Modern French Poetry Generation with RoBERTa and GPT-2

Hämäläinen, Mika

Hämäläinen , M , Alnajjar , K & Poibeau , T 2022 , Modern French Poetry Generation with RoBERTa and GPT-2 . in M M Hedblom , A A Kantosalo , R Confalonieri , O Kutz & T Veale (eds) , Proceedings of the 13th International Conference on Computational Creativity . Association for Computational Creativity , Bolzano , pp. 12-16 , International Conference on Computational Creativity , Bozen-Bolzano , Italy , 27/06/2022 .

http://hdl.handle.net/10138/351506

Modern French Poetry Generation with RoBERTa and GPT-2

Mika Hämäläinen[1,2], Khalid Alnajjar[1,2] and Thierry Poibeau[2]

[1]University of Helsinki, Finland

[2]École Normale Supérieure-PSL and CNRS and Université Sorbonne nouvelle, Paris, France

firstname.lastname@{helsinki.fi[1] or ens.psl.eu[2]}

Abstract

We present a novel neural model for modern poetry generation in French. The model consists of two pretrained neural models that are fine-tuned for the poem generation task. The encoder of the model is a RoBERTa based one while the decoder is based on GPT-2. This way the model can benefit from the superior natural language understanding performance of RoBERTa and the good natural language generation performance of GPT-2. Our evaluation shows that the model can create French poetry successfully. On a 5 point scale, the lowest score of 3.57 was given by human judges to *typicality* and *emotionality* of the output poetry while the best score of 3.79 was given to *understandability*.

Introduction

Poem generation is a challenging creative natural language generation task. As a form of art, it has undergone several changes in the history. Classical poetry incorporates typically meter and rhyme as their function was to help people recall poems, especially when poetic tradition was still mostly oral rather than written.

In the modern era, the role of the poetry has changed from an art form that has to follow a fixed structure that defines its meter and rhyming such as iamb, haiku or anapest. Modern poetry is more concerned about creating something new by breaking any strict structural rules and by continuously questioning what poetry is, what it can be and what it should be (see Kantokorpi, Lyytikäinen, and Viikari 1990).

In the field of poem generation, meter is a feature that is very often considered in generated poetry (Colton, Goodwin, and Veale 2012; Lau et al. 2018; Hämäläinen and Alnajjar 2019b; Zugarini, Melacci, and Maggini 2019; Lewis, Zugarini, and Alonso 2021). By incorporating meter, people can be more forgiving when evaluating the output of the system as it is known that people are ready to interpret more into the content of the output of a computationally creative system if the form is correct (Veale 2016). In other words, a poem that looks like a poem, as in that it follows a certain meter, must be a poem. A truly competent computational poet should be capable of generating something that is recognizable as a poem even if its output was modern free-form poetry.

In this paper, we explore the topic of modern poetry generation in French. We fine-tune a novel encoder-decoder architecture which consists of a RoBERTa (Liu et al. 2019) based model as the encoder and a GPT-2 (Radford et al. 2019) based model as the decoder. Because RoBERTa is very good at natural language understanding tasks but poor at generation tasks and GPT-2 is good at generation, but bad at understanding, it makes sense to incorporate both of the models. The task of RoBERTa is to encode the input (i.e. to understand poetry) and the task of GPT-2 is to decode the output (i.e. to generate poetry).

Related work

Poem generation has sparked a lot of interest in the past as we can see in a recent survey on the field (Gonçalo Oliveira 2017). There is also some work on generating French poetry in particular (Van de Cruys 2019; Poibeau et al. 2020). In this section, we will provide a quick overview on some of the related work.

Poetry has been generated by using rules. This can, for instance, be done by modeling certain poetic features to find suitable verses (Lamb and Brown 2019), by writing different rules to generate verses with different poetic features (Hämäläinen 2018) or by using templates (Colton, Goodwin, and Veale 2012).

Genetic algorithms have been popular in poem generation as well (Manurung, Ritchie, and Thompson 2012; Yang et al. 2016; 2021). In many of these approaches, the generation itself is not governed by rules, but rules are used to assess the quality of each generated poem. These approaches can be seen as a step away from fully relying on rules in the generative process.

Recently, different neural approaches have become commonplace in poetry generation such as LSTM models (Zhang and Zhang 2020; Atassi and El Azami 2022) and GPT-2 (Hu and Sun 2020; Beheitt and Hmida 2022). Our approach follows this line of work. How we set ourselves apart from the existing models is that we propose a novel poem generator model that can benefit from both RoBERTa and GPT-2 instead of relying only on one model. In addition, we condition the beam search on rhyme instead of solely relying on the output of a vanilla model.

Verse	Input	Output	Output in English
1	amant galamment escrime dame	Vainement, paladin des dames, tu t'escrimes	In vain, paladin of the ladies, you fight me with a sword
2	Vainement, paladin des dames, tu t'escrimes	À crier que j'ai fait, ici, peu galamment.	Shouting what I have done, here, little gallantly.
3	À crier que j'ai fait, ici, peu galamment.	si de Laure Pétrarque avait été l'amant,	if Pétrarque had been the lover of Laure
4	si de Laure Pétrarque avait été l'amant,	Qui sait ce qu'il eût dit de Laure dans ses Rimes?	Who knows what he said about Laure in his Rhymes?

Table 1: Example of the training data for one poem

Data

As machine learning requires data, we need a poem corpus. For this reason, we crawl all the French poems that are available on Wikisource[1]. The poems are not free of noise as some of the poems include verses in Greek alphabets, multiple different types of quotation marks, hyphens and spaces of different lengths etc. We clean the data from all of these inconsistencies by manually inspecting odd characters and either by replacing them (e.g. only one type of a hyphen) or removing them (e.g. Greek letters). The corpus contains 7553 poems. In addition, we use the French sonnet corpus introduced by Poibeau et al. 2020. This corpus has 1039 sonnets.

Because these poems and sonnets are of different lengths, we split all of them into stanzas. From this point on, we treat a stanza as a poem so that all poems in our corpus are of a similar length. This gives us altogether 25,215 French poems and sonnets. For the purposes of our models, we do not make a distinction between poems and sonnets.

Poem generator

In this section, we describe our poem generation model. The model follows an encoder-decoder architecture where the encoder is a RoBERTa model and the decoder is a GPT-2 model. Rather than training these models from scratch, we use pretrained language models and fine-tune them for the task of poem generation using a transfer learning approach. We chose a RoBERTa-based model as the encoder given their great ability in capturing contextual semantics. GPT-2 is well-known for modeling a language; hence, making an optimal decoder for text-generation tasks.

First we have to pick the suitable pretrained models. As we use Transformers library (Wolf et al. 2020), we select our models from their repository. The current state-of-the-art French RoBERTa model is CamemBERT[2] (Martin et al. 2020) which is based on the RoBERTa (Liu et al. 2019) architecture and trained on the large OSCAR corpus (Abadji et al. 2022) in French. We use CamemBERT as our encoder.

As for the selection of the GPT-2 model, there were several alternatives. By trying the models out, we could see that all of them except for Belgian GPT-2[3] (Louis 2020) predicted rather poor output. The model was trained on a variety of genres (such as news, Wikipedia, novels, European parliament text etc.) on a relatively big, around 60 GB, corpus. For this reason, we opted for Belgian GPT-2 as our decoder model.

We use Spacy[4] (Honnibal et al. 2020) to extract up to 4 keywords from each poem in the corpus. We train our encoder-decoder architecture for sequence to sequence generation, where it predicts the next verse in a poem given a previous verse. In the absence of a previous verse, we train the model to predict the first verse of a poem from the up to 4 keywords extracted from the poem. An example of input and output in the training data for one poem can be seen in Table 1. The first input consists of the keywords *amant* (lover), *galamment* (gallantly), *escrime* (fencing) and *dame* (lady), which are used to predict the first verse of the poem.

The poem corpus is split randomly to 80% for training and 20% for validation. The model is trained for 10 epochs. We use the Adam algorithm (Kingma and Ba 2014) with decoupled weight decay regularization (Loshchilov and Hutter 2017) and learning rate of 5e-05 to optimize the parameters of the model, with cross entropy loss as the loss function to reduce the difference between gold standard token and predicted tokens.

Rhyming is taken into account during the generation phase. The model is requested to generate a sequence between 4 to 20 tokens, with a length penalty of 1.0 using a greedy approach. At each step of generating the output sequence (i.e., when predicting the next token), we use the model to predict the top 10 possible tokens instead of just one highest scoring output. We then sort these candidate tokens based on their probabilities and rhyming scores. The rhyming score is calculated by counting the number of tokens in the output that rhyme (full rhyme, consonance or assonance) with the input (i.e., the previous verse and any subsequent words generated during the run).

Because it is not easy to know whether two French words rhyme or not based on the orthography (similarly to English), we use eSpeak-ng[5] to produce an IPA (international phonetic alphabet) representation for each token in the model's vocabulary. IPA alphabets are designed to represent how words are pronounced by writing out the actual phonemes. We use a simple set of rules to compare the IPA strings of two tokens with each other to determine whether they rhyme or not.

In practice, we first ensure that both of the IPA strings are equally long, if this is not the case, we remove characters from the beginning of the longer string until the IPA strings are equally long. If the strings are identical, no rhyme is considered, because a word does not make a good rhyme with itself. For full rhyme, the two IPA strings rhyme if they are identical from the first vowel onward. For assonance, we replace all consonants with a placeholder character *C*, if the

---

[1]https://fr.wikisource.org/wiki/Catégorie:Poèmes

[2]https://huggingface.co/camembert-base

[3]https://huggingface.co/antoiloui/belgpt2

[4]The *fr_core_news_sm* model

[5]https://github.com/espeak-ng/espeak-ng

D'un beau travail, d'une bonne pose,	From a beautiful work, from a good pose.
De la paix, de la beauté.	From the peace, from the beauty
Que je plains la beauté	Oh, I lament the beauty
De la femme, qui m'inspire	Of the woman, who inspires me
C'est ici que s'éveille le soleil,	It is here where the sun wakes
C'est ici que repose le grand créateur,	It is here where the great creator rests
Dont la ruine, hélas! se renouvelle	Whose ruin, alas! renews itself
De l'Enfant du Progrès	From the Child of Progress
C'est un des mois les plus beaux de l'année,	It is one of the most beautiful months of the year,
C'est le printemps, c'est l'été, c'est	It is the spring, it is the summer, it is
Le ciel où mon printemps se joue.	The sky where my spring plays.
À mon jardin qui s'effondrit.	In my garden that collapses

Table 2: Examples of generated poetry and their translations.

IPA strings are identical, i.e. they share the same vowels in the same positions, they are considered to have assonance rhyme. For consonance, we do the same as with assonance, but by replacing all vowels with a placeholder *V*.

Results and evaluation

For evaluation purposes, we generate 20 different poems consisting of 4 verses each. For each poem, we use a set of four randomly selected keywords among all the keywords extracted from the poem corpus. None of the keyword combinations is identical to what the model saw during the training. We generate the poems similarly to the example shown in Table 1. This means, that the keywords were used to generate the first verse, which was then used to generate the second verse and so on.

Some of the generated poems and their translations can be seen in Table 2. As we can see, the generated output is cohesive and quite grammatical. We can, however, see that sometimes the verb conjugation might be wrong such as in the case of *effondrit* which is a non-existing inflectional form of *effondrer* (to collapse). Also, the model has a tendency of starting every verse with a capital letter even if it was a continuation to the sentence started in the previous verse.

We conduct a crowd-sourced evaluation on Appen[6]. We set French as a language requirement for the crowd-workers so that we know that they actually speak French and are able to assess French poetry. Each poem is evaluated by 20 different crowd-workers. An individual worker can evaluate all 20 different poems or just some of them, in which case the remaining unevaluated poems are shown to a different crowd-worker. An individual crowd-worker cannot evaluate the same poem multiple times.

For evaluation, we use the same parameters as used by several authors for evaluating poetry (Toivanen et al. 2012; Hämäläinen and Alnajjar 2019a; Shihadeh and Ackerman 2020): (1) *The poem is typical* (2) *The poem is understandable* (3) *The poem is grammatical* (4) *The poem evokes imagination* (5) *The poem evokes emotions* (6) *I like the poem*. These statements are evaluated in a 5 point Likert scale, where 1 represents the worst and 5 the best grade.

	Q1	Q2	Q3	Q4	Q5	Q6
Avg	3.57	3.79	3.77	3.65	3.57	3.77
STD	0.88	0.84	0.81	0.79	0.88	0.77

Table 3: The evaluation results and standard deviation

The results can be seen in Table 3. All in all, the results are good and show that the system can generate poetry successfully. The lowest scores were obtained for *typicality* and *emotionality*. The highest score was given to *understandability*. In the future, more robust human evaluation methods need to be applied to understand why these parameters scored high and low (Hämäläinen and Alnajjar 2021a; 2021b).

Conclusions

In this paper, we have presented a novel approach to French poem generation. We have presented an architecture that consists of RoBERTa and GPT-2 models that are fine-tuned on a poem corpus. In addition, we have modeled rhyme as a part of the prediction pipeline of the model.

The results obtained in human evaluation are promising and they indicate that the model performs well in the task it was designed to do. In order to make the evaluation results more transparent, we have released them in full on Zenodo[7] together with the generated poems that were used in the evaluation.

Pretrained neural language models have been proven to be useful in poem generation. In the future, it would be interesting to study them in a multilingual setting, where a pretrained multilingual model is fine-tuned to generate poetry using the corpora of some languages other than the desired target language.

Author Contributions

The first two authors contributed to the work presented in this paper equally. The third author was involved in planning the methods and writing the paper.

Acknowledgments

References

Abadji, J.; Ortiz Suarez, P.; Romary, L.; and Sagot, B. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints* arXiv:2201.06642.

Atassi, A., and El Azami, I. 2022. Comparison and generation of a poem in arabic language using the lstm, bilstm and gru. *Journal of Management Information & Decision Sciences* 25.

Beheitt, M. E. G., and Hmida, M. B. H. 2022. Automatic arabic poem generation with gpt-2. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,*, 366–374. INSTICC.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *ICCC*, 95–102.

Gonçalo Oliveira, H. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, 11–20. Santiago de Compostela, Spain: Association for Computational Linguistics.

Hämäläinen, M., and Alnajjar, K. 2019a. Generating modern poetry automatically in Finnish. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5999–6004. Hong Kong, China: Association for Computational Linguistics.

Hämäläinen, M., and Alnajjar, K. 2019b. Let's face it. finnish poetry generation with aesthetics and framing. In *Proceedings of the 12th International Conference on Natural Language Generation*, 290–300.

Hämäläinen, M., and Alnajjar, K. 2021a. The great misalignment problem in human evaluation of NLP methods. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, 69–74. Online: Association for Computational Linguistics.

Hämäläinen, M., and Alnajjar, K. 2021b. Human evaluation of creative NLG systems: An interdisciplinary survey on recent papers. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, 84–95. Online: Association for Computational Linguistics.

Hämäläinen, M. 2018. Harnessing nlg to create finnish poetry automatically. In *Proceedings of the ninth international conference on computational creativity*. Association for Computational Creativity (ACC).

Honnibal, M.; Montani, I.; Van Landeghem, S.; and Boyd, A. 2020. spacy: Industrial-strength natural language processing in python, 2020. *URL https://doi.org/10.5281/zenodo* 1212303(6).

Hu, J., and Sun, M. 2020. Generating major types of chinese classical poetry in a uniformed framework. *arXiv preprint arXiv:2003.11528*.

Kantokorpi, M.; Lyytikäinen, P.; and Viikari, A. 1990. *Runousopin perusteet*. Gaudeamus.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. In *arXiv*.

Lamb, C., and Brown, D. G. 2019. TwitSong 3.0: towards semantic revisions in computational poetry. In *Proceedings of the Tenth International Conference on Computational Creativity*, 212–219.

Lau, J. H.; Cohn, T.; Baldwin, T.; Brooke, J.; and Hammond, A. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1948–1958.

Lewis, D.; Zugarini, A.; and Alonso, E. 2021. Syllable neural language models for english poem generation. In *12th International Conference on Computational Creativity (ICCC'21)*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I., and Hutter, F. 2017. Decoupled weight decay regularization. In *arXiv*.

Louis, A. 2020. BelGPT-2: a GPT-2 model pre-trained on French corpora. https://github.com/antoiloui/belgpt2.

Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence* 24(1):43–64.

Martin, L.; Muller, B.; Ortiz Suárez, P. J.; Dupont, Y.; Romary, L.; de la Clergerie, É.; Seddah, D.; and Sagot, B. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7203–7219. Online: Association for Computational Linguistics.

Poibeau, T.; Maignant, M.; Mélanie-Becquet, F.; Plancq, C.; Raffard, M.; and Roussel, M. 2020. Sonnet combinatorics with OuPoCo. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, 133–137. Online: International Committee on Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Shihadeh, J., and Ackerman, M. 2020. Emily: An emily dickinson machine. In *ICCC*, 243–246.

Toivanen, J.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-Based Generation of Content and Form in Poetry. In *Proceedings of the Third International Conference on Computational Creativity*.

Van de Cruys, T. 2019. La génération automatique de poésie en français (automatic poetry generation in French). In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume I : Articles longs*, 113–126. Toulouse, France: ATALA.

Veale, T. 2016. The shape of tweets to come: Automating language play in social networks. *Multiple Perspectives on Language Play* 1:73–92.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Yang, W.; Cheng, Y.; He, J.; Hu, W.; and Lin, X. 2016. Research on community competition and adaptive genetic algorithm for automatic generation of tang poetry. *Mathematical Problems in Engineering* 2016.

Yang, W.; Weng, W.; Chen, G.; and Jiang, Z. 2021. Elitist strategy of genetic algorithms for writing tang poetry. *International Arab Journal Of Information Technology* 18(4):604–610.

Zhang, H., and Zhang, Z. 2020. Automatic generation method of ancient poetry based on lstm. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 95–99. IEEE.

Zugarini, A.; Melacci, S.; and Maggini, M. 2019. Neural poetry: Learning to generate poems using syllables. In *International Conference on Artificial Neural Networks*, 313–325. Springer.