Poem Generation - Milestone I

Jakub Bilski, Jakub Brojacz, Jakub Kosterna

March-April 2022

Contents

1 Literature Analysis

1. Oliveira, Hugo. "Automatic generation of poetry: an overview." Universidade de Coimbra (2009)

The oldest article analyzed by us includes generators based on existing poems with the classic approach - no deep learning methods used. The methodologies used there contain among others creating templates and then filling them with matching words and making rules on how to match words to generate matching lines. In addition, the paper included such approaches as creating structures of matching words, word accents, words sorted by subjects, grammatical templates and more. Also the evolutionary algorithm is mentioned.

2. Van de Cruys, Tim. "Automatic poetry generation from prosaic text." Proceedings of the 58th annual meeting of the association for computational linguistics (2020)

The publication presents encoder-decoder architecture, generating poems based on Prosaic Text. The model is made up of gated recurrent units (GRUs) - almost the same as in our Jupyter Notebook. The system is exclusively trained on standard, non-poetic text, and its output is constrained in order to confer a poetic character to the generated verse. Even though it only uses standard, non-poetic text as input, the system yields state of the art results for poetry generation.

3. Tanel Kiis, Markus Kängsepp. "Generating Poetry using Neural Networks." (2018)

The paper describes LSTM and Variational Autoencoders methods. LSTM (both word-by-word and sign-by-sign) can figure out grammar, but poems are without meaning; on the other hand, VAE focuses on repeating phrases commonly occured in training data, but it is able to generate at least some poems that could interpreted as a original and coherent poetry.

4. Tikhonov, Aleksey, and Ivan P. Yamshchikov. "Sounds Wilde. Phonetically extended embeddings for author-stylized poetry generation." Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology (2018)

Using a version of the language model with extended phonetic and semantic embeddings to generate poetry, the paper showed that phonetics makes a comparable contribution to overall model performance as author information. Phonetic information was shown to be important for English, and people tend to attribute machine-generated texts to the target author.

5. Praneeth Bedapudi. "DeepCorrection2: Automatic punctuation restoration" (2018)

The blog post brings closer an implementation for automatic punctuation of text without commas or dots. A seq2seq model was created in keras, along with an LSTM encoder-decoder.

2 Exploratory Data Analysis

Data is taken from Poems kaggle dataset. The poems there are categorized by the form (e.g. haiku, sonnet, etc.) or topic (love, nature, joy, peace, etc.).

In order to improve our model's results, poems from the Project Gutenberg may also be included. A free eBook library consists of about texts, which can be useful if aforementioned kaggle dataset will prove to be insufficient.

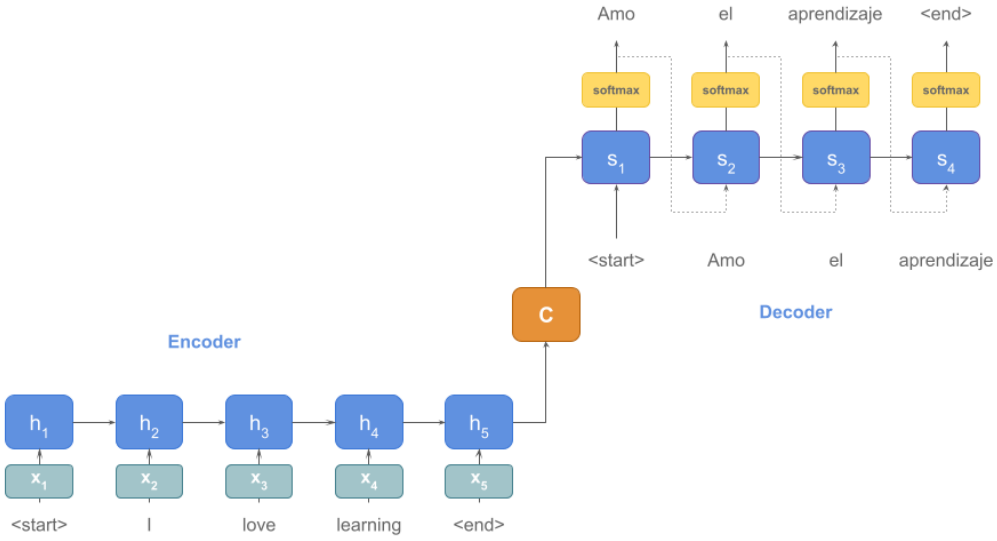

Figure 1: Encoder-decoder schema - in term of translating

Data is divided by type of poem and by subject, so we can check how our model will behave when only selected types of poems are present in training data.

The dataset's site contains also two potentially helpful notebooks by other kaggle users - solutions *Poem Generation using FastAI* and *Starter: Poems Dataset (NLP)* have been briefly reviewed by us so far, but will probably be useful for writing further code.

3 Solution Concept

Our application will be naturally developed in Python - this decision was made because of the largest number of tools adapted to the problems of natural language processing in this language, and moreover its capabilities meet the needs of writing algorithms based on the cited literature.

We will use **encoder-decoder architecture**, with concept of generating text line-by-line. From set of generated lines we will take the best fit based on rhymes, accents and overall cohesion with the rest of the text. We intent to choose the best objective function that will optimize the creation of poems resembling those created by a living person.

Given the availability and also the thorough testing, we are likely to use a solution involving the pretrained BERT model. While we considered building the model from scratch, it is important to keep in mind that our dataset is too small to teach such a big system. Another problem is a big number of words appearing only once which would result in no option for a neural net to learn what such words mean. However, the solution will eventually be adapted to load a larger dataset, and we will try to ensure that the time involved in training potentially larger data will have optimal time complexity [or at least not the most naive ;)].