Heuristic Search using Language Models and Reinforcement Learning

Authors: Carolina Carvalho (cr.carvalho@gmail.com),

Paulo Quaresma (pq@uevora.pt)

eKNOW 2025

University of Évora (Portugal)

Neural Machine Translation (NMT)

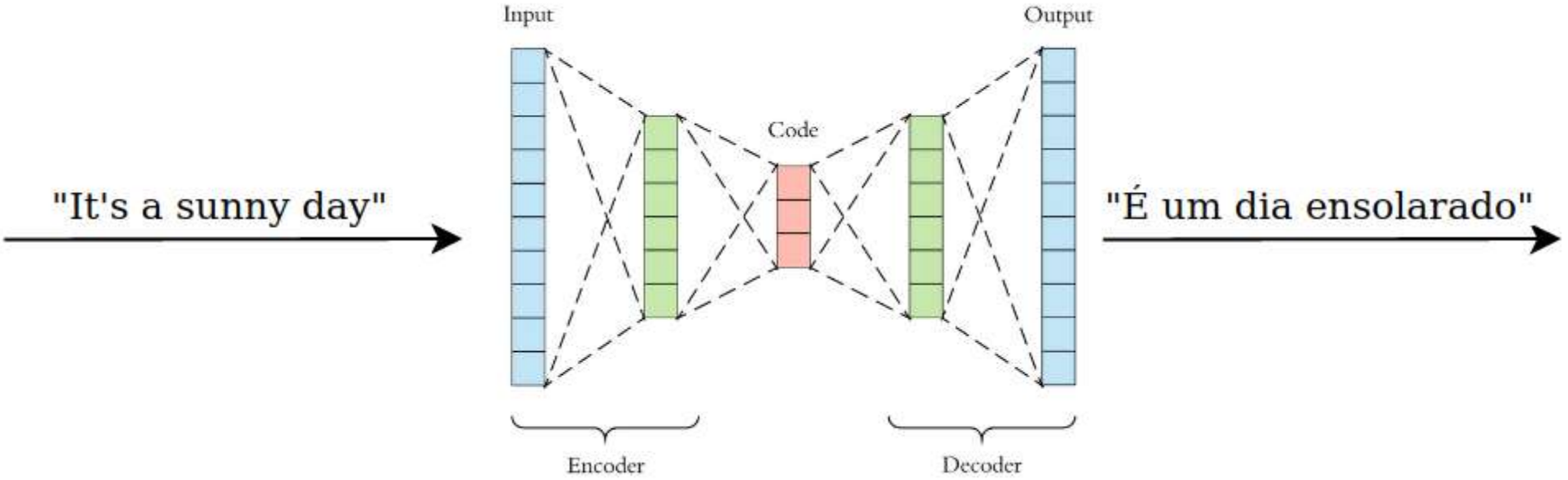


Image source: https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

Sequence to Problem (seq2prob)

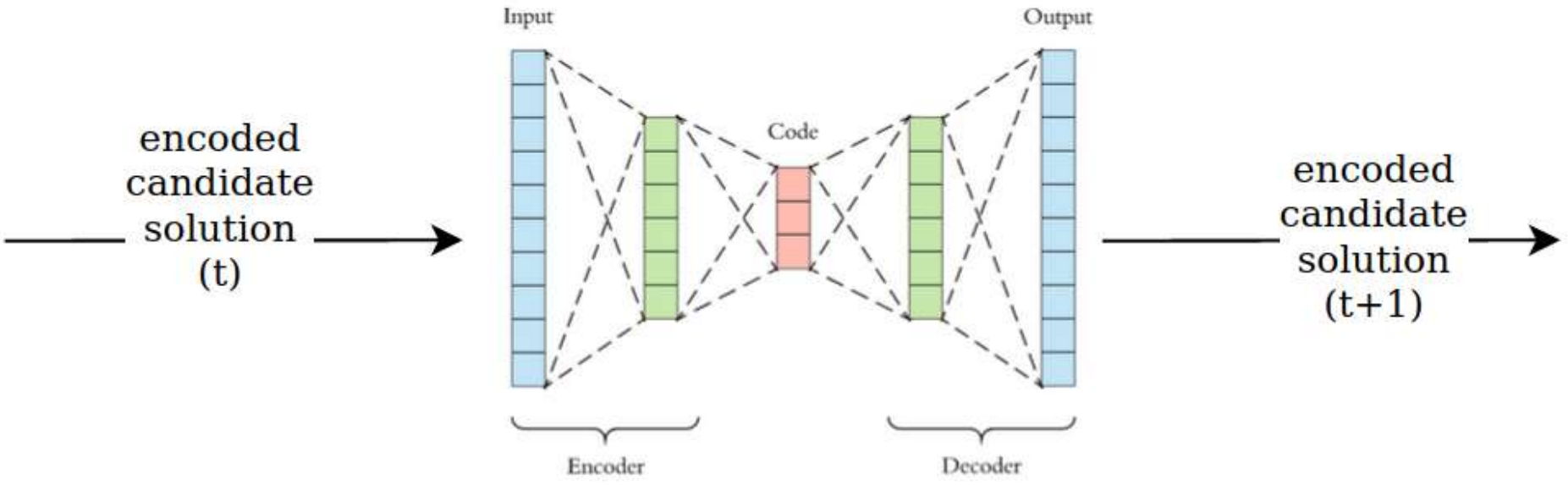
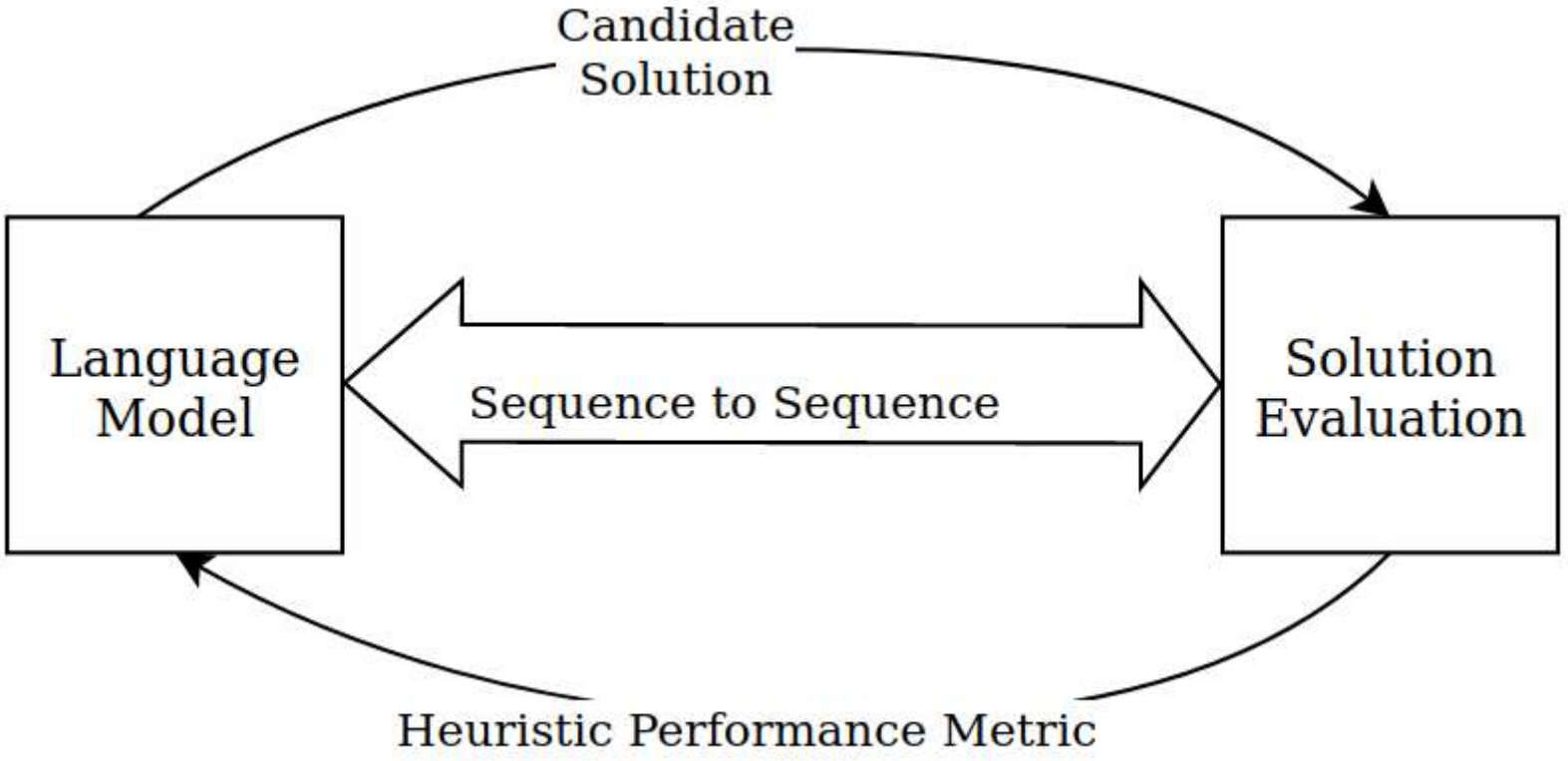


Image source: https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

Encoding a candidate-solution, sequence2problem

- Considering the NMT case, the entry set (X) and the output set(Y) are the original and the final translation languages.
- In the Seq2Prob, X=Y and they correspond to the problem's search space.
- Similary to the NMT setting, the candidate solutions are encoded through generated integer sequences.
- In order to help the encoding process and express conditions, special chars can be used (e.g. separators, padding char) in order to express the parameters used to build and evaluate the candidate-solution. Allowing a more complex onthology encoding and variable lenght solutions.

Proposed Architecture, Parallel with Evolutive Algorithms



NAS using RL

Considering the RL state as the Network Structure Code (NSC) and using the RL reward as a performance metric of the child-network, built using the NSC [1].

[1] Neural architecture search with reinforcement learning, (Zoph, Barret and Le, Quoc V.)

Difficulties using RL to train NMT models

In order to specify the desired behaviour it is necessary to stabilize the NMT model’s training using RL.

In a complex environment such as NAS, it is desirable to model it stochastically when regressing the odds.

Solving the stabilization problem

By breaking the problem of encoded candidate solution generation into sequence composition, is possible to decompose the learning needs into several pieces and design a dedicated architecture for each piece.

Solving the stabilization problem (Divide and Conquer)

With a position-value decoupling, the task of candidate solution generation (shared RL state) is simplified for each RL agent. This iterative refinement process is also continuous, matching the RL requirements of a Markov Decision Process (MDP).

Roadmap

Similary to the GAN, RL architectures use a self-critique methodology to improve the actor agent’s preformance.

Following the core concept of Auto-Encoders, an encoder-core is used to model the main problem features together with a costum decoder to access the downstream tasks.

The modular thinking in general informatics leads to a **functional** and **architectural decoupling strategy**.

Functional Decoupling (Divide and Conquer)

Sequence generation as a continuous refinement process of the composition of two models, one for sequence’s position generation of the new element and the other for the actual assigned value.

Choosed Language Models

- Char-Conv [1]
- Transformer-based VQ-VAE [2][3]

[1] Text Understanding from Scratch, (Zhang, Xiang and LeCun, Yann)

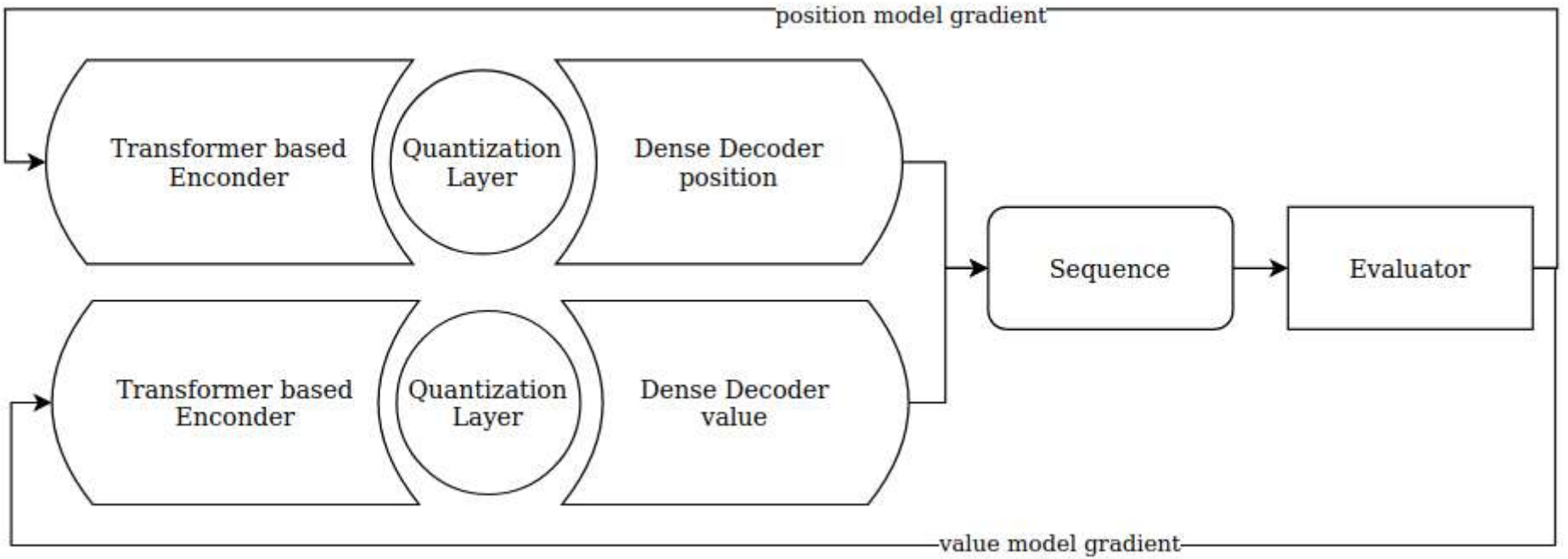
[2] Attention is all you need, (Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N. and Kaiser, Lukasz and Polosukhin, Illia)

[3] Neural Discrete Representation Learning, (Vinyals, Oriol and Kavukcuoglu, Koray)

Architectural Decoupling – A3C +VQ-VAE (Divide and Conquer)

- Regarding the A3C framing, the same core model (Transformer encoder and a Vector Quantized Layer) has two outputs: one for the actor odds, and the other for the critique’s performance.
- These output decoders are based on Dense layers.

VQ-VAE - Decoupling

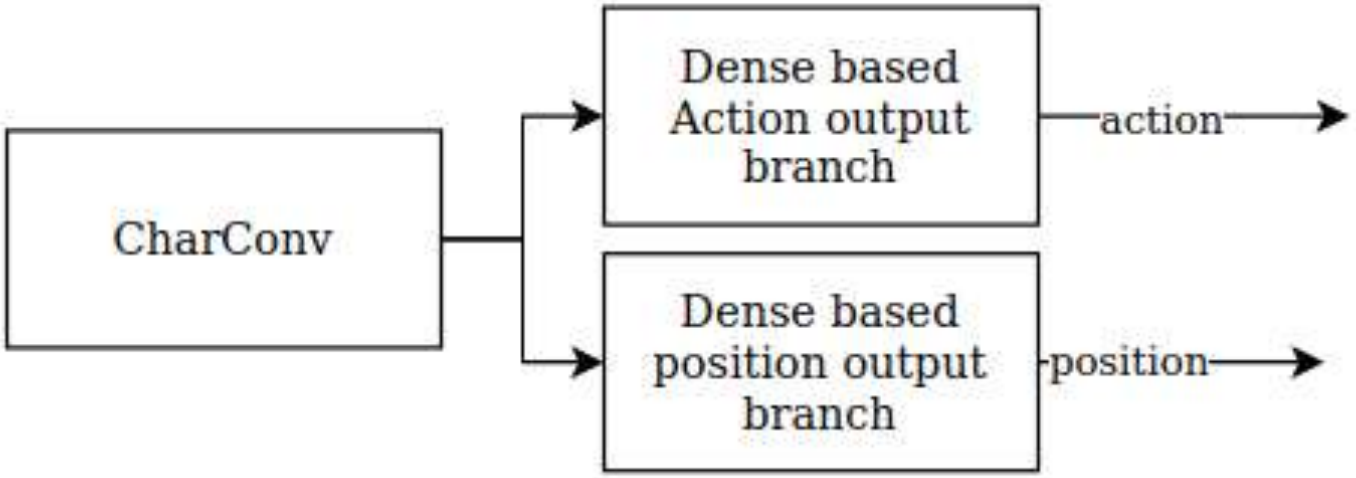


Architectural Decoupling – CharConv + SoftQLearning

- Regarding the CharConv, the same core model logic, with two outputs is used. The core is the CharConv network and two dense-based outputs are used for value+position generation.

- Another dedicated model is used to generate the corresponding Q-values for evaluation of the first model.

CharConv - Decoupling



Adopted RL Training Strategies

- Assynchronous Advantage Actor Critic (A3C)
- DeepQ Net-Policy Learning (DQN-PL) – does not supports odds modeling
- Soft-QLearning (SoftQL) – used entropy regularization to promote training convergence

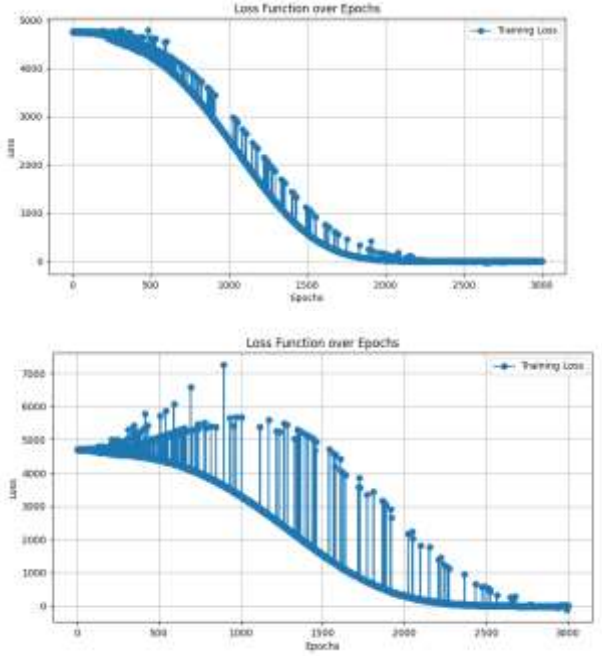
NAS environment

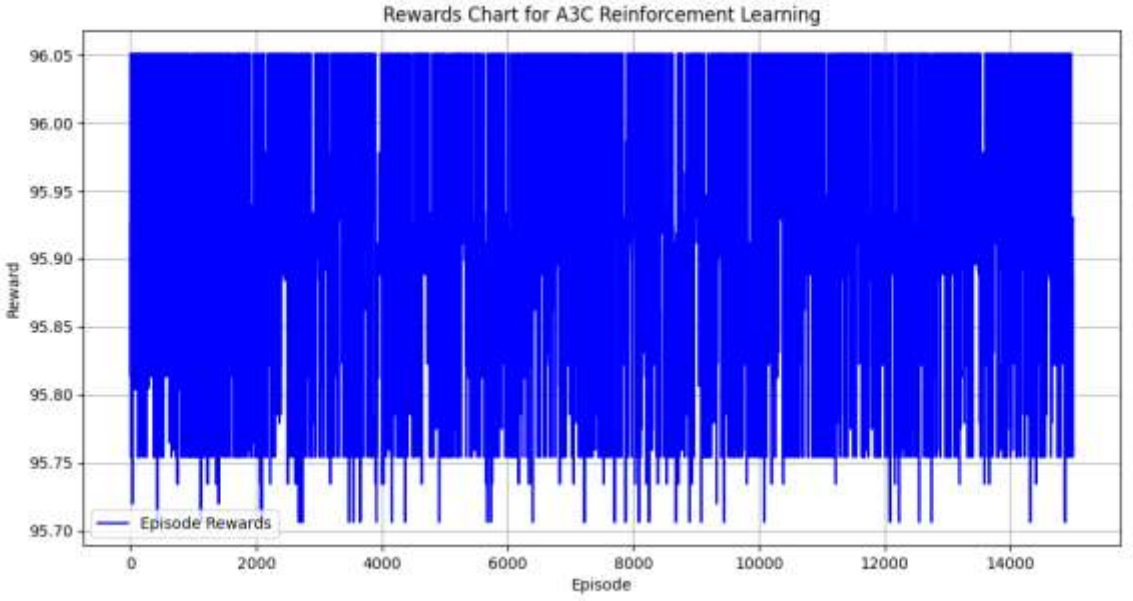
The used NSC was a depht-first-search inverted n-ary tree encoding an architecture of multi-branch classifiers.

Each decision kernel, encoded by an “1” in this three, is a Conv1D-based inceptionV1 block.

For the embeddings, a pre-trained roBERTa model was used.

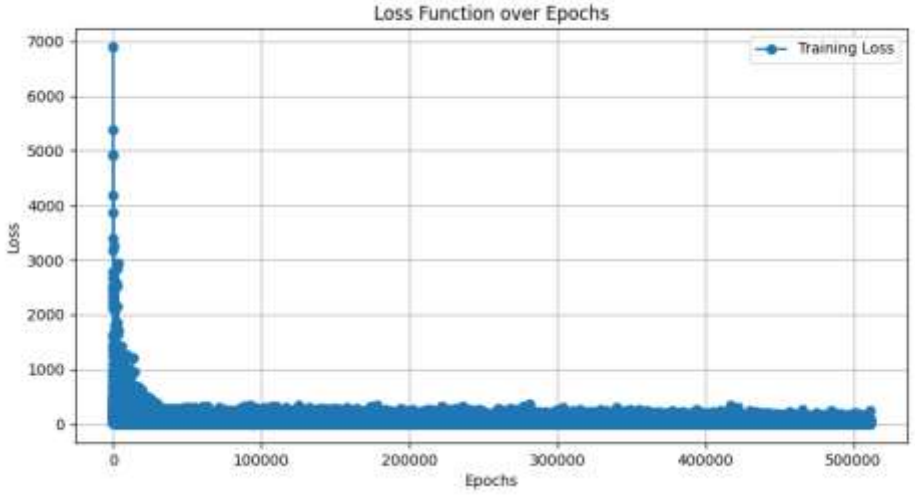
NAS environment, A3C

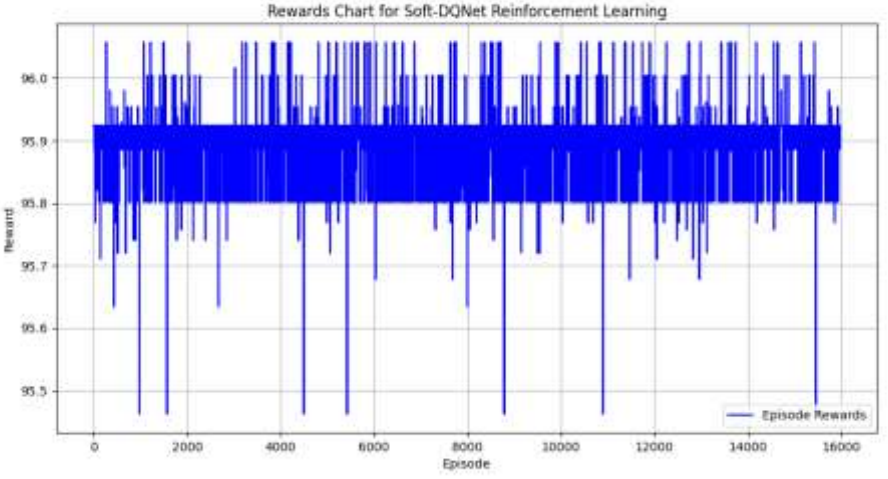




VQ-VAE and A3C training.

NAS environment, Soft-QLearning





Char-Conv + Soft-QLearning, in NAS e environment.

Conclusions

This work enables the continuous sequence generation training's stabilization by providing necessary framing to allow RL application in the context of Language Models.

(Final Slide)

Thank you! :)