WOPE!
A tool for poetry composition through metaphoric text generation with Language Models

Pablo Pérez Benavente and **Pavlos Andreadis**
pabloperezbenavente@gmail.com pavlos.andreadis@ed.ac.uk
University of Edinburgh

Abstract

Traditionally, research on Automatic Poetry Generation has tried to apply formal templates to language in the most natural-sounding way. In this work we aim at capturing the use of language that is characteristic of poetry itself. We do so by identifying poetic genre with metaphors, and define metaphors as cross-domain mappings. We design a system where users generate poems by selecting the domains through which metaphors will emerge. Our method proves to be useful to control the presence of more than one domain in a text and to generate metaphors. Although poeticness of the resulting texts is vague if the system is used autonomously, user engagement results in texts that are deemed more poetic than those of other intelligent poets.

1. Introduction

Past research on Automatic Poetry Generation has mostly been focused on structural aspects of poetry, with rhetoric aspects being largely forgotten (Liu et al. 2019). This work tries to fill in that gap through metaphoric text generation. Our system struggles to create poems autonomously, but when used as a tool in a co-creative way, generated texts are perceived as more poetic than other state of the art poetry generation systems.

This work is structured as follows. In section 2 we describe the methodology that was followed to define and tackle the problem of Automatic Poetry Generation. In section 3 we walk through our data collection procedure and the experiments that we carried out, and in section 4 we describe the results. Section 5 summarizes related previous work in the field, and section 6 draws the main conclusions.

2. Methodology

The problem of defining poetry relies in its fluid nature. We overcome this by following the formalist theory according to which poetry 'defamiliarizes' concepts (Shklovskiĭ 2019), typically through metaphors (Oita 2019), (Willis 2002). We define metaphors as mappings across conceptual domains (Lakoff 1993) and identify the "conceptual domain" of a text with its topic. Mappings across conceptual domains are then mappings across topics.

We hypothesize that whenever a text incorporates various topics, mappings -and thus metaphors- will naturally emerge. Thus, if we want to generate metaphors, we simply have to find a way to incorporate multiple topics in a text. We can do so through controlled-text generation methods (Garbacea and Mei 2020), where the output of a Language Model is shifted to steer the generated language. Previous work (Pascual et al. 2021) (Ghazvininejad et al. 2017) has shown that in a transformer-based model the topic of a text can be controlled by increasing the scores output by the model for topic-related words by a certain value α (Pascual et al. 2021) (Ghazvininejad et al. 2017) as shown the following equation:

$$score_{final} = score_{output} + \alpha$$

But as predicted by (Garbacea and Mei 2020), multiple constraints result in text degradation. So we introduced a second value β that controls the number of generation time-steps that are left untouched, reformulating the equation as follows:

$$score_{final} = score_{output} + (\alpha \times \mathbb{1}^{t\%\beta=0})$$

where t indicates the generation time-step. If a constrain p has a β of 1, then only half of the generation time-steps will see their scores for p-related words increased by α. Experiments show that by adjusting α and β we can successfully incorporate various topics avoiding text degradation. The similarities between these values and the intensity and period of a wave give name to our system: *WOPE* (Waved Output for Poetic Experimentation).

3. Experiment setup

We implement these topic constraints through HuggingFace Transformers Library[1] on top of GPT2 (Radford et al. 2019). Topic-related words are computed as the 100 words with highest cosine similarity to the topic word. To generate poems, for each topic we randomly sample an α-β pair; a topic from a subset of 52 labels obtained by (Van de Cruys 2020); and a poem beginning from the Gutenberg Poetry Corpus[2]. The first verse is conditioned on the poem

[1] https://huggingface.co/docs/transformers/index
[2] https://gutenberg.org/

beginning, and subsequent verses are conditioned on verses already generated. We generate 4 verses, each with 15 words. With this procedure we generate 50 poems that form our test data.

Experiment 1 evaluates text correctness and domain incorporation. Correctness is measured through a BERT-based model trained on CoLA[3], and domain incorporation through domain coherence (Boggia et al. 2022) and topic generalization (Yang and Klein 2021). Domain incorporation is computed by averaging the scores obtained by each of the incorporated topics, and total results are compared with poems generated by (Boggia et al. 2022) (*Boggia*) and (Van de Cruys 2020) (*Cruys*).

Experiment 2 examines how two topics with different $\frac{\alpha}{\beta}$ show up differently in the same text. To do this, each poem is associated with two values. First, the difference in $\frac{\alpha}{\beta}$ value between the two incorporated topics. Then, the difference in domain coherence score obtained by those two topics. The relation between those two variables across poems will evaluate whether α and β can be used to control the prominence of a domain.

Experiment 3 consists in a 30 minute co-creative session. 25 participants from the University's Informatics department mailing list who agreed to take part on the study were asked to go through a Python notebook. This walked them through poem generation and selecting topics, α-β value pairs and a prompt. Participants created their own poems and were given a question form adapted from (Mirowski et al. 2022) with statements regarding their impressions on the system. They were asked to rate those statements from 1 to 5 according to their level of agreement. At the end of the session they were told that a different part of the study required poems generated with *WOPE* by humans, and some decided to voluntarily provide the poems they had generated.

Experiment 4 used poems from experiment 3 as well as the test data used in experiments 1 and 2 to evaluate *WOPE* as a poetry generation system. 10 participants who replied to the mailing list but did not participate in experiment 3 were asked to rate them from 1 to 5 according to their fluency, coherence, meaningfulness and poeticness (Zhang and Lapata 2014). Apart from $WOPE_{auto}$ and $WOPE_{user}$, we also included poems generated by *Boggia*, by *Cruys* and real poems extracted from the Gutenberg project.

4. Results

Domain coherence and topic generalization metrics used in experiment 1 gave similar results for our test data (0.39, 0.003), *Boggia* (0.35, 0) and *Cruys* (0.34, 0.009). *WOPE*'s grammaticality (0.55) is similar to *Boggia* (0.45) but still far from that obtained by *Cruys* (0.76). In experiment 2, we group poems according to the $\frac{\alpha}{\beta}$ difference between their two topics and observe that indeed this translates into higher difference in domain prominence. In experiment 3, higher acceptance was given to statements related to ease of using WOPE, enjoyment and surprise, while less accepted ones were those related with helpfulness, pride or ownership towards the generated artifacts. Users highlighted "word associations" and that "elements of metaphor were clearly realized" while also acknowledging that "poems end abruptly" and "overall meaning is not consistent". Regarding experiment 4, $WOPE_{auto}$ obtained the lowest scores of all for meaningfulness (2.62) and poeticness (1.95), both compared to *Boggia* (3.21, 3.70) and *Cruys* (2.92, 3.21). However, $WOPE_{user}$ obtained the highest scores for both of these categories (3.46, 3.71), only below human baseline (3.60, 4.12).

5. Previous work

A slightly-outdated review on Poetry Generation can be found in (Gonçalo Oliveira 2017). Most previous work has focused on generating text with particular formal requirements, namely meter and rhyme. Some recent work has leveraged constrained generation techniques (Yang and Klein 2021). (Van de Cruys 2020) uses these methods to enforce both rhyme and topic. Topic alone has been controlled indirectly in (Pascual et al. 2021) (Ghazvininejad et al. 2017). Very few works have attempted to generate poetic text without formal constraints. (Bena and Kalita 2020) trained an LM on dream-like descriptions and fine-tune it on poems with specific emotions to generate dream-like poems with the desired emotions. (Oita 2019) and (Liu et al. 2019) gather a corpus of labeled metaphors and then train a LM to reproduce them. Automatic generation of metaphors has also been attempted through lexical templates and word co-occurrences (Veale 2016) (Galván et al. 2016).

6. Conclussions

Results show that our implementation is capable of modeling the presence of different domains while still generating fluent text, at least in the same range as other poem generators. Additionally, our method can be used to adjust the presence of a given domain. When it comes to generating poetry, if deployed autonomously, poeticness of the generated texts is vague, probably because it depends on factors such as a) the discourse structure of the text, which we have not modeled at all, and b) the poetic that is inherent to certain words. Thus, it is probably incorrect to state that poeticness lies entirely in metaphoricity, although users confirm that metaphors were generated. However, when used as a tool by a user, texts are more likely to be seen as poetic than other systems, and users experience feelings of enjoyment and surprise. The reason why such a need for engagement does not translate in ownership or pride towards the created artifacts is left as a future line of research.

7. Acknowledgements

This project received funding from the University of Edinburgh MSc Speech and Language Processing. The project

[3]https://textattack.readthedocs.io/en/latest/

that gave rise to these results received the support of a fellowship from "La Caixa" Foundation (ID 100010434). The fellowship code is LCF/BQ/EU22/11930014.

References

Bena, B., and Kalita, J. 2020. Introducing aspects of creativity in automatic poetry generation.

Boggia, M.; Ivanova, S.; Linkola, S.; Kantosalo, A.; and Toivonen, H. 2022. One line at a time — generation and internal evaluation of interactive poetry. In Hedblom, M.; Kantosalo, A.; Confalonieri, R.; Kutz, O.; and Veale , T., eds., *Proceedings of the 13th International Conference on Computational Creativity*, 7–11. International: The Association for Computational Creativity. International Conference on Computational Creativity, ICCC ; Conference date: 27-06-2022 Through 01-07-2022.

Galván, P.; Francisco, V.; Hervás, R.; Méndez, G.; and Gervás, P. 2016. Exploring the role of word associations in the construction of rhetorical figures.

Garbacea, C., and Mei, Q. 2020. Neural language generation: Formulation, methods, and evaluation. *ArXiv* abs/2007.15780.

Ghazvininejad, M.; Shi, X.; Priyadarshi, J.; and Knight, K. 2017. Hafez: an interactive poetry generation system. In Bansal, M., and Ji, H., eds., *Proceedings of ACL 2017, System Demonstrations*, 43–48. Vancouver, Canada: Association for Computational Linguistics.

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

Lakoff, G. 1993. The contemporary theory of metaphor. In Ortony, A., ed., *Metaphor and Thought*. Cambridge University Press. 202–251.

Liu, Z.; Fu, Z.; Cao, J.; de Melo, G.; Tam, Y.-C.; Niu, C.; and Zhou, J. 2019. Rhetorically controlled encoder-decoder for Modern Chinese poetry generation. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1992–2001. Florence, Italy: Association for Computational Linguistics.

Mirowski, P.; Mathewson, K.; Pittman, J.; and Evans, R. 2022. Co-writing screenplays and theatre scripts with language models: An evaluation by industry professionals.

Oita, M. 2019. *Incremental Alignment of Metaphoric Language Model for Poetry Composition*. 834–845.

Pascual, D.; Egressy, B.; Meister, C.; Cotterell, R.; and Wattenhofer, R. 2021. A plug-and-play method for controlled text generation. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Findings of the Association for Computational Linguistics: EMNLP 2021*, 3973–3997. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners.

Shklovskiĭ, V. B. 2019. Art as technique. *From Symbolism to Socialist Realism*.

Van de Cruys, T. 2020. Automatic poetry generation from prosaic text. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2471–2480. Online: Association for Computational Linguistics.

Veale, T. 2016. *3. The shape of tweets to come: Automating language play in social networks*.

Willis, P. 2002. Don't call it poetry. 2.

Yang, K., and Klein, D. 2021. Fudge: Controlled text generation with future discriminators.

Zhang, X., and Lapata, M. 2014. Chinese poetry generation with recurrent neural networks. 670–680.