4 Dataset

For our experiments, we make use of a dataset composed of Reber sequences. A Reber sequence is a grammar string made of finite states, in simple words, formed by using a confined set of characters. In the research paper that proposed the LSTM [HS97], the authors use embedded Reber grammar due to its short time lags. For our purpose, we use a simple version of this embedded Reber grammar, called Reber grammar ([Reb76]).

The following figure shows the flow diagram to generate Reber grammar sequences:

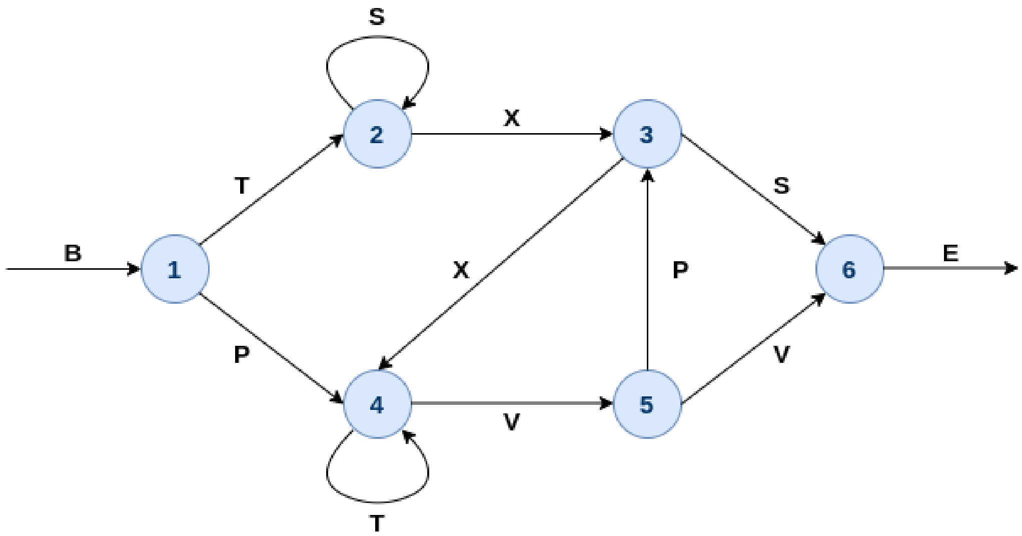

Figure 4.1: Flow diagram to generate Reber grammar sequences

As shown in the above graph, a sequence is a true Reber sequence if

1. it contains only these characters: *B, E, P, S, T, V, X*
2. it starts with a *B* and ends with an *E*, *always!*
3. it strictly follows the transition diagram shown in figure 4.1.

After starting at B, we move from one node to the next until we reach E. If we have more than one path to choose from, we can randomly select one of them. Two of the allowed characters, S and T, have a self-loop at nodes 2 and 4, respectively, meaning we can have more than two consecutive Ss and Ts. For example, the following table shows some of the true and false Reber sequences from our dataset:

True Reber sequences	**False Reber sequences**
BPTVPXTTVVE	BTTVPXTVPSE
BTSXXTTTTVVE	BPSXXTTTVPSE
BTSSXXTTVVE	BPSSSXXTVVE
BTXXVPXTVVE	BTTTVPXTVVE
BPTTTTTVPSE	BPSSXXTTVVE

Table 4.1: A few examples of true and false Reber sequences.

Since, to our knowledge, there is no public dataset of Reber sequences available, we created our own dataset of 25000 true and false Reber sequences combined. We follow the same flow, as shown in figure 4.1, to generate true Reber sequences. Then, by slightly altering the same procedure, we generate false Reber sequences.

We then split this dataset into a train-test set, details of which are shown in the following bar chart:

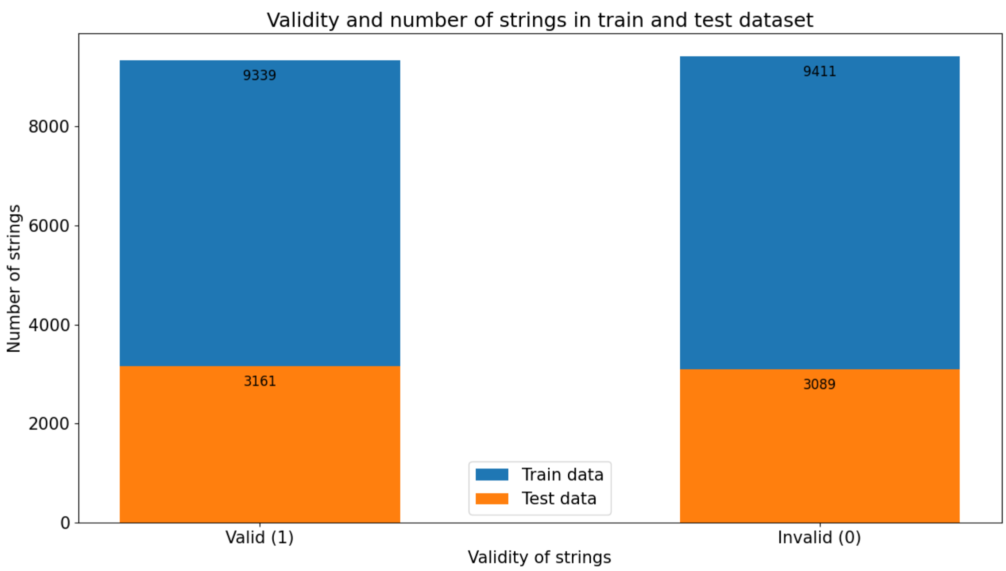

Figure 4.2: Number of valid (true) and invalid (false) Reber sequences in our train-test set

Our dataset has 12500 true Reber sequences and 12500 false, thus totaling 25000. Although the true Reber sequences only contain allowed characters (i.e., *B, E, P, S, T, V, X*), the false sequences may contain any character from A to Z.

Out of 25000 total sequences, our train-test set contains 18750 and 6250 sequences, respectively. Furthermore, this train-test individually include the following number of true and false Reber sequences:

	Training set	**Test set**
Valid (true)	9339	3161
Invalid (false)	9411	3089

Table 4.2: The number of true and false Reber sequences in our train-test set.

Next, we visualize the string length distribution that shows not only the minimum and maximum string length but also the number of strings for each length value.

The following plot displays the string length distribution of our entire dataset:

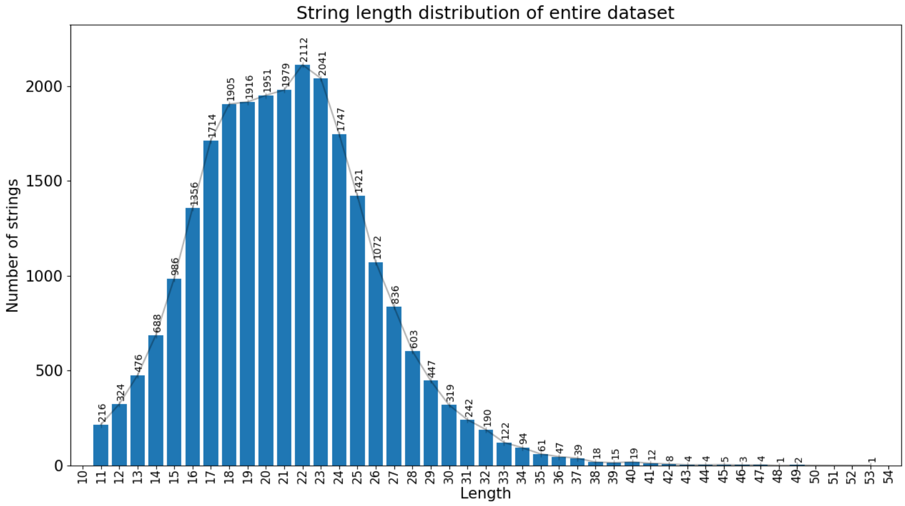

Figure 4.3: String length distribution of entire dataset

As we can see in the above plot, in our entire dataset, the minimum string length is 11 with 216 sequences, and the maximum string length is 53 with 1 sequence. The highest number of strings, 2112 strings, in our entire dataset is of length 22.

The following plot displays the string length distribution of our train dataset:

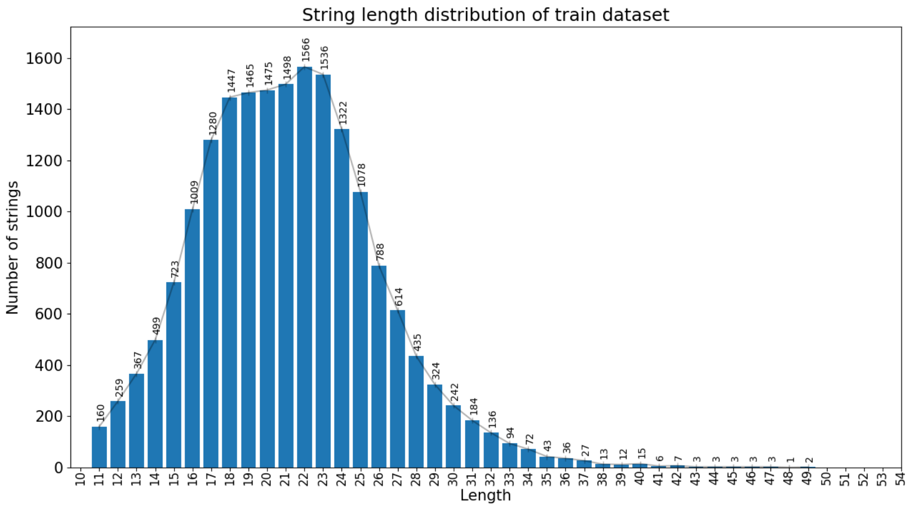

Figure 4.4: String length distribution of train dataset

As we can see in the above plot, in our train dataset, the minimum string length is 11 with 160 sequences, and the maximum string length is 49 with 2 sequence. The highest number of strings, 1566 strings, in our train dataset is of length 22.

The following plot displays the string length distribution of our test dataset:

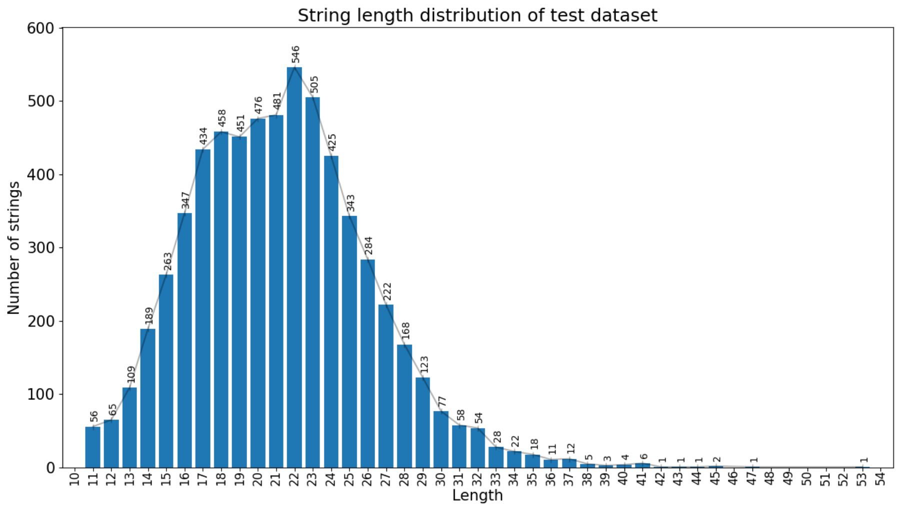

Figure 4.5: String length distribution of test dataset

As we can see in the above plot, in our test dataset, the minimum string length is 11 with 56 sequences, and the maximum string length is 53 with 1 sequence. The highest number of strings, 546 strings, in our test dataset is of length 22.