Mimicking the writing style of Odysseus Elitys (Pt1. .with RNNs)

Introduction

The project “Mimicking the Writing Style of Odysseus Elytis” is the result of work completed for the course "Introduction to Data Science and Machine Learning II" at the Physics Department of the University of Crete. In this project, we explore three different approaches to mimicking a writing style, using the works of the Nobel Prize-winning Greek poet as our foundation. This project is intended purely for learning purposes and does not aim to achieve anything beyond that at this point.

Connecting Fields through Machine Learning

One of the best aspects of the revolutionary technology of machine learning is that it allows students to connect many fields of interest. I think academic classes that teach the subject have the potential to be very interesting in contrast to other more technical mathematical branches. As part of the "Introduction to Data Science and Machine Learning II" course at the Physics Department of the University of Crete, I developed this project on the creation of poetic speech as a final exercise. Whenever I start a new ML project, I aim to integrate knowledge from various fields, leveraging what I think is the best quality of machine learning. Therefore, I wanted a project that would not only get me in contact with natural language processes but also provide insights into non-technical domains.

Why Odysseus Elytis?

As a poetry enthusiast, I once read one of my poems to a dear uncle, who bluntly told me it was terrible and suggested that I study real poets to learn a thing or two. He recommended starting with the great Odysseus Elytis, one of Greece's two Nobel Prize winners in literature, so I thought I should base my model on his work. We all know how a student is always busy with reading and researching. Exploring the work of Elytis this way has been fascinating.

Choosing Recurrent Neural Networks (RNNs) for Language Generation

Recurrent Neural Networks (RNNs) are a specialized type of neural network architecture designed to handle sequential data. This specific architecture was chosen for the project due to its unique ability to retain and utilize information from previous time steps, allowing it to make informed predictions or decisions at later stages in the sequence. This capability makes RNNs particularly effective for tasks involving sequential relationships or dependencies, such as language generation.

In this project we are going to explore two different kinds of RNNs. As you will see the so called stateful models are better at grasping long term patterns in contrast with the simpler stateless models.

Data Collection: Gathering Elytis's Works

The first thing we have to do is gather the works of Elytis into a single text file that will serve as the basis for us to later train our model. There are many ways to do this, including browsing various relevant pages and extracting the information we need. While there are other possible solutions—some of which I have explored for future projects—this specific project did not required anything more than just browsing.

Alternative Data Extraction Techniques

As a quick overview, I have to mention some other possible ways to extract information from various resources using the Lang-Chain Documents Loader library. The Lang-Chain Documents Loader module is a tool designed to facilitate the extraction, transformation, and loading (ETL) of large volumes of text data from various resources. This module is particularly useful in projects requiring the aggregation of information from multiple sources for analysis, research, or reporting purposes. This very useful solution can help you directly download data from YouTube videos, web pages, PDFs, and more. My experimentation with the module can be found in the notebook “Database_Builder.ipynb”.

Text Cleaning and Normalization

First, we should clean the text by removing any whitespace and converting all letters to lowercase. This step is crucial for creating a consistent text file containing raw information, which helps eliminate the chances of predicting irrelevant characters. It's worth noting that Elytis occasionally uses Latin in his poems, which adds another layer to consider during cleaning.

Vectorization: Representing Text for Machine Processing

After preparing the dataset, the next step is to represent the words in a way that the machine can process. This is done through vectorization, which is the process of encoding words so that the relationships between them are preserved as distances in a mathematical space called a vector space. In this space, every point represents a word, and words that are contextually close to each other are also close in the vector space. While this may not be thrilling if you're not a fan of mathematics, it is an essential concept in machine learning. The vector space provides a comprehensible and visualizable way of training a model, making easier to understand and work with it.

Creating Input-Output Pairs

Another crucial step in dataset preparation is the creation of input-output pairs. The overall goal is to build a model capable of predicting the next character in a given sentence. To achieve this, we need to format our data in a way that aligns with the task. Specifically, we should structure our dataset so that it consists of unfinished sentences or phrases of a certain length paired with their corresponding completed versions.

This approach ensures that the model is trained on the typical inputs it will encounter during inference, along with the expected outputs. By repeatedly exposing the model to these input-output

pairs, it can learn the underlying patterns and structures of the text, eventually adjusting itself to mimic the writing style of the author. In the context of our project, this means the model will gradually learn to replicate the distinctive style and nuances of Odysseus Elytis's poetry. The ultimate goal is for the model not just to predict the next character accurately, but to generate text that is stylistically consistent with Elytis's unique literary voice. The careful structuring of data is a foundational step in ensuring that the model can generalize well to new, unseen inputs.

Splitting the Dataset: Training, Validation, and Testing

The final step in the data preparation process involves dividing the dataset into training, validation, and testing sets. This step is crucial as it ensures that the model is trained effectively, validated for hyperparameter tuning, and tested for performance evaluation. The proportions of these sets are adjustable and can vary depending on the size and nature of the dataset.

In this project, the complete poetic works of Elytis comprised approximately 500,000 lines of text. Given the substantial size of the dataset, it's essential to allocate the data appropriately to maximize the model's learning potential while also ensuring robust validation and testing.

For this specific case, I chose to allocate 80% of the data to the training set, which provides the model with the bulk of the information needed to learn patterns and structures in Elytis's poetry. The remaining 20% was evenly split between the validation and testing sets, with 10% of the data dedicated to each. The validation set is used to fine-tune the model's parameters, helping to prevent overfitting by offering a separate dataset to guide the optimization process. The testing set, on the other hand, serves as a final check on the model’s performance, providing an unbiased assessment of its ability to generalize to unseen data.

Understanding "Stateless" and "Stateful" RNNs

Let’s clarify those two terms, “stateless” and “stateful” RNNs that was introduced in the earlyer. Firstly, we should briefly describe the basic structure of a neural network. A neural network consists of an input layer, where the seed for the outcome is received, the hidden layers where the magic happens, and last is the output layer.

The network is using what we call “hidden states” to transfer information from the previous layers to the next layers. Hidden states are crucial because they represent the internal knowledge of the neural network. As data passes through the hidden layers, these states evolve, capturing patterns, correlations, and abstractions that are vital for tasks like image recognition, natural language processing, or time-series forecasting. The key difference between the stateless models in comparison with the stateful ones is in the management of those hidden states. For the stateless models, after processing a sequence, the hidden state at the final time step is discarded. When a new sequence starts, the RNN begins with a fresh hidden state (usually initialized to zeros). Whereas in the stateful models, the hidden state from the end of one sequence is carried over as the initial hidden state for the next sequence. This allows the RNN to maintain continuity across sequences, effectively "remembering" what it learned from the previous sequence.

Stateless vs. Stateful Models

The described process was applied to both the stateless and stateful models, with only a small but important difference. However, as mentioned earlier, the data preparation process for fine-tuning is

a bit more sophisticated, requiring additional steps and considerations, so it will be discussed separately.

Model Architecture: Stateless vs. Stateful Networks

Both the stateless and the stateful networks consisted of an input layer of 16 nodes, a single hidden layer of 128 nodes, and an output layer of 36 characters. Below, you can see some results produced by the stateless model:

When to Use Stateless and Stateful Models

Stateless models are suitable for tasks where the context is not dependent on previous inputs. In these models, each input sequence is treated independently, with no memory of prior inputs. While this approach can be effective for certain tasks, it may not be ideal for capturing the complexity of a nuanced writing style, such as that of Elytis.

In contrast, stateful models retain information from previous inputs in their hidden layers, allowing them to better understand and mimic longer writing trends within a text. This ability to preserve context over time makes stateful models more effective for tasks involving complex sequential data, where maintaining the flow and coherence of the content is crucial.

Hence, you can probably see the difference in writing. Let’s consider some examples of the outcomes of both models.

temp=0.01

stateless_model : “Κάπου εδώ πρέπει να είναι αυτό που ανοίγει το παράθυρο που ανεβαίνει το πρώτο που με που σε καιρούς που από την καρδιά της μια στιγμή στο μέλλον το καταλαφτές που από την καρδιά της μια στιγμή στο μέλλον το καταλαφτές που από την καρδιά της μια...”

translation: Somewhere here must be the one that opens the window that goes up the first that with that in times that from her heart a moment in the future you will understand it that from her heart a moment in the future you will understand it that from her heart a…

As we can see, Google Translate managed to handle most of the words generated by the model. For this inference, we used a low temperature value, which controls the level of creativity in the model's output. However, there are still some issues! It is easy to spot a loop that, in fact, continued for an entire paragraph beyond this excerpt. Choosing the most confident prediction too frequently often results in such loops. Let's experiment with different values for the temperature setting.

temp = 0.1

“Κάπου εδώ πρέπει να είναι αυτό που ανεβαίνει το πρώτο που από την παρά με το πρώτο που το παράθυρο που με το πέρασμα του παραθεί το πρώτο που με το πρώτο που με το πρώτο που το πρώτο που με το πρώτο που το παράθυρο που με το πρώτο που το πρώτο που με που σε καιρούς που ανάμεσα στην προσεχθεί και …“

translation: “Somewhere here it must be the one that goes up the first that from the than with the first that the window that with its passage is listed the first that with the first that with the first that the first that with the first that the window that with the first that the first that with that in times that between the attention and..”

Okay, I assure you, Elytis was a far better poet than his AI mimic. Nevertheless, in this second attempt, we can observe that the model eventually managed to escape its repetitive loops. There is an obsession with using very ordinary words, but these details can all be used to retrain this initial, simple model into something better. A common approach would be to apply a penalty for overly common words.

Moving forward, and leaving the stateless Elytis model behind (as no poet would want to be forgotten), let me present what I believe to be its most profound thought.

“Ισα τερματίζουμε όλοι στερνά
τα ματιά του κορίτσα της
που με το φυλάδι τη σταγό
το περιμένι το του ανοιχτά
το περιβέρι το φράχτα
Μεσημιά της ματιά
το περιβόλι της ανταλλινιάς
το απότημα την κλάμα
κι ότι σε μικρά στη σκοτά το αγόρα
τα μπράτσα που που 'ναι η καλό
τη ζωή μου τι και την αυτή
που φτάνει μες στο πλάι
στη φυλά στο μαύρο στα μάτια
στο μαντραφτάρι του προστά
- δεν αναβάλα που αντικρά
την αγγαλιά τη λίγη του μικρού
και το πανερό του πλάι
με το περιβάλι το περιβέρα
με το φαγγεριά του πάντα κι ερνός
όλα τα μακρινό μες στα δάκρυ
την άλλη στη μονάκα το κρεφό
και το περιβέρα του και το ανάμε
το περιβάρι του καθεμό
μια στιγμή του και το πράσουν
τι πάντα με την αντικρτή
Όμως το πλάι και μια φωτιά
την παράμνη στη ζωή μου τι μια
με το χρόνο του σκοτού
στα μονάχια και τι και το περιστέρι
και το πλάι στα μαλλιά
τη ζωή μου τις φυλάξει το περιάγι
το σπίτι με την αντικρτά
πάνε κι όλα τα φυγάνισσα
κι εμνός από την υποσταλίνη
και το μπλε μικρά στο μακρινό και στη ζωή
μια σταφό το σπαρμόν..”

“We all end up in the same position
the glances of her daughter
that with the race the dot
wait for it openly
around the fence
Her middle glance
the andalini orchard
the stock the cry
and that he bought it in the dark
the arms where the good
my life and hers
which reaches to the side
in the tribe in the black in the eyes
in his hideout
- I didn't put off what you said
the hug of the little one
and its bready side
with the surrounding the surrounding
with his panta ki ernos
all distant in tears
the other to the nun the crefo
and his girdle and aname
the entourage of each
a moment of it and they do it
what's up with the opposite side?
But the side and a fire
the missing person in my life what a
with the time of death
in the monasteries and what about the dove
and the side of the hair
my life is guarded by the periagi
the house opposite
everything goes as a fugitive
and men from under Stalin
and the blue small in the far and in the life
a bunch of sparmon..”

It makes you wonder, doesn't it?

If you enjoyed the machine-generated poetry from the stateless model, you will be amazed by its stateful counterpart. Although the training process for this model involved a different preparation, it

still used a moving window approach. In future experiments, we will explore different vectorization methods that could be more efficient for training models to create poetry. Here is the initial result from the stateful model.

“0.1: 'καλώς εχόντων των αντρών με το καλοκαίρι στο στήθος το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού τη στα ματιά της αποστήθες και το παιδί με το παντού το παιδί με το παντού το παιδί με το κατά το παντού το παιδί με το καλοκαίρι μες στο στήθος το παιδί με το παντού το παιδί με το καλοκαίρι μες στο μαντικά του παντού με το παντού το περιμένο μες στο στήθος το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού τα περιστέρια με το παντού το παιδί με το καλοκαίρι μες στο στήθος το παιδί με το παντού τα παιδιά του παντού μες στο στήθος το παιδί με το καλοκαίρι μες στο στήθος το παιδί με το παντού το παιδί με το παντού το παιδί με το κατά το παιδί μες στα σκοτεινά τη στα ματιά του παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το παντού το παιδί με το περιμόνι το παιδί με το περιμόνε και “

0.1: welcome to men with summer on their chests the child with everywhere the child with everywhere the child with everywhere the child with everywhere the child with everywhere the child with everywhere the child with everywhere the child with the everywhere the child with the everywhere the in her eyes the child with the everywhere the child with the everywhere the child with the everywhere the child with the summer in the chest the child with the everywhere the child with the summer in the oracles of everywhere with everywhere the waiting in the chest the child with everywhere the child with everywhere the child with everywhere the child with everywhere the pigeons with everywhere the child with summer in the chest the child with everywhere the his children everywhere in the chest the child with the summer in the chest the child with the everywhere the child with the everywhere the child with the child in the dark the in the eyes of the everywhere the child with the everywhere the child with the everywhere the child with everywhere the child with everywhere the child with the perimone the child with the perimone and “

There are broader patterns emerging! This means the model is starting to capture larger trends in Elytis's poetry. However, the results are still quite chaotic. Let's see if they improve with further training.

Temp = 0.4
“καλώς εχόντων των κάμπου τα περνά και τ αποματιά της το παράθυρο και το αντροβαλιά
τα μαχαίρι στα μάτια μα τα παντέρια και το περιμένο μες στο σπίτι στο στήθος μου
το μαντάρι το που τα περιστέρι τ ανεμογάρι του παιδού που το παιδί κοιτάει που τα
ούρον το κοιμού το παντερού το κεφάλι με το παναγιά του δείνου στο σκοτεινό και
το ρωτικό το μικρό και βράχουν το μικρά το μαύρο και το παιδί μετά και το περιμένο
το φεγγάρι του αντροπικρά μες στα σεμάτια που το παιδί με τα περιστέρι μα τα κεράκια
παντολιά η μαρίνα και τα παντέρια με τα σεντόνια στη σύννεφα το παράπνωσε στην αποκουγούλια
της και τρικυλί του μιλάνι και τα μαλλιά της αποσπασμένα τα παιδιά τη μαύρη του
κουρού τη μησερό μικρού και τα μαλλιά του να μας κι εγώ στο φως στα δύο του κάθε
το στήθος το νερό μου το μπουστά και το κορίτσι μες στα σκοτεινά τη μικρό μου και
τρι την παρασκευτη παραμονιές τι τι πικραμένο με το περιμόνι του αν το παναγί που
μας η φωτιά τη στιγμή και το εσύ της να τρεις της τι στιγμή που την παράξενο από
μια στα σκοτεινά τα πα”

“well-being of the plains, she passes them and glances at the window and the antrovalia
the knife in my eyes and the panthers and the waiting in the house in my chest
the corral, the doves, the child's windmill, which the child is looking at
urine the sleep the panther's head with the virgin of the deino in the dark and
the rosy the small and rock the small the black and the child then the expected
the moon mischievously in the fields that the child with the doves and the candles
the marina pants and the panthers with the sheets in the clouds breathed it in the apokougoulia
her and his trikyli milani and her hair detached the children his black
cut the middle of the little one and his hair so that we and I in the light in two of each
my breast my water is bubbling and the girl in the dark my little one and
three days before the day, what is bitter with his perimony if the panagi where
us the fire the moment and the you of to three of what moment that the strange from
once in the dark”

At this point, I am convinced that Elytis actually used many of the words generated by our model. This is a promising start and forms a solid foundation for the next part of this poetry method examination: fine-tuning! We started from scratch, making an initial attempt to generate something interesting, but the truly useful and exciting results may come from further training a model that already understands the language of the poet—a model that already knows Greek.

The fine-tuning process was too extensive to include in this brief introduction and should be discussed separately in the next section. This work, along with our exploration of RNNs, could serve as the starting point for creating beautiful things. As technology evolves and new resources become available, the training process will likely shift to the cloud. Fine-tuning a model like GPT-2, for example, can produce really interesting results. New concepts come into play, as well as new subtypes of RNNs.