**Amoghaya Bhatia**

**Mobile No.: +91 9606862148**

**Email Handle: amoghayabhatia.456@gmail.com**

**Trexquant Hangman Challenge**

The aim of the given challenge was to play the game of Hangman with a server. The server would send a word, with each letter depicted as a blank space, and we as the opponent would have to guess the letters. The game ended if we either guessed the word correctly within the allotted number of attempts (six) or exhausted them.

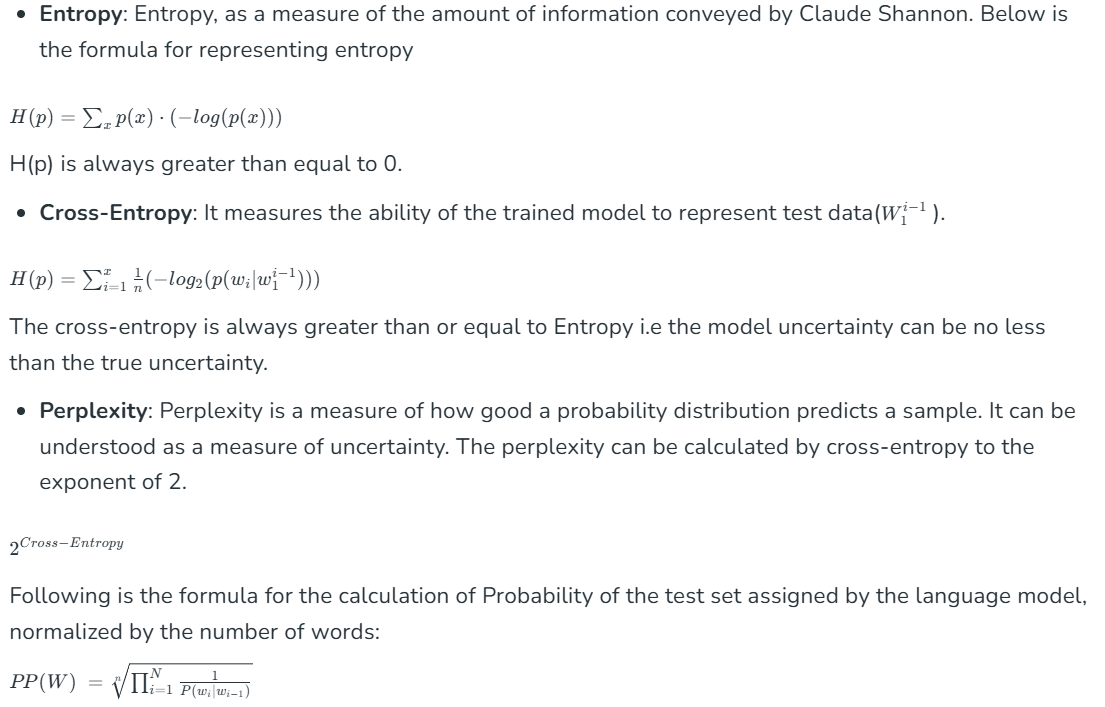
**Approach:**

My initial approach was to train a Recurrent Neural Network (RNN) and have it guess the words with a high accuracy. However, this was proving to be a cumbersome task, owing to a high number of technical difficulties such as resolving dependencies within installed packages.

In order to overcome these issues, I chose a more statistical approach – The “**N-grams method**”.

The N-grams method is a probabilistic method for guessing the next **gram** (a letter or a word) in a sequence from its context (the words or letters that have been encountered so far). This method works with a high accuracy because it tabulates the number of times a certain **gram** has occurred within a certain context, and then calculates the probability of it occurring in the given context.

The predictions of this model are further enhanced by the calculation and optimization of certain metrics like the **Entropy, Cross-Entropy** and **Perplexity.**



**Code Strategy:**

First, we iterate through the dictionary of all the words provided in the training dataset, and extract the **unigrams**, **bigrams** and **trigrams**. Next, for each **N-gram**, we calculate their frequency across all words. We repeat the same process for N-letter prefixes and N-letter suffixes.

The frequency tables of the above prefixes, suffixes and N-grams are given specific weights to quantify their degree of suggestiveness (i.e., matches of greater length are more indicative). We place a restriction to ensure that there are not vague matches.

The aforementioned N-Grams approach is further enhanced by the concept of **Information Gain**, which enables us to reach a certain information threshold before implementing the **N-Grams** approach while simultaneously minimizing the number of lives lost in the process.

We iterate through the set of words in our dictionary to calculate each word’s **Information Gain**, and guess the one with the highest **Information Gain**.

In the beginning, we simply guess a vowel, as vowels have a high chance of being correct because they are distributed frequently in the training dataset.

**Error Analysis:**

The strategy implemented above is on the right trach and can be optimized for further accuracy. The above strategy had its **Entropy** and **Cross-Entropy** within an acceptable threshold and could guess the words with approximately **51.2%** accuracy.