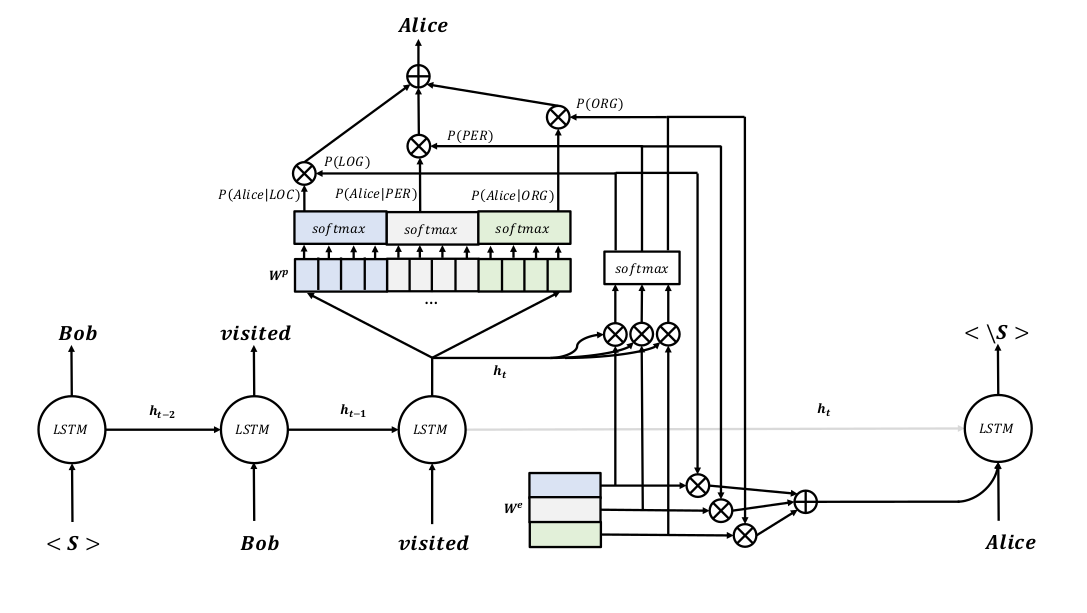
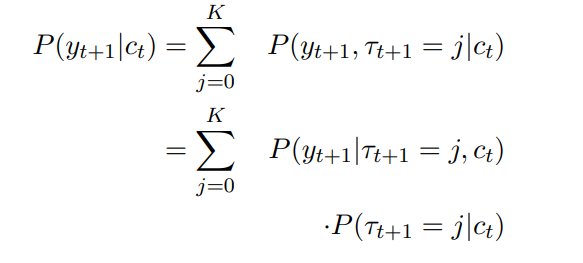
**Knowledge Augmented Language Modelling**

Implemented a KALM based on the paper <https://arxiv.org/abs/1904.04458> The architecture we implemented is as follows..

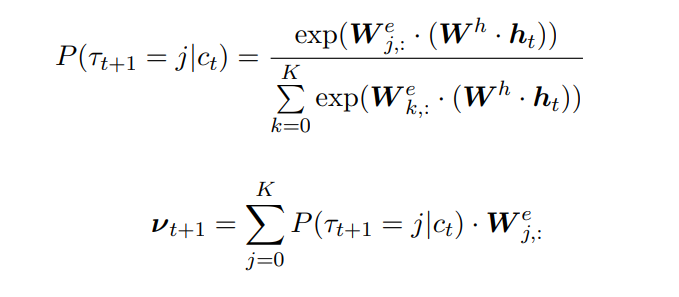


We implemented the Whole architecture using various classes. The classes is as described below.

**Projection Layer:** This class calculates the probability of next word given the history. It implements the below equation**.** It takes the input as hidden state and type probabilities.



**Type Embedding Layer:** This class is the implementation for the below equation. We take the output of previous layer and apply Dense layer (Wh) to reduce the dimensionality and then apply another Dense layer of We as per the equations. Please see the class code for the more details.



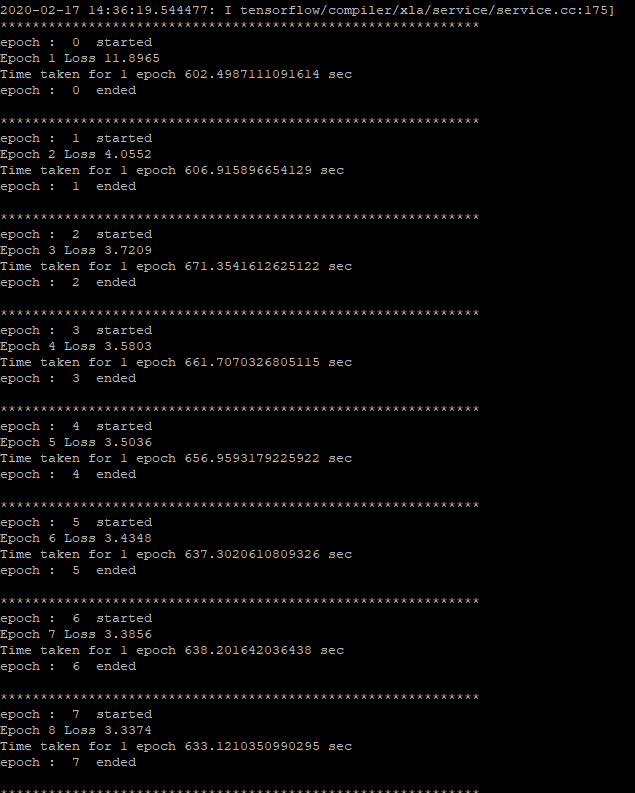
**Decoder Layer:** This is the final class which implements the whole model. At First, we initialize the hidden state and then pass it to the first-time step with the input. Herewith the flow of code for this class**.**

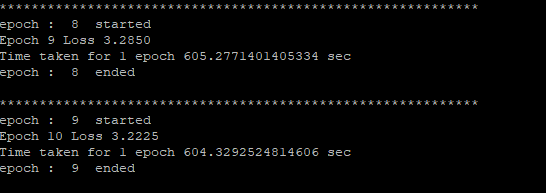
1. **Initialize the hidden state ht.**
2. **Pass the initialized hidden state to the Type Embedding class to predict the type probabilities.**
3. **Pass the hidden state with type probabilities to the Projection Layer class to get the final output (Predicted word)**
4. **Repeat the steps 1,2 and 3 for all time steps.**

**Implementation Details:**

We have used the 3 stacked LSTM layers with the other classes mentioned above. No dropout was used. Adam optimizer is used for optimization.

**Results:** Loss reduced over the epoch. The reduction in loss was faster model was found. Please check below for more details.





**Disclaimer:**

1. **We have used the same number of units for the Wh and We to simplify the implementation.**
2. **Simple LSTM was used instead of AWD-LSTM as mentioned in the paper.**
3. **No dropouts were used as per the paper.**

**Prediction:**

We tried to predict the next word for the given input word, but the model was always predicting the zero index as start word.