Definition of transformers

GLyC

October 14, 2025

During this work we wont use the definition of transformers from Li, et all. We will use similar one but equivalent. The structure will be the same but we will make more explicits the conditions for accepting and rejecting. Also the condition for stopping the CoT will be more clear.

As we said, the structure will be the same but we will change the alphabet. We will require that it has two special characters that represents the acceptance and rejection of the words. Also this characters should not appear in the input word. We will say that a transformer accepts a word when it prints the special character for acceptance and that it rejects a word when it prints the one for rejection.

probably we can eliminate this precondition somehow

Note that with this definition the CoT stops when it prints any of the two stopping characters. The transformers of Li, et all can easily simulate the ones defined by us because we can force the transformer through the token embedding so that when it sees a stopping character it replicates it until it reaches n_{max} and then it prints a 0 or a 1.

this looks super doable but maybe it requires some thinking