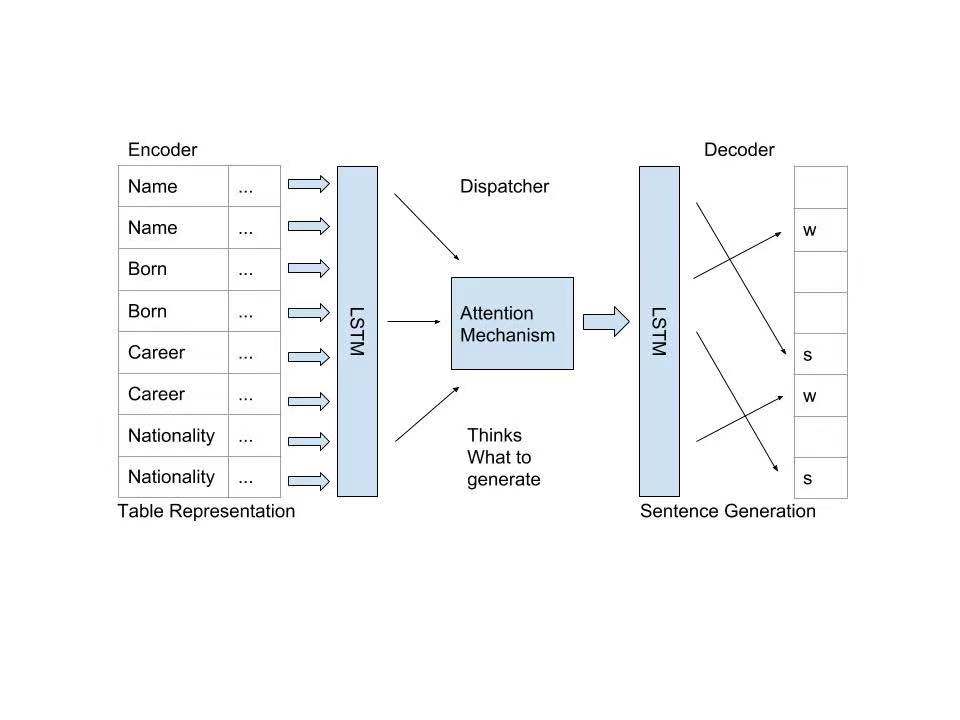
**System Diagram**

****

**Build Instructions and README**

* Their project approach for generating summarization consist three parts.
* An encoder that captures the table forms of information and go through LSTM.
* A dispatcher that applies attention mechanism and plans what to generate.
* A decoder that generates the output summarization using RNN.
* Readme is well-structured format but lack of information on instructions. It seems like it’s still under process so it’s understandable.

**Technology Selection**

* Python 3
* Tensorflow-Recurrent Neural Network
* Google Collaboratory for ML environment
* Long Short-Term Memory (LSTM):
  + This is actually a very popular Tensorflow RNN implementation for solving sequential problems by its spectacular encoder/decoder design.
  + They mainly implemented the seq2seq model for their project. This is a general-purpose encoder-decoder framework for Tensorflow. It’d be the best fit for processing huge data such as text summarization.
* Attention mechanism
* By implementing this attention could bring a significant for this project.

**Issues**

* Since LSTM is a takes the input and covert them into a vector forms for decoder to output. I didn’t see much related explanation or clear implementation on word embedding system. It seems like some very complicated matrix methods. However, I’ve found a pre-trained vector word-embedding package called “GloVe” which can be found <https://nlp.stanford.edu/projects/glove/>. Not entirely sure if this can be fir for your project but definitely worth to consider.
* Codes and implementation seem very vague to me. I found it very hard to use your code to test run your project. Maybe it’s still incomplete at the moment or lack of instructions.
* As far as I investigate the codes, though the system seems very reliable, there aren’t enough clear instructions on how to address errors or prevent errors methods.

**Positive things I like about this project**

* The direction of their research on this project is very legitimate.
* By implementing attention mechanism could bring a significant effect for their summarization. It computes the attention weighs not only selecting important clues of the words to generate, but also directly links the relationship between different fields which enhance the efficiency and accuracy of the summarization.
* Good dataset selection, it has enough resources to train the model.
* Contained clear comments in codes.
* Very realistic and useful project that can help a lot of people in the future.