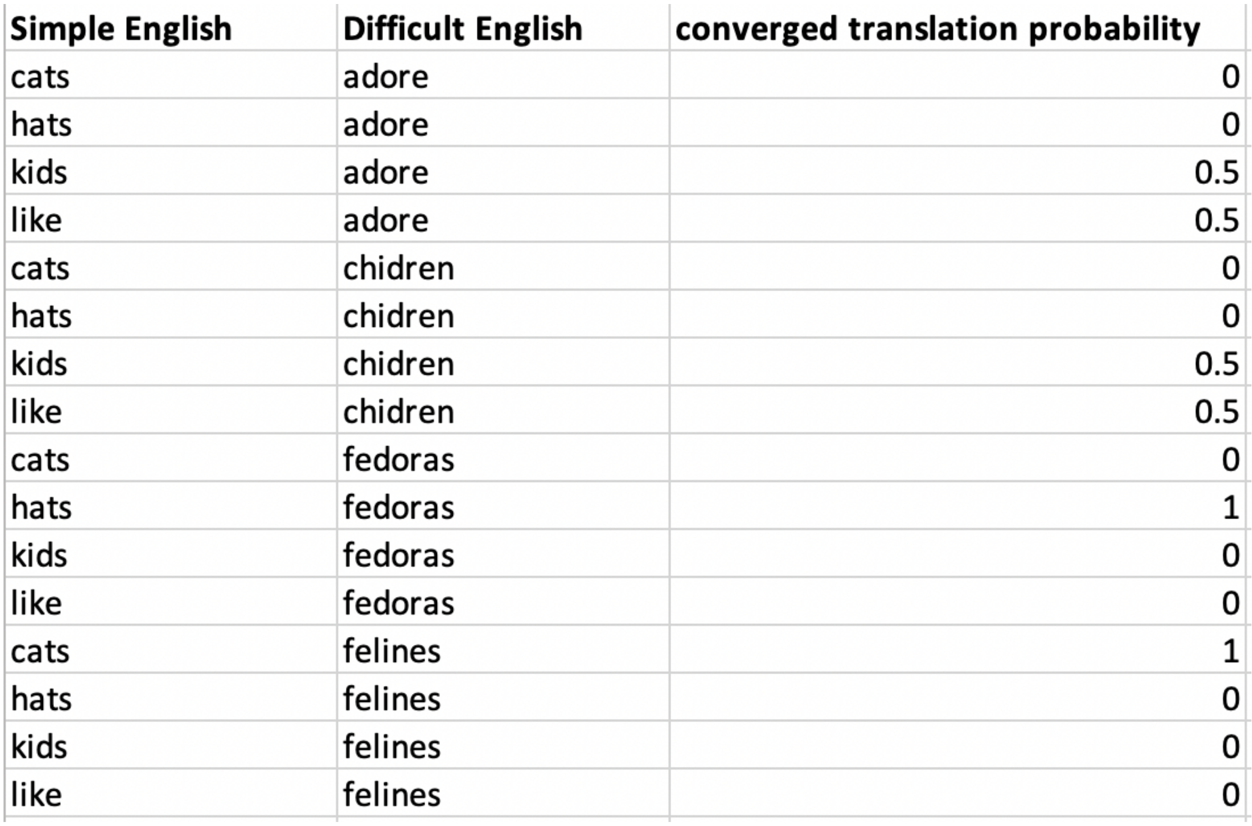
Simple English	Difficult English	translation probability after 1 iteration	Expectation of count (u,v)	difficult English word (u)	count(u)	translation probability after 2 iteration
cats	adore	0.33	0.3	adore	1.1	0.272727273
hats	adore	0	0	adore	1.1	0
kids	adore	0.33	0.4	adore	1.1	0.363636364
like	adore	0.33	0.4	adore	1.1	0.363636364
cats	chidren	0.33	0.3	chidren	1.1	0.272727273
hats	chidren	0	0	chidren	1.1	0
kids	chidren	0.33	0.4	chidren	1.1	0.363636364
like	chidren	0.33	0.4	chidren	1.1	0.363636364
cats	fedoras	0.5	0.5	fedoras	1.2	0.416666667
hats	fedoras	0.5	0.7	fedoras	1.2	0.583333333
kids	fedoras	0	0	fedoras	1.2	0
like	fedoras	0	0	fedoras	1.2	0
cats	felines	0.415	0.8	felines	1.6	0.5
hats	felines	0.25	0.3	felines	1.57	0.191082803
kids	felines	0.165	0.2	felines	1.6	0.125
like	felines	0.165	0.2	felines	1.6	0.125

Answer to the question 2

From response to the question 3, we have found that for (hats, fedoras) and (cats, felines) the probabilities are increasing as expected.

However, for between (kids, children), (like, adore), (like, children), (kids, adore) the probabilities are also increasing by same amount though the latter two pairs are wrong translation. Which means for these 4 tuples the translation model is confused.

Based on the above observations, following will be the converged translation probability:



Simple English	Difficult English	converged translation probability
cats	adore	0
hats	adore	0
kids	adore	0.5
like	adore	0.5
cats	chidren	0
hats	chidren	0
kids	chidren	0.5
like	chidren	0.5
cats	fedoras	0
hats	fedoras	1
kids	fedoras	0
like	fedoras	0
cats	felines	1
hats	felines	0
kids	felines	0
like	felines	0

From the above table we can see that the translation model failed here. It can translate (hats, fedoras) and (cats, felines) but it cannot translate the other words.

The model is confused between (kids, children) and (like, adore). This is because in the dataset we have only one sentence of these two pairs. Consequently, the translation model cannot distinguish between them. Based on this observation, if a dataset does not have multiple sentences for same pair of word it will fail in the way it fails here.

Answer to the Question 3

Here $\lambda_{j+1}^{(t)}$ is the loss at word $j+1$ of the target. We need to find the derivative $\frac{\partial \lambda_{j+1}^{(t)}}{\partial h_n^{(s)}}$ where $h_n^{(s)}$ is the hidden state at word n of source. So using the chain rule we get:

$$\frac{\partial \lambda_{j+1}^{(t)}}{\partial h_n^{(s)}} = \frac{\partial \lambda_{j+1}^{(t)}}{\partial \lambda_j^{(t)}} \frac{\partial \lambda_j^{(t)}}{\partial h_{j-1}^{(t)}} \cdots \frac{\partial \lambda h_{n+1}^{(s)}}{\partial \lambda h_n^{(s)}}$$

The above equation represents that we need to apply chain rule from the loss of current target word to the loss of the $n+1$ source word.

To find the shortest backpropagation path we need to consider the length of the sequences. specifically, we need to consider j term of the target sequence and $m^{(s)}-n$ term of the source sequence (assuming $m^{(s)}$ is the length of source sequence.

So total terms on the shortest backpropagation $= m^{(s)}-n+j$

Answer to the Question 4

For the attention model, the derivative
$$\frac{\partial \lambda^{(t)}_{j+1}}{\partial z_n} = \frac{\partial \lambda^{(t)}_{j+1}}{\partial c_j} \frac{\partial c_j}{\partial z_n}$$

Here c_j is taken from the textbook (Einstein's book equation 18.39),

where $c_j = \sum_{n=1}^{M^{(s)}} \alpha_{j\to n} z_n$

So, $$\frac{\partial \lambda^{(t)}_{j+1}}{\partial z_n} = \frac{\partial \lambda^{(t)}_{j+1}}{\partial c_j} \frac{\partial \sum_{n'=1}^{m^{(s)}} \alpha_{j\to n'} z_{n'}}{\partial z_n}$$

As we are differentiating the second term with respect to z_n, in the shortest path we don't need to consider $n' = n$. So, the shortest path $= \frac{\partial \lambda^{(t)}_{j+1}}{\partial c_j} \frac{\partial \sum_{n' \neq n}^{M^{(s)}} \alpha_{j\to n'} z_{n'}}{\partial z_n}$