

# 1 Implementation

For the implementation part of our project, we have decided to make a quote generator using RNNs. For this implementation, we used the Pytorch library because we found it more intuitive and easier to debug than other frameworks. The dataset we used for the training is from Kaggle and is composed of 36.2k quotes [1]. First, we build our dictionary by extracting all the different words from our dataset. We then create an embedding layer which is responsible to learn a vector representation for each of our words. We have integrated the possibility to initialize our embedding layer with a pretrained GloVe model if we wish to [2]. We are feeding sequences of 4096 encoded words to 4096 LSTM units with a hidden state of 100 dimensions. The output of each LSTM unit is then fed in a linear layer before passing through a softmax activation for the identification of the predictions. We are using the cross-entropy loss as objective to optimize, thus maximizing the loglikelihood of the training data. For the optimization technique, we used the Adam optimizer, since it seems to be the state of the art now [3]. To generate text, we initialize the network with a seed word, in our case it was 'What', and then feed back the predictions in the network in loops. To incorporate randomness in the generated text, we choose the predicted word by defining a multinomial distribution on the output of the softmax layer. Here's an example of generated text: "What is the best thing that has arisen from me is coming out of mental exaltation."

## References

- [1] Quotes dataset. <https://www.kaggle.com/coolcoder22/quotes-dataset/home>.
- [2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.