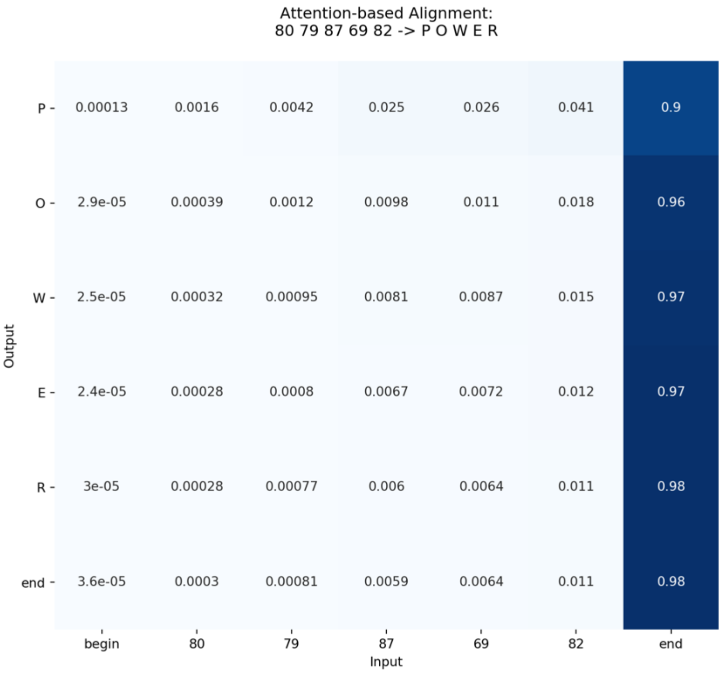
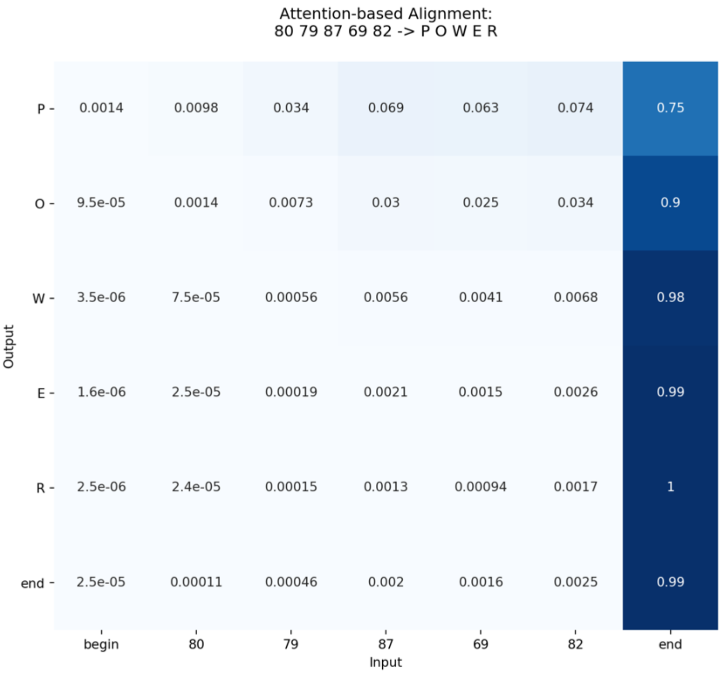
Part 3 – Analysis

Epoch no. 1



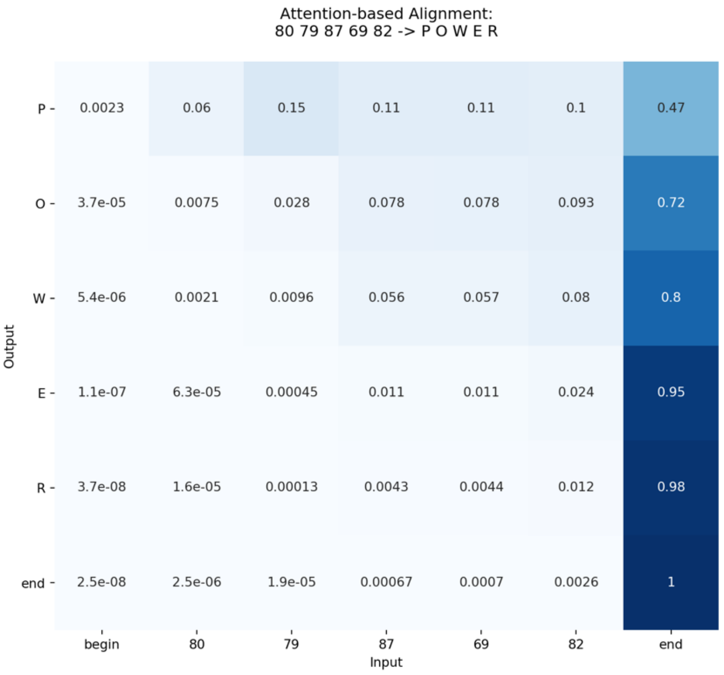
	begin	80	79	87	69	82	end
P	0.00013	0.0016	0.0042	0.025	0.026	0.041	0.9
O	2.9e-05	0.00039	0.0012	0.0098	0.011	0.018	0.96
W	2.5e-05	0.00032	0.00095	0.0081	0.0087	0.015	0.97
E	2.4e-05	0.00028	0.0008	0.0067	0.0072	0.012	0.97
R	3e-05	0.00028	0.00077	0.006	0.0064	0.011	0.98
end	3.6e-05	0.0003	0.00081	0.0059	0.0064	0.011	0.98

Epoch no. 2



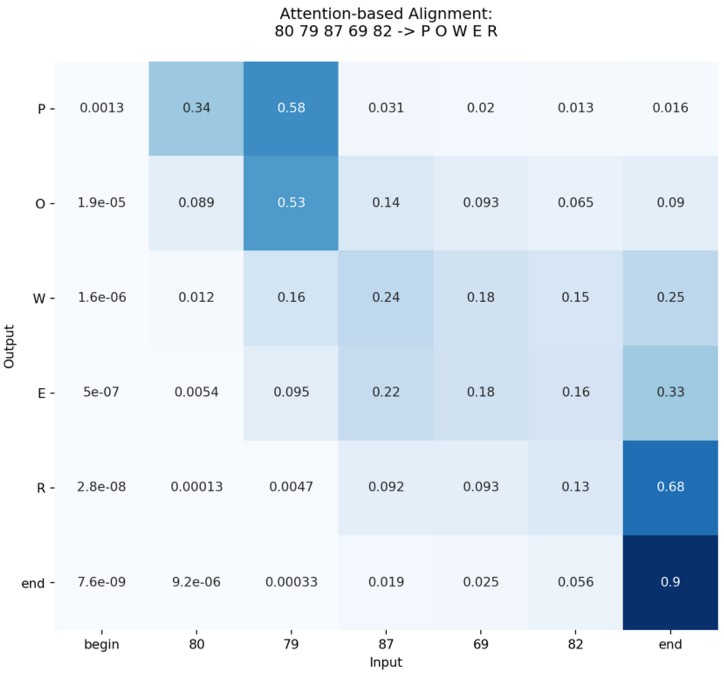
	begin	80	79	87	69	82	end
P	0.0014	0.0098	0.034	0.069	0.063	0.074	0.75
O	9.5e-05	0.0014	0.0073	0.03	0.025	0.034	0.9
W	3.5e-06	7.5e-05	0.00056	0.0056	0.0041	0.0068	0.98
E	1.6e-06	2.5e-05	0.00019	0.0021	0.0015	0.0026	0.99
R	2.5e-06	2.4e-05	0.00015	0.0013	0.00094	0.0017	1
end	2.5e-05	0.00011	0.00046	0.002	0.0016	0.0025	0.99

Epoch no. 3



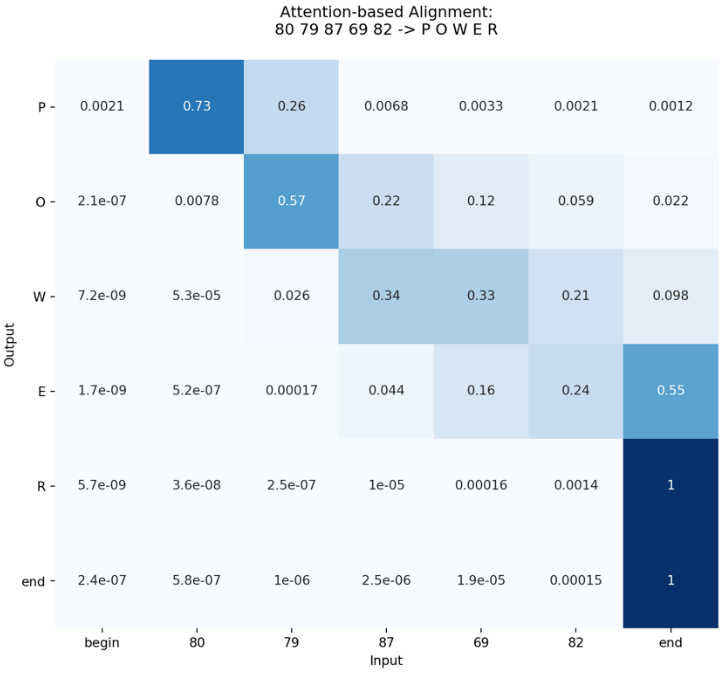
	begin	80	79	87	69	82	end
P	0.0023	0.06	0.15	0.11	0.11	0.1	0.47
O	3.7e-05	0.0075	0.028	0.078	0.078	0.093	0.72
W	5.4e-06	0.0021	0.0096	0.056	0.057	0.08	0.8
E	1.1e-07	6.3e-05	0.00045	0.011	0.011	0.024	0.95
R	3.7e-08	1.6e-05	0.00013	0.0043	0.0044	0.012	0.98
end	2.5e-08	2.5e-06	1.9e-05	0.00067	0.0007	0.0026	1

Epoch no. 4

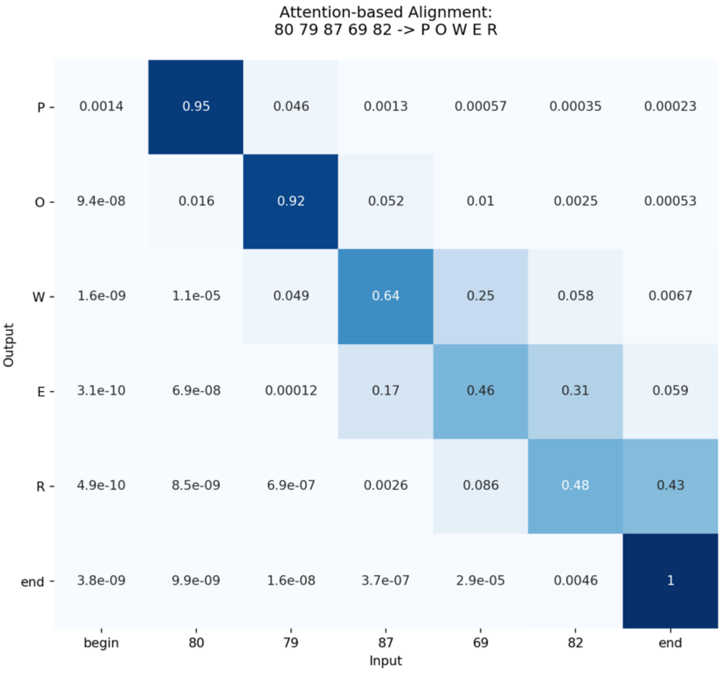


	begin	80	79	87	69	82	end
P	0.0013	0.34	0.58	0.031	0.02	0.013	0.016
O	1.9e-05	0.089	0.53	0.14	0.093	0.065	0.09
W	1.6e-06	0.012	0.16	0.24	0.18	0.15	0.25
E	5e-07	0.0054	0.095	0.22	0.18	0.16	0.33
R	2.8e-08	0.00013	0.0047	0.092	0.093	0.13	0.68
end	7.6e-09	9.2e-06	0.00033	0.019	0.025	0.056	0.9

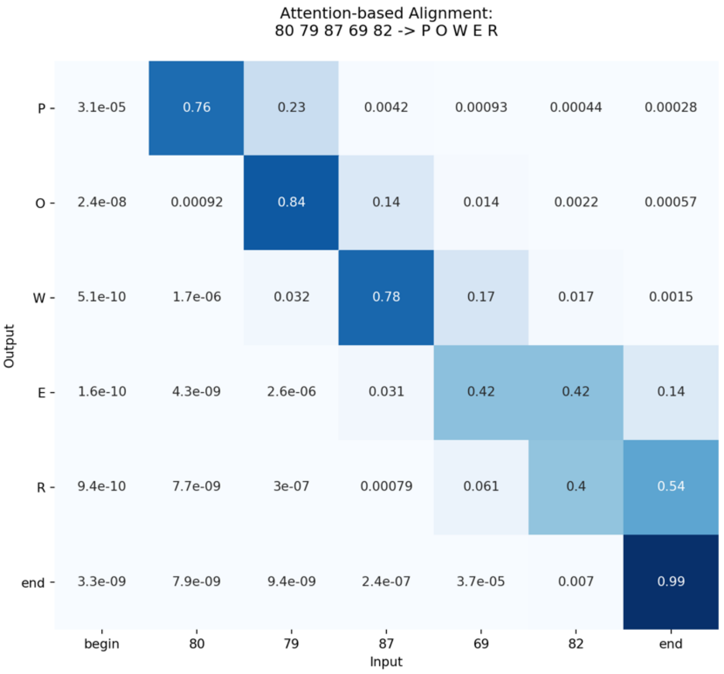
Epoch no. 5



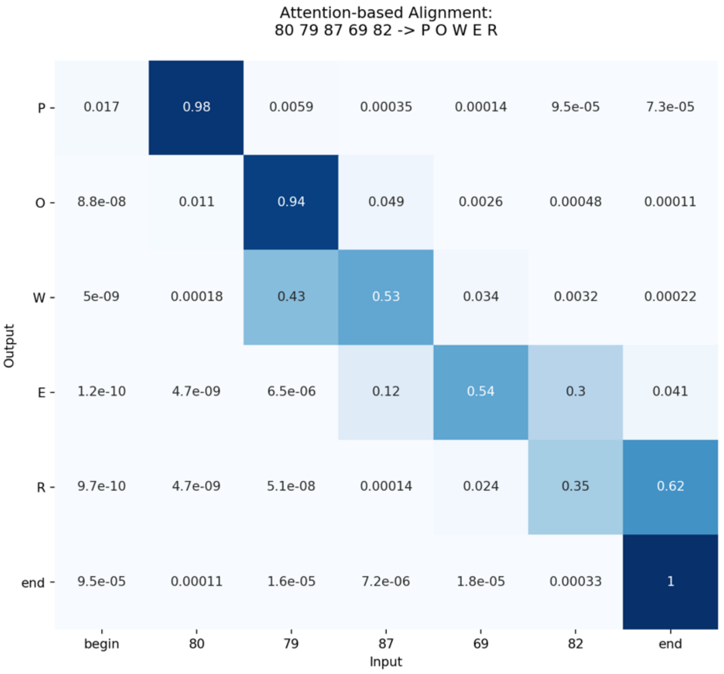
Epoch no. 6



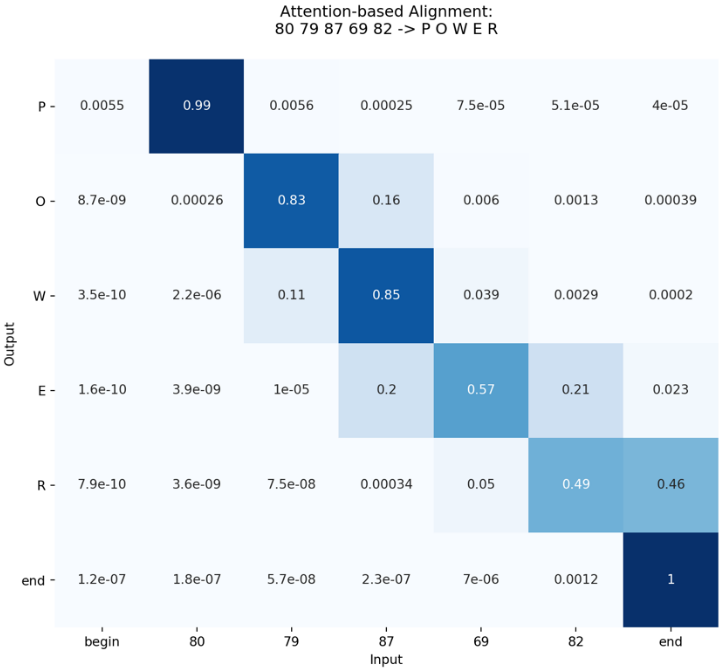
Epoch no. 7



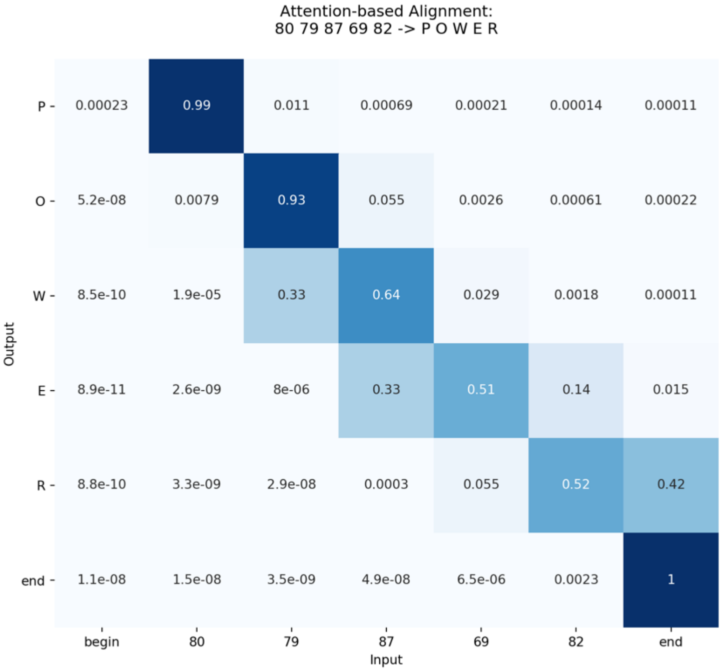
Epoch no. 8



Epoch no. 9



Epoch no. 10



Chosen word from the development set: **no. 41 - POWER**

Describe how the attention visualization changes during training:

In the earliest epochs (1-3) we can see that the input `</s> (end) token gets almost all the attention on each decoder step. On the first epoch it's because the context vector that was concatenated to the lstm input was zeros vector. On the 4th and 5th epochs we can see that other input tokens are getting more attention now in the calculation of the attention weights in each decoder step. Which can be seen as progress in the right direction in which we aim that the input token that will get most of the attention in the current decoder step is the one that in this step the decoder has to predict its translation token. From epoch 6 and on, we can see that as we expected in each decoder step, the input token that gets most of the attention is the one that now the decoder has to predict its translation and the rest of the attention is given to the input tokens surrounding it.

How would you explain that?

It’s basically the attention mechanism that helps the decoder “pay attention” to the most relevant tokens in the input sentence so that they will help him to predict the next correct token. This mechanism, with proper calculation of the attention weights of course, helps the model give more attention to the relevant input token the decoder should predict its translation token in the current decoder step.