1. What hyperparameters are used in SHA-RNN ?

= The hyperparameters that are used in SHA-RNN are listed below:

RNN cell, layers, all hidden sizes,put embedding size, boom hidden size, dropout(e/h/i/o), optimizer, weight decay, BPTT length, memory window size, batch size and learning rate.

2. What is language modeling and how LSTM is used in language modeling ?

= Language modeling is one of the foundational tasks of natural language processing. It is the task of predicting what word comes next. Using various statistical and probabilistic techniques a language model determines the probability of a given sequence of words occurring in a sentence. Alternatively, we can say that it is the task that involves predicting the (n+1)th token in a sequence given the n preceding tokens. Language models analyze large bodies of text data to provide a basis for their word predictions. Language modeling is used in natural language processing (NLP) applications like your phone’s autosuggest, search engines, etc.

LSTM (long short-term memory) is a type of RNN that was proposed as a solution to the vanishing gradient problem in vanilla RNNs. This problem arises in a RNN when there is a large amount of information and the gradient becomes vanishingly small over longer distances and thus resulting in inaccurate predictions. The main problem is that it’s too difficult for the RNN to learn to preserve information over many timesteps as the hidden state in a vanilla RNN is constantly being rewritten. As a result, an RNN with separate memory i.e., LSTM was proposed. LSTMs are used for the connection of hidden layers. It is designed for remembering the long-term memory, so that it should be able to consider relationships of distant words, such that a word at beginning of sentence and it at the end. The gates in LSTMs can alter which parts of the text to remember and which to drop resulting in a much more accurate prediction. Although excellent performance is obtained for large vocabulary tasks, tremendous memory consumption prohibits the use of LSTM language model in low-resource devices. The memory consumption mainly comes from the word embedding layer.

3. What datasets authors used to evaluate their experiments?

= The author used 3 datasets to evaluate his experiment. They are: the byte-level enwik8 dataset, the WikiText-103 dataset and the WikiText-2 dataset.

4. What do you understand by the BOOM LAYER ? How we can mitigate it?

= The Boom Layer is a modified feedforward layer that is similar to a large feedforwards layer found in Transformers and other architectures. The layer takes a vector of the form v ∈ R^H and uses a matrix multiplication with GeLU activation to produce a vector u ∈ R^(N×H). Then u is broken into N vectors which are summed together to produce a vector w ∈ R^H. This minimizes computation and removes an entire matrix of parameters compared to traditional down-projection layers.

5. What is tokenization attacks ? Explain with some examples by authors point of view?

= In cryptography, a timing attack is a side channel attack where a timing signal (in our case “tokenization” of a temporal sequence of operations) leaks information. As we’re already using the SHA-RNN and there are crude comparisons to timing attacks in cryptography we will refer to this as a tokenization attack. If we check our password a character at a time, stopping when a character doesn’t match, we know how much of our candidate password is correct and can search through the remaining possibilities. Each token reveals (leaks) far more information. These tokenization attacks are a natural result of our language models accessing and relying on side-channel information that researchers may not be aware they’re providing.

According to the author when you need to choose between "specialised" and "specialized," which appear 403 and 1091 times in the training data, respectively, a word level model will have a far higher confusion than if your wordpiece model splits them down into specialized or similar. The same is true when copying words and named entities, with attention processes assisting greatly. If we're responding to the question, "Who was killed?" With a word-level model, we only have one guess: Dumbledore or Gandalf. However, a wordpiece model provides a significant advantage because entropy can be put primarily on the first token (i.e. Dumbledore).