Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Early Career Track

Controllable Text Generation for Open-Domain Creativity and Fairness

Nanyun (Violet) Peng
Computer Science Department, University of California, Los Angeles
violetpeng@cs.ucla.edu

Abstract

Recent advances in large pre-trained language models have demonstrated strong results in generating natural languages and significantly improved performances for many natural language generation (NLG) applications such as machine translation and text summarization. However, when the generation tasks are more open-ended and the content is under-specified, existing techniques struggle to generate long-term coherent and creative content. Moreover, the models exhibit and even amplify social biases that are learned from the training corpora. This happens because the generation models are trained to capture the surface patterns (i.e. sequences of words), instead of capturing underlying semantics and discourse structures, as well as background knowledge including social norms. In this paper, I introduce our recent works on controllable text generation to enhance the creativity and fairness of language generation models. We explore hierarchical generation and constrained decoding, with applications to creative language generation including story, poetry, and figurative languages, and bias mitigation for generation models.

1 Introduction

Natural language generation (NLG) tasks can be mapped into a spectrum based on their conditional entropy, i.e., the uncertainty of the output distribution given the inputs. One side of the spectrum is the low conditional entropy tasks, such as machine translation, abstractive summarization, and task-oriented dialogue systems, where the inputs largely determine the contents of the outputs. On the other side is the tasks that are open-ended and have high conditional entropy, such as story, poetry, or lyric generation given a title or a prompt, and open-domain dialogue systems (chit-chat), dubbed *creative generation*. In addition to be able to compose grammatical and fluent sentences to articulate given contents, these tasks usually also require extensive world and common sense knowledge, discourse-level coherence modeling to make sure the outputs are long-term coherent, sensible, and creative, making them especially challenging NLG tasks.

To tackle the challenges, my group has been working on several synergistic directions to push the frontier of open-domain (creative) generation, with a shared modeling theme of controllable text generation and applications to story, poetry, and dialogue response generation.

The first direction is to build hierarchical models that disentangle the content planning (plan out the structured representation of the content) from surface realization (convert the structured content to natural language sentences). This has three major benefits: i). It enhances the controllability of the generation through the plan and humans can more easily collaborate with the models on the plan-level to create novel contents. ii). It improves long-term conherence of the generation, since the compact plan representation helps the model to capture the high-level outline of the generation target, and iii). It enables the introduction of creativity on both the planning-level (e.g., better events and their temporal and causal relation modeling to introduce surprise) and the diction-level (e.g., using figurative language to improve the writing). More details about this direction will be discussed in Section 2. Additionally, we develop fundamental models to support the controllable, hierarchical generation pipeline. i) We design insertion-based models that break the left-to-right generation order to better incorporate constraints/control factors (more details in Section 3). ii) We propose various constrained decoding algorithms to control the models. Finally, we have also been working on analyzing and reducing biases in open-domain generation (more details in Section 4).

2 Hierarchical Text Generation

Hierarchical text generation follows the *plan-and-write* [Yao *et al.*, 2019] paradigm to first plan out the main content of the output, and then convert these structured plans into natural language texts (also called surface realization). The idea of content planning for text generation dates back to the late 70s [Meehan, 1976] with various plan representations. Prior works predominately rely on *manually* constructed or *rule-based* plans, which are either expensive to construct, or restricted to specific domains.

2.1 Automatic Content Plan Extraction

My group has been exploring the direction of learning content planning from existing corpora by using information extraction tools to extract "silver standard" plan representations from data for models to learn planning [Yao *et al.*, 2019; Goldfarb-Tarrant *et al.*, 2020; Han *et al.*, 2022]. Specifically, we have extensive works on extracting entities [Huang *et al.*, 2019; Huang *et al.*, 2021], entity relations [Peng *et al.*, 2017; Hsu *et al.*, 2022a], events [Ahmad *et al.*, 2021; Hsu *et al.*, 2022b], and event temporal relations [Han *et al.*, 2019; Han *et al.*, 2020; Han *et al.*, 2021] from text corpora, to compose "silver standard" content plans for models to learn and generate novel plans during the test time.

2.2 Content Planning with Temporal Modeling, Literary Principles, and Surprise

While content plans have shown to be effective in improving the coherence of the generation [Yao *et al.*, 2019], prior works also have shown that generated plans are much lower quality than the silver plans as they are usually repetitive, boring, and violate common sense and world knowledge [Martin *et al.*, 2017; Yao *et al.*, 2019; Fan *et al.*, 2019]. To this end, we have been working on improving the planning model by introducing knowledge about literary principles [Goldfarb-Tarrant *et al.*, 2020], event temporal knowledge [Han *et al.*, 2022], and common sense knowledge. Specifically, we represent the plan generation model as a probabilistic model $p(\mathbf{z}|\mathbf{x})$ that is fitted by auto-regressive generative neural networks, where $\mathbf{z}$ represents the structured plan, $\mathbf{x}$ represents a given prompt. We modify the decoding objective to incorporate several rescoring models $a \in A$ to re-rank the original, or "naive" plans generated by the graph generative model. These rescoring models bring the generated plan graph closer to each rescoring model's specialty (such as relevance, coherence, and temporality). The modified decoding objective becomes:

$$f_\lambda(\mathbf{x}, \mathbf{z}) = \sum_{i}^{m} -\log p(z|z < i, \mathbf{x}) + \sum_{j}^{|A|} \lambda_j a_j(\mathbf{x}, z_{i\ldots m})$$

where λ_j is the learned weight of the score given by a_j. What differs for each model a_j that specializes in a different rescoring aspect is the set of training data we generated to train the discriminators.

We also explored improving the interestingness of the generation, by introducing temporal diversity to the events, such as *flashbacks* that insert past events into current storylines as we commonly observe in novels and plays [Han *et al.*, 2022]. It is challenging for machines to generate *flashbacks* as it requires solid understanding of event **temporal order** (e.g. *feeling hungry* ⟨*before*⟩ *eat*, not vice versa), and the creativity to arrange storylines so that earlier events do not always appear first in **narrative order**. Two major issues in existing systems exacerbate the challenges: 1) temporal bias in pretraining and datasets that leads to monotonic event temporal orders; 2) lack of explicit guidance that helps machines decide where to insert *flashbacks*. We address these issues by introducing *temporal prompts* to the structured plans to encode events and their pair-wise temporal relations (⟨*before*⟩, ⟨*after*⟩ and ⟨*vague*⟩) to guide how stories should unfold temporally. We showed that temporal prompts helped generate more interesting stories with *flashbacks* while maintaining textual diversity, fluency and temporal coherence.

2.3 Introducing Literary Aesthetics in Writing

Given a well-crafted plan, the other important component for the hierarchical generation is surface realization that converts plans into natural language sentences. The state-of-the-art pre-trained language models [Radford *et al.*, 2019; Lewis *et al.*, 2020] are capable of generating grammatical and fluent sentences, but they are usually plain and dull. To improve aesthetics in writing, we have explored incorporating figurative languages into the generation.

Figures of speech are literary devices that are commonly employed in narratives and stories [Boudens, 2005]. Proper usage of figurative languages can improve the effectiveness, interestingness, and enjoyment of communications [Stein *et al.*, 2018]. However, the composition of figures of speech usually requires extensive contextual, common sense, and world knowledge [Justo *et al.*, 2014; Yoshimura *et al.*, 2015], which remains an open challenge for NLG. My group has been working on generating puns, similes, sarcasms, and metaphors [He *et al.*, 2019; Chakrabarty *et al.*, 2020a; Chakrabarty *et al.*, 2020b; Chakrabarty *et al.*, 2021; Stowe *et al.*, 2021; Mittal *et al.*, 2022], and incorporate them into story [Chakrabarty *et al.*, 2020b] and poetry [Chakrabarty *et al.*, 2021; Tian and Peng, 2022] generation to improve the aesthetics of the writing.

2.4 A Showcase: Zero-Shot Sonnet Generation

As a concrete example, in [Tian and Peng, 2022], we proposed a *zero-shot* sonnet generation framework that generates high-quality sonnets without training on any poetic data. Our framework consists of four components: content planning, rhyme pairing, polishing for aesthetics, and final decoding. The first three steps provide salient points for the sketch of a sonnet. The last step is responsible for "translating" the sketch into well-formed sonnets.

To achieve zero-shot generation, the content planning and the final decoding components are both trained on a combination of news and story corpora. The trained planning module is aimed to generate several keywords for each sentence to equip the system with *general world knowledge to construct a coherent text world*. However, the language used by poems is different from that of standard texts because it follows certain rhetorical rhythm and is full of vivid descriptions that appeals to readers' senses and imagination [Gibbs Jr, 1994]. To this end, in the polishing step we incorporate two figurative speeches (i.e., simile and imagery) into the planned keywords to boost vividness and imagination. As the final step, a constrained decoding algorithm is designed to impose the meter-and-rhyme constraints while incorporating the plan keywords. Human evaluation shows that our model generates more discourse-level coherent, poetic, creative, and emotion-evoking sonnets than strong baselines.

3 Controllable Text Generation

One advantage of the hierarchical generation framework is the improved controllability of the generation through the plan – human writers can more easily collaborate with the models on the plan-level to control the generation contents.

However, it is not straightforward to incorporate such controls in the state-of-the-art language generation models. The pre-trained auto-regressive language models [Radford *et al.*, 2019; Lewis *et al.*, 2020] usually generate sentences word by word from left to right, making the control of the generation process challenging.

3.1 Insertion-based Generation

Prior works [Fan *et al.*, 2019; Yao *et al.*, 2019] observed that fine-tune sequence-to-sequence models, while works decent for controllable language generate, cannot guarantee faithful incorporation of the plan. For example, we [Yao *et al.*, 2019] showed that there are slightly less than 80% of storyline keywords (a form of plan) appeared in the generated story. [Fan *et al.*, 2019] thus designed special verb-attention mechanism and incorporated copy mechanism to enhance the incorporation of the plans in the generation outputs. In light of these observations, we have been exploring a fundamentally different direction of controllable text generation framework – insertion-based text generation, or text infilling.

We proposed a general algorithm for *efficient* insertion-based text generation [Lu and Peng, 2021] to train a permutation language model with a delicate design of the permutations to reflect the insertion orders. To suit the non-monotonic nature of the insertion-based generation process, a modified relative positional encoding mechanism is introduced such that each token is only aware of its relative position with respect to the generated partial sequence, but not the relative position with respect to the complete sequence. We showed the training efficiency, controllability, and the advantages of the model to generalize over partial observations. We plan to explore other applications for this novel *efficient* insertion-based NLG formulation and build a large-scale pre-trained insertion-based NLG model for lexically constrained text generation.

3.2 Controllable Decoding

There are also methods to control auto-regressive language models to leverage the powerful pretraining. They can be summarized into three major categories: *fine-tuning*, *refact/retraining*, and *post-processing*. The first two categories are usually inefficient considering the size of language models (billions of parameters). Therefore, my group has been mostly developingg post-processing based models to control pre-trained auto-regressive language models. We have been pushing two lines of research: 1) adapting the decoding algorithm to algorithmically encourage models to incorporate constraints [Sheng *et al.*, 2021; Tian and Peng, 2022] and 2) using auxiliary models to guide the decoding by changing the output distribution [Goldfarb-Tarrant *et al.*, 2020; Chakrabarty *et al.*, 2021]. We plan to continue exploring and dive deeper in these two directions.

4 Evaluating and Mitigating Biases in Open-Domain Text Generation

While large pre-trained language models have learned to produce human-like writings from training on massive amounts of free online text corpora, they have simultaneously inherited or even amplified social bias in them [Sheng *et al.*, 2019; Shah *et al.*, 2020; Bender *et al.*, 2021], such as using derogatory words (e.g. "naughty", "stupid") near female pronouns and hate-mongering words (e.g. "terrorism", "slaughter") near "Islam". When applying to open-domain, creative text generation, these problems are more pronounced, because the high-entropy nature of the output distribution given the input grant the language models more freedom to generate biased or offensive contents.

As NLG applications *directly interact* with diverse user groups to communicate valuable information in various domains (e.g., writing assistant systems, emotion support chatbot, and automatic email reply), these biased and offensive machine generated texts would have a profound impact on society by shaping people's beliefs and values [Hovy and Spruit, 2016]. Specifically, systematically biased outcomes against certain populations run the risk of inadvertently amplifying inequality through unequal treatment of different groups, harming user experience, and increasing social polarization.

There are several unique challenges in evaluating fairness of NLG systems. First, unlike classification problems, NLG problems face a high-dimensional output space (i.e. all possible sentences that are grammatically correct), thus fairness cannot be defined by system performance (e.g. accuracy) gaps alone. Instead, the notion of fairness has to be established on the *perception* of the outputs that delineate their rich connotations in various social and cultural contexts. It thus must consider a multifaceted measure of fairness. For example, a sentence can be unfair to women because it makes blatant sexist comments (toxicity), depicts women as weak and subordinate (stereotypes), or drives away female users because of unfriendly content (inclusiveness). Second, the generation process is usually stochastic, meaning that given the same context, the generated text could be different. Taking open-domain dialog system as an example, with the same conversational context "how's going", the same dialog system can generate different valid responses such as "Going well", "I'm good", "It's great". The evaluation of the systems' fairness needs to take such stochasticity into consideration. Finally, due to the close interplay between language and social hierarchy, language ideologies play a vital role in deciding how social groups are labeled and what language use is considered offensive, which are largely influenced by the dominant group. Consequently, as argued by [Blodgett *et al.*, 2020], research in language fairness risks "measuring or mitigating only what is convenient to measure or mitigate, rather than what is most normatively concerning." Therefore, bias curation should be an interactive and continual process with humans in the loop.

My group pioneered and have been actively exploring automatic evaluation of fairness aspects of open-domain generation, including social stereotypes [Sheng *et al.*, 2019; Sheng *et al.*, 2020] and ad-hominem (a type of micro-aggression that is considered as toxic language) [Sheng *et al.*, 2021] of open-domain generation. We define "fair" as *similar distributions* of features (e.g., sentiment, toxicity, regard, etc.) of the generation outputs involving different groups, and consequently, "bias" as discrepancies between the distributions of the features depicted in generated texts that involve different groups. To measure biases in generated texts, we built surrogate classifiers to extract features (e.g., sentiment, toxicity) of the generated texts. We also develop controllable text generation models to reduce biases in generated texts [Sheng *et al.*, 2020; Sheng *et al.*, 2021].

5 Conclusion and Future Work

There are many exciting ongoing research directions in my group to push the frontier of natural language understanding and generation, with focused applications to creative and controllable text generation. Due to the space limit, I only briefly highlight three of them: 1) We are developing generation-based model for information extraction tasks. This can be applied to general natural language understanding, as well as better plan extraction for hierarchical generation. 2) We are trying to incorporate common sense knowledge into the control using common sense knowledge bases, semantic loss, and lexically constrained generation. and 3) We are diving deeper into insertion-based text generation and decoding-based controllable generation to adapt large pre-trained auto-regressive language models for controllable/constrained generation tasks. We also have ambition to pre-train the first insertion-based language model on large corpora.

Acknowledgments

My research has been generously supported by DARPA CwC, MCS, and KAIROS programs, the IARPA BETTER program, an NIH R01, and several industrial awards including CISCO, Amazon Alexa AI, and JP Morgan Chase. The views expressed in this paper are mine and do not reflect the views of the U.S. Government nor the company sponsors. I am indebted to my advisors, mentors, senior collaborators for their help, support, and coach. The works presented in this paper would not have been possible without my PhD students and other coauthors. Finally, I thank IJCAI 2021 program committee for the invitation to speak at the early-career spotlight.

References

[Ahmad *et al.*, 2021] Wasi Ahmad, Nanyun Peng, and Kai-Wei Chang. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021.

[Bender *et al.*, 2021] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

[Blodgett *et al.*, 2020] S. L. Blodgett, S. Barocas, H. Daume III, and H. Wallach. Language (technology) is power: A critical survey of "bias" in NLP. In *Association for Computational Linguistics (ACL)*, 2020.

[Boudens, 2005] Connie J Boudens. The story of work: A narrative analysis of workplace emotion. *Organization Studies*, 26(9):1285–1306, 2005.

[Chakrabarty *et al.*, 2020a] Tuhin Chakrabarty, Debanjan Ghosh, Smaranda Muresan, and Nanyun Peng. r^3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge. In *ACL*, 2020.

[Chakrabarty *et al.*, 2020b] Tuhin Chakrabarty, Smaranda Muresan, and Nanyun Peng. Generating similes effortlessly like a pro: A style transfer approach for simile generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[Chakrabarty *et al.*, 2021] Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. Mermaid: Metaphor generation with symbolism and discriminative decoding. In *The 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

[Fan *et al.*, 2019] Angela Fan, Mike Lewis, and Yann Dauphin. Strategies for structuring story generation. In *ACL*, 2019.

[Gibbs Jr, 1994] Raymond W Gibbs Jr. *The poetics of mind: Figurative thought, language, and understanding*. Cambridge University Press, 1994.

[Goldfarb-Tarrant *et al.*, 2020] Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. Content planning for neural story generation with aristotelian rescoring. In *the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[Han *et al.*, 2019] Rujun Han, Qiang Ning, and Nanyun Peng. Joint event and temporal relation extraction with shared representations and structured prediction. 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)*, pages 434–444, 2019.

[Han *et al.*, 2020] Rujun Han, Yichao Zhou, and Nanyun Peng. Domain knowledge empowered structured neural net for end-to-end event temporal relation extraction. In *the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020.

[Han *et al.*, 2021] Rujun Han, Xiang Ren, and Nanyun Peng. Econet: Effective continual pretraining of language models for event temporal reasoning. In *The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[Han *et al.*, 2022] Rujun Han, Hong Chen, Yufei Tian, and Nanyun Peng. Go back in time: Generating flashbacks in stories with event plots and temporal prompts. In *2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2022.

[He *et al.*, 2019] He He, Nanyun Peng, and Percy Liang. Pun generation with surprise. In *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019)*, volume 1, 2019.

[Hovy and Spruit, 2016] D. Hovy and S. L. Spruit. The social impact of natural language processing. In *Association for Computational Linguistics (ACL)*, 2016.

[Hsu *et al.*, 2022a] I-Hung Hsu, Xiao Guo, Premkumar Natarajan, and Nanyun Peng. Discourse-level relation extraction via graph pooling. In *The Thirty-Sixth AAAI Conference On Artificial Intelligence Workshop on Deep Learning on Graphs: Method and Applications (DLG-AAAI)*, 2022.

[Hsu *et al.*, 2022b] I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. Degree: A data-efficient generation-based event extraction model. In *2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2022.

[Huang *et al.*, 2019] Xiao Huang, Li Dong, Elizabeth Boschee, and Nanyun Peng. Learning a unified named entity tagger from multiple partially annotated corpora for efficient adaptation. In *The 2019 SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2019.

[Huang *et al.*, 2021] Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. Document-level entity-based extraction as template generation. In *The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[Justo *et al.*, 2014] Raquel Justo, Thomas Corcoran, Stephanie M Lukin, Marilyn Walker, and M Inés Torres. Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web. *Knowledge-Based Systems*, 69:124–133, 2014.

[Lewis *et al.*, 2020] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[Lu and Peng, 2021] Sidi Lu and Nanyun Peng. On efficient training, controllability and compositional generalization of insertion-based language generators. *arXiv preprint arXiv:2102.11008*, 2021.

[Martin *et al.*, 2017] Lara Martin, Prithviraj Ammanabrolu, William Hancock, Shruti Singh, Brent Harrison, and Mark Riedl. Event representations for automated story generation with deep neural nets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2017.

[Meehan, 1976] James Richard Meehan. The metanovel: writing stories by computer. Technical report, YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE, 1976.

[Mittal *et al.*, 2022] Anirudh Mittal, Yufei Tian, and Nanyun Peng. Ambipun: Generating humorous puns with ambiguous context. In *2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), short*, 2022.

[Peng *et al.*, 2017] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-Tau Yih. Cross-sentence n-ary relation extraction with graph LSTM. *Transaction of the Association for Computational Linguistics (TACL)*, 2017.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Shah *et al.*, 2020] D. S. Shah, H. A. Schwartz, and D. Hovy. Predictive biases in natural language processing models: A conceptual framework and overview. In *Association for Computational Linguistics (ACL)*, 2020.

[Sheng *et al.*, 2019] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. In *2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), short*, 2019.

[Sheng *et al.*, 2020] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. Towards controllable biases in language generation. In *the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)-Findings, long*, 2020.

[Sheng *et al.*, 2021] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. "nice try, kiddo": Ad hominems in dialogue systems. In *The 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics, 2021.

[Stein *et al.*, 2018] Elizabeth Stein, Robert A Pearlman, and Tyler P Tate. Using metaphors and figurative language to improve communication about patient suffering. *Journal of Pain and Symptom Management*, 56(6):e68–e69, 2018.

[Stowe *et al.*, 2021] Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan, and Iryna Gurevych. Metaphor generation with conceptual mappings. In *Proceedings of the Conference of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[Tian and Peng, 2022] Yufei Tian and Nanyun Peng. Zero-shot sonnet generation with discourse-level coherence and poetic features. In *2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2022.

[Yao *et al.*, 2019] Lili Yao, Nanyun Peng, Weischedel Ralph, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-And-Write: Towards better automatic storytelling. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.

[Yoshimura *et al.*, 2015] Eriko Yoshimura, Misako Imono, Seiji Tsuchiya, and Hirokazu Watabe. A simile recognition system using a commonsense sensory association method. *Procedia Computer Science*, 60:55–62, 2015.