Sonnet Combinatorics with OuPoCo

Thierry Poibeau, Mylène Maignant, Frédérique Mélanie-Becquet, Clément Plancq
LATTICE (CNRS & ENS/PSL & Univ. Sorbonne nouvelle)
1, rue Maurice Arnoux, 92120 Montrouge, France
`{firstname.lastname}@ens.psl.eu`

Matthieu Raffard, Mathilde Roussel
Atelier Raffard-Roussel
189 rue Ordener, Bâtiment B4, atelier 68, 75018 Paris, France
`raffard.roussel@gmail.com`

Abstract

In this paper, we describe OuPoCo (*l'Ouvroir de Poésie Combinatoire*), a system producing new sonnets by recombining verses from existing sonnets, following an idea that Queneau described in his book *Cent Mille Milliards de poèmes, Gallimard, 1961*. We propose to demonstrate different outputs of our implementation (a Web site, a Twitter bot and a specifically developed device, called *La Boîte à poésie*) based on a corpus of 19th century French poetry. Our goal is to make people interested in poetry again, by giving access to automatically produced sonnets through original and entertaining channels and devices.

1 Introduction

The starting point of the OuPoCo project is the book by Raymond Queneau *Cent Mille Milliards de poèmes* (*A Hundred Thousand Billion Poems*, in English) (Queneau, 1961). The book is composed of ten sonnets printed on separate strips, allowing thus a hundred thousand billion of different readings. The combinatory poetics characteristic of this book encourages the reader to play with the meanings of the poems, the various language registers present in the book or simply with its movable form. It is this stimulating relationship between poetry and constraints, which constituted the central mainspring of this project.

Queneau's book seems an ideal candidate to be transposed on a computer since a machine can perfectly combine verses and produce a comprehensive list of all the possible poems. But Queneau's work is still under copyright, which prevent us from working directly on the sonnets contained in his book. This is the reason why, instead of working on Queneau's poems, we decided to focus on a collection of 19th century French poetry instead. Beyond the availability of massive databases in this field, which facilitated the creation of our corpus, 19th century poetry also appeared as a fertile ground to reflect upon the sonnet and to work and play with various constraints. This means that original poems have to be analyzed so as to determine the rhyme, as well as other features (length of a verse, topics addressed, etc.).

This paper is thus about OuPoCo, *l'Ouvroir de Poésie Combinatoire*, a system able to generate sonnets by recombining verses taken from a corpus of French 19th century poetry. We describe the project, the different project outputs and the interest of this experience to reconnect people with poetry and literature in an entertaining way.

2 The Corpus

The first part of this research consisted in gathering a corpus of French sonnets from the 19th century. We first used available resources freely available on the Web, especially the Gutenberg project, Wikisource and Gallica, a rich collection of digitized resources provided by the Bibliothèque nationale de France

(the French national library). The BnF later gave us access to an even larger corpus of French poems[1], so that the implementation currently integrates a corpus of more than 4,000 French sonnets.

All major French authors from the 19th century are included in the database, but also some less known ones. Each sonnet is encoded in a XML format along with related metadata; a TEI version of the database is publicly available (see `https://github.com/clement-plancq/oupoco-api`) and is regularly expanding.

3 Corpus and Rhyme Analysis

The OuPoCo project has nothing to do with the recent neural approach to poetry generation (Ghazvininejad et al., 2017; Van de Cruys, 2020), but it requires to get access to a formal representation of rhymes (as proposed by (Beaudouin, 2002)). In order to do this, the first step is to get a phonetic transcription of the last word of each verse, but this is not enough: for example, "*aimé*" and "*aimée*" have the same phonetic transcription, but do not rhyme, according to French rhyming rules (feminine and masculine words, for example words that end with -*é*, as opposed to -*ée*, do not rhyme); there are also cases where the phonetic transcription is slightly different but words actually rhyme (for example with sounds like [e] and [ε]). All these cases are not marginal and must be handled appropriately.

Phonetisation (the process of transforming a word into a phonetic transcription) is done with eSpeak, a free software available on the Web (`http://espeak.sourceforge.net`), that provided satisfactory results on our data. We analyse the whole verse and not only the last word of each verse, so that a full rythmic analysis is possible. However, as we have just seen, the phonetic transcription provided by eSpeak is not enough.

A series of rules written in Python had thus to be defined to get a proper analysis of rhyme derived form the phonetic transcription of the last word of each verse. These rules are manually defined and maintained, as there is no way they could be learnt directly from the data. A part of these rules can be easily derived from a treaty of French versification, but another part is directly linked to the output of eSpeak as we have to overwrite some phonetic distinctions produced by this software that are not relevant to analyse poetry.

The sonnet generator uses this analysis to produce sonnets, with different possible structures, respecting the rules of French versification (the code and the resources used, especially the sonnet database, are open source and freely available for research, see: `https://github.com/clement-plancq/oupoco-api`).

4 Constrainsts

In the footsteps of the OuLiPo and following the comments of Queneau who did not like the idea of pure random poetry generation, we chose to implement constraints to enable the reader to interact with the database, control the generation process in different ways, and discover 19th century French literature from a different and more playful angle.

The first constraint correspond to the different existing forms of sonnets proposed in the course of history. Giacomo da Lentini, Petrach, Marot, Peletier, Shakespeare and Spencer all proposed and initiated a slightly different rhyming scheme (for example, Marot proposed the following structure: ABBA ABBA CCD EED, while Petrarch proposed several slightly different structures: ABBA ABBA CDE CDE / ABBA ABBA CDC DCD / ABBA ABBA CDE DCE, etc.). All these forms of sonnets are available and the user can choose any structure s/he prefers for generation.

The second constraint enables the reader to generate random sonnets from texts within a chosen time framework. In other words, the reader selects a period of twenty or thirty years for example that s/he wants to explore on the timeline, and sonnets are generated according to this time frame. This option makes it possible to have a quick overview of the productions of French sonnets over a period. Thanks to this option, it is for example possible to note that very few French sonnets were written at the beginning of the 19th century while their production tends to increase 20 years or so later.

---

[1] This corpus is available for research on the Bibliothèque nationale de France website: `api.bnf.fr/sonnets-de-gallica`

The third constraint deals with the authors themselves. The reader can select one or multiple poets as the source corpus to generate new sonnets. It is thus possible to explore the poems written by Baudelaire for instance, or to combine them with the ones written by Rimbaud or Verlaine.

The fourth constraint is based on a semantic analysis of the original sonnets. Six main themes were identified (beauty, love, death, nature, spirituality), later on reduced to five, as melancholy and death were hard to distinguished. A sample of sonnets were annotated using these categories (50 sonnets per category) and a classifier trained on the manually annotated corpus. It is thus possible to generate sonnets based on the part of the corpus that have been classified as pertaining to a specific theme. Note that the annotation operates at the sonnet level, whereas generation operates at the verse level. However, we assume that the theme gives a general flavor to the text, not every verse has to be relevant from a thematic point of view.

The approach is quite simple from a computational point of view. However, it is difficult to control on the fly the number of sonnets that can be generated, depending on the number of contraints chosen by the end user. It is however important to keep track of this, otherwise the end user may frequently arrive at a dead end, with a number of contraints that prevents the possibility to generate new sonnets (a basic rule being that one verse cannot rhyme with itself, and even cannot be selected twice).

5 Overview of the Demonstration

The main interest of the OuPoCo project is to present French poetry through a new and original setting. With our system, poetry is not any more just a literary genre (Derrida and Ronell, 1980), but a dynamic object that can be manipulated and experienced. For lots of people, poetry is seen at best as something related to school years, at worse as something boring and uninteresting from the past. Our new setting, in itself, makes it possible to show that playing with poetry can be fun. Our setting puts in perspective the notion of text coherence (Reinhart, 1980) since the result of the generator can be more or less satisfactory from a semantic point of view.

It is possible to interact with OuPoCo through a web site (https://oupoco.org/fr, see figure 1).
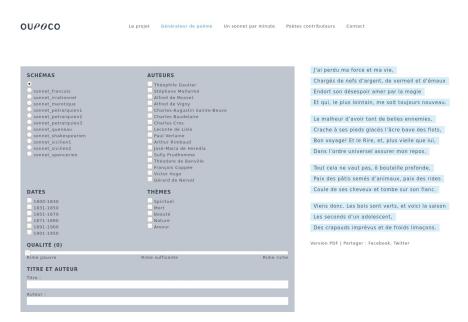

Figure 1: An overview of the system on the Web. On the left, the set of possible constraints; on the right, an example of generated sonnet

It is also possible to regularly have a look at the bot posting a quatrain every 6 hours on Twitter (see figure 2). Finally, we also had a collaboration with a duo of artists who produced a "poetry box" (*La boîte à poésie*, see figure 3), a portable version of the original idea that can be demonstrated in public

events (based on Raspberry Pi components). Through these devices our goal is to reach a wider audience and engage people to reconnect with poetry.

Figure 2: The Oupoco bot on Twitter

6 Evaluation and Interest of the System Output

At the crossroad between surrealism and absurdism, generated sonnets are generally quite funny and convey a dreamlike atmosphere. The themes specific to the Romantic period – such as love, whether it is magnified or lost, death or the fleetingness of time to name but a few – contribute to creating bizarre but intriguing poems. If most of them lack coherence in terms of punctuation or pronouns, these syntactic confusions can actually reinforce their poetic overtones.

The OuPoCo system is intended to be presented in front of an audience, to elicit reactions. We did not perform a formal evaluation, as we would be unable to provide meaningful evaluation criteria (Gervás, 2013) and moreover, as this is not the goal. The interest is to create reactions, to use our piece of software as a mean to make people rediscover poetry and literature.

One should also note that, because the machine produces structurally impeccable sonnets, the experiencer is unconsciously encouraged to find coherence in them, simply because we are used to coherence in our everyday life and because incoherence is bewildering (Reinhart, 1980). The second consequence is a frequent need for the experiencer to go back to the original poem, to see where from a given verse originates (tooltips always allows the experiencer to go back to the original sonnet). The project is thus not just a sacrilege game over venerated texts, but a way to make people experience and rediscover poetry.

7 Conclusion

We have described OuPoCo, a system inspired by Queneau and implemented through different devices. It aims at reconciling people with literature, especially poetry, a genre that is not very popular outside the educational system. The project was also the opportunity of a collaboration with a couple of artists who produced *La Boîte à poésie*, an interesting spin-off of the project mixing art and technology. In the future, we plan to study the potential impact of our system in different (real world) contexts, especially in educational settings.
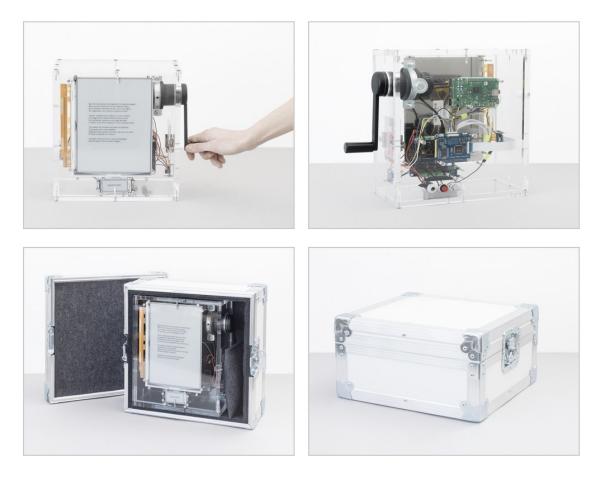
Figure 3: La Boîte à poésie, integrating the OuPoCo sonnet generator. This device has been developed by Atelier Raffard-Roussel, a couple of artists based in Paris. See `http://www.raffard-roussel.com/fr/projets-boite-a-poesie/` for details.

Acknowledgements

References

Valérie Beaudouin. 2002. *Mètre et rythmes du vers classique. Corneille et Racine*. Honoré Champion (Lettres numériques), Paris.

Jacques Derrida and Avital Ronell. 1980. On narrative: The law of genre. *Critical Inquiry*, 7(1):55–81.

Pablo Gervás. 2013. Computational modelling of poetry generation. In *Artificial Intelligence and Poetry Symposium, AISB Convention*, University of Exeter.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, pages 43–48, Vancouver, Canada.

Raymond Queneau. 1961. *Cent mille milliards de poèmes*. Gallimard, Paris.

Tanya Reinhart. 1980. Conditions for text coherence. *Poetics Today (Narratology II: The Fictional Text and the Reader)*, 1(4):161–180.

Tim Van de Cruys. 2020. Automatic poetry generation from prosaic text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Held online.