AfriKI: Machine-in-the-Loop Afrikaans Poetry Generation

Imke van Heerden
Dept. of Comparative Literature
College of Social Sciences and Humanities
Koç University, Istanbul, Turkey
ivanheerden@ku.edu.tr

Anil Bas
Dept. of Computer Engineering
Faculty of Technology
Marmara University, Istanbul, Turkey
anil.bas@marmara.edu.tr

Abstract

This paper proposes a generative language model called AfriKI. Our approach is based on an LSTM architecture trained on a small corpus of contemporary fiction. With the aim of promoting human creativity, we use the model as an authoring tool to explore machine-in-the-loop Afrikaans poetry generation. To our knowledge, this is the first study to attempt creative text generation in Afrikaans.

1 Introduction

Afrikaans[1] is a language spoken largely in South Africa, Namibia, Botswana and Zimbabwe. Masakhane (∀ et al., 2020a,b) draws important attention to the current disproportion of NLP research and resources with respect to African languages. In fact, in the entire ACL Anthology,[2] of the thirteen studies that mention "Afrikaans" in their titles, only four (Sanby et al., 2016; Augustinus et al., 2016; Dirix et al., 2017; Ralethe, 2020) appeared in the last five years. By no means do we ignore studies with inclusive (Eiselen and Puttkammer, 2014) and multilingual approaches (Ziering and Van der Plas, 2016) or those published via other platforms (Van Zaanen and Van Huyssteen, 2003). This is simply an indication that NLP research in Afrikaans is limited, especially in comparison to resource-rich languages, i.e. the so-called "winners" in the taxonomy of Joshi et al. (2020).

In this paper, we present a generative language model called AfriKI, an abbreviation for "Afrikaanse Kunsmatige Intelligensie" (*Afrikaans Artificial Intelligence*). We use this model as an authoring tool to explore machine-in-the-loop poetry generation in Afrikaans. Machine-in-the-loop frameworks promote human creativity through computational assistance, as opposed to human-in-the-loop pipelines, which aim to strengthen machine learning models (Clark et al., 2018). We treat poetry generation as a hybrid system, an experimental approach that enables the generation of high-quality poetic text with very limited data. To our knowledge, this is the first study in creative text generation as well as an initial step towards automatic poetry generation in Afrikaans.

Whereas NLG in its quest for full automation may frown upon human involvement, our human-centred framework does the opposite. According to Lubart (2005),

> one criticism of artificial intelligence programs that claim to be creative is exactly that a human plays a role at some point, which reduces the autonomy of the machine. From the HCI perspective [...] these "failed" AI creativity programs are examples of successful human–computer interactions to facilitate creativity.

This study demonstrates that human-machine collaboration could enhance human creativity. We agree with Shneiderman (2002) that support tools "make more people more creative more often".

2 Related Work

Several computational models focus on automatic poetry generation. First approaches follow rule-based, template-based systems (Gervás, 2001; Díaz-Agudo et al., 2002). Levy (2001) and Manurung et al. (2012) apply genetic algorithms while Jiang and Zhou (2008) and He et al. (2012) use statistical machine translation, with Yan et al. (2013) utilising text summarisation to generate poetry.

[1] The Constitution of the Republic of South Africa recognises Afrikaans as one of eleven official languages, alongside Sepedi, Sesotho, Setswana, siSwati, Tshivenda, Xitsonga, English, isiNdebele, isiXhosa and isiZulu (Assembly, 1996). In South Africa, there are approximately 6.9 million first-language speakers of Afrikaans, according to the most recent census (Lehohla, 2012).

[2] https://www.aclweb.org/anthology/

Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing, pages 74–80
April 20, 2021.

Oliveira (2009) provides a clear overview of early systems and presents a comparable method (2012).

Starting with Zhang and Lapata (2014), we have seen great advancements in poetry generation using neural networks. Wang et al. (2016a) extend this using the attention mechanism (Bahdanau et al., 2015). There are many attempts to improve the quality of learning-based generated poetry, by using planning models (Wang et al., 2016b), finite-state machinery (Ghazvininejad et al., 2016), reinforcement learning (Yi et al., 2018) as well as variational autoencoders (Yang et al., 2018).

Conventional recurrent neural networks (RNN) are not suitable for learning long range dependencies (Wang et al., 2016a) due to the vanishing gradient problem (Bengio et al., 1994). Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) address this issue and are widely used for language modeling (Sundermeyer et al., 2012). Tikhonov and Yamshchikov (2018) propose word-based LSTM to generate poetry. Potash et al. (2015) adopt a similar technique to produce rap lyrics. Zugarini et al. (2019) apply syllable-based LSTM to generate tercets. Finally, composed of various LSTM models, Deep-speare (Lau et al., 2018) generates Shakespearean sonnets.

The remarkable quality and results of these studies are indisputable. However, they all concentrate on data-rich languages such as English, Chinese, Italian and Russian. For example, the character language model of Hopkins and Kiela (2017) uses a poetry corpus consisting of 7.56 million words and 34.34 million characters. Likewise, a recent study by Liu et al. (2020) trained on over 200 thousand poems and 3 million ancient Chinese prose texts.

We trained an LSTM network for poetic text generation as well. However, our approach differs in significant ways. First, whereas these studies generate verse in a fully automatic manner, we emphasise human creativity, introducing a strong computational component to the creative writing process. Second, the aforementioned studies either trained on comprehensive poetry datasets or model poetic qualities. To illustrate the latter, the recent work of Van de Cruys (2020) focuses on specifically non-poetic text in English and French, however, is able to model the rhyme constraint using phonetic representation of words from Wiktionary. Since there is no publicly available large-scale poetry dataset in Afrikaans, we follow an alternative approach, constructing our model as a text generator that produces individual sentences and phrases instead of stanzas of verse. In other words, the model outputs a set of lines, which we arrange vertically into short poems without modification.

Figure 1: Frequently occurring words in *Die Biblioteek aan die Einde van die Wêreld*. Stop words were removed. Note that Ian and Thuli are the protagonists.

3 Model

In this section, we explain the dataset, model architecture as well as the co-creative poetry generation process.

Corpus: AfriKI trained on a lengthy (208,616-word) literary novel titled *Die Biblioteek aan die Einde van die Wêreld* (*The Library at the End of the World*) (Van Heerden, 2019) by the South African novelist Etienne van Heerden. In 2020, the book was awarded the University of Johannesburg Prize for Literature (Pienaar, 2020). This work of new journalism combines fictional techniques with documentary language, and is particularly suitable given its use of rich imagery, figurative language as well as different Afrikaans varieties like *Kaaps* (or Cape Afrikaans) and Standard Afrikaans. Figure 1 shows a word cloud of its most commonly used words.

Model Architecture: Experimenting with several architectures, including LSTM, Multi-Layer LSTM and Bi-LSTM, we obtain best results with the following two-layer LSTM architecture. We use a vanilla LSTM structure (Hochreiter and Schmidhuber, 1997) and, to avoid repetitiveness, omit to describe the network diagram and equations, similar to Sundermeyer et al. (2012). We start with 100-dimensional word embeddings with a vocabulary size of 23,317 words, where weights are randomly initialised from a normal distribution with zero mean and standard deviation 0.01. Next, we stack two LSTM layers with 50 units in each layer followed by dropout layers with the

Original (Afrikaans)	Translation (English)
Die konstabel se skiereiland	*The constable's peninsula*
Afrika drink onheil in die water. Die landskap kantel sy rug in sigbewaking en vlam. Ons oopgesnyde sake brandtrappe vir die ander state. Hierdie grond word intimidasie.	Africa drinks disaster in the water. The landscape tilts its back in surveillance and flame. Our cut-open affairs fire escapes for other states. This soil becomes intimidation.
Gedigte, daar by die brul van 'n brander	*Poetry, there near the roar of a wave*
Hier is die oë katvoet vir die spoelrotse onder uitdrukkings die golwe van gister wat getol en woes en water saam met die son skuim in hul woorde	Here the eyes are cautious of the sea rocks under expressions the waves of yesterday that whirled and wild and water froth with the sun in their words
die ingedagte see lig die geure en praat 'n asemhaal	the introspective sea lifts the scents and utters a breath
Kaapstad	*Cape Town*
Vandag is ons nie net die stad nie maar die vertaler van die son	Today we are not just the city but the translator of the sun
Vanaand se gordyne glinster by skuifvensters in die stadsliggies	Tonight's curtains glitter at sliding windows in the city lights
Die uur van die winde sorg dat dit rondom klink Sy wil die glasvensters deurkosyn eens iets te beskerm	The hour of the winds takes care it sounds around She wants to doorframe the glass windows to protect something
Tafelberg maak 'n vraag waarbinne ons 'n duisend name genoem word	Table Mountain creates a question in which we are given a thousand names

Table 1: Example results of machine-in-the-loop poetry generation.

rate of 0.2. This is followed by a fully connected layer and a softmax layer. We use the Adam optimiser (Kingma and Ba, 2015) with a learning rate = 0.001, batch size = 16, and train for 300 epochs. Although tweaking the parameters did change the model performance, it was not significant.

Machine-in-the-Loop: Human-machine collaboration for the enhancement of creative writing has been examined under automated assistance (Roemmele and Gordon, 2015, 2018), co-authorship (Tucker, 2019), co-creativity (Manjavacas et al., 2017; Kantosalo and Riihiaho, 2019; Calderwood et al., 2020), interactive storytelling (Swanson and Gordon, 2012; Brahman et al., 2020) and machine-in-the-loop (Clark et al., 2018; Akoury et al., 2020).

Applying Clark et al. (2018)'s terminology, we employ an iterative interaction structure that follows a push method of initiation with low intrusiveness. To clarify, our process consists of a single loop with two stages. First, the model generates a sizable set of unique individual lines (hundreds). Although memory networks may repeat parts of the training data (Ghazvininejad et al., 2016), the generated phrases are highly distinct from the dataset, with hardly any repetition of word order. Second, the first author responds by choosing phrases at will. To create the final artefact, the author arranges the selected lines vertically. Generated text is used strictly without modification (except for some capitalisation and punctuation). The result of our collaborative writing system is short, compelling works of poetry that draw inspiration from the literary movements Imagism (Hughes, 1972) and Surrealism (Balakian, 1986).

4 Results

Table 1 presents three examples of poems produced by means of the co-creative process. Here, we discuss quality from a literary perspective.

Trained on prose, the text is generated as free verse (i.e. free from the restrictions of rhythm and rhyme) which we associate with contemporary poetry. In the lines, various poetic devices can be identified, such as alliteration (e.g. "**g**olwe van **g**ister") and assonance (e.g. "m**aa**k 'n vr**aa**g w**aa**rbinne").

The generated lines abound with figurative language as well. As an instance of an extended metaphor, the first stanza of the second poem suggests sensitivity to the country's turbulent history. Personification is particularly prevalent, lending a visceral quality to the text: Africa drinks, the landscape tilts its back, the sea breathes, and Table Mountain poses a question. The imagery is vivid, portraying sight (*Tonight's curtains / glitter at sliding windows / in the city lights*), smell (*the introspective sea / lifts the scents and utters / a breath*) and sound (*roar of a wave*). The language can be described as minimalist, evocative and abstract, and therefore open to interpretation, resembling Imagist and Surrealist poetry.

Afrikaans has a rich poetic tradition (Brink and Opperman, 2000), and we believe that creative text generation has the potential to enrich poetic language. Alongside Afrikaans varieties, the corpus contains some English as well, which influenced the generated text in interesting ways. As one example, it is grammatically incorrect in Standard Afrikaans to use "sun" as both noun and verb, e.g. "to sun in the garden". The model, however, adopted this and other patterns from the English, generating novel phrases (that do not sound anglicised) such as "sonlig son die promenade" – *sunlight suns the promenade*.

5 Conclusion

In this study, we present Afrikaans poetry generation in a machine-in-the-loop setting. Each and every line of poetry is automatically generated by the proposed LSTM network. In order to clearly identify the machine's contribution to the process, the human writer's interaction is limited to the selection and vertical arrangement of the lines – without any modification. We believe this is the first creative text generation study in the Afrikaans language. More broadly, the work encourages human-centred design in low-resource languages. Creative industries would benefit from co-creative tools and methods (Hsu et al., 2019), perhaps more than fully automatic approaches.

6 Future Work

There are many ways in which this work can be extended.

First, similar to Yi et al. (2017), we could follow line-to-line poem generation, where the network takes the previous line as prompt and generates a new line which, in turn, is the prompt for the next entry. We could also experiment with different architectures, such as Transformer (Vaswani et al., 2017), as well as training schemes. For example, we could borrow AfriBERT (Ralethe, 2020), the recent BERT (Devlin et al., 2019) adaptation for Afrikaans, to apply transfer learning.

Second, as demonstrated in Van de Cruys (2020), poetry generation is also possible by training on prosaic (non-poetic) text and modeling poetic constraints (e.g. rhyme). This way, we could expand to fully automatic poetry generation. Naturally, this would require an extensive literature corpus.

Third, regarding the unconventional use of some nouns as verbs in Afrikaans, future research could explore how prevalent this type of novel, cross-language variation is. To improve textual quality, we could incorporate Afrikaans datasets such as the NCHLT Annotated Text Corpora (Eiselen and Puttkammer, 2014; Puttkammer et al., 2014) as well as the Afrikaans treebank (Augustinus et al., 2016), which are available via SADiLaR (Roux, 2016) in addition to others.

Finally, a promising direction to pursue would be the involvement of poets and writers to investigate whether this approach could inform and improve their creative writing practices.

Acknowledgments

This paper has been produced benefiting from the 2232 International Fellowship for Outstanding Researchers Program of TÜBİTAK (Project No: 118C285). However, the entire responsibility of the paper belongs to the owner of the paper. The financial support received from TÜBİTAK does not mean that the content of the publication is approved in a scientific sense by TÜBİTAK.

We would like to thank Etienne van Heerden for providing his manuscript to be used in this study.

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A dataset and platform for human-in-the-loop story generation. In *Proc EMNLP*, pages 6470–6484.

Constitutional Assembly. 1996. Constitution of the Republic of South Africa. *Cape Town*, 230(38):1241–1331.

Liesbeth Augustinus, Peter Dirix, Daniel Van Niekerk, Ineke Schuurman, Vincent Vandeghinste, Frank Van Eynde, and Gerhard Van Huyssteen. 2016. AfriBooms: An online treebank for Afrikaans. In *Proc LREC*, pages 677–682.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc ICLR*.

Anna Balakian. 1986. *Surrealism: The Road to the Absolute*. University of Chicago Press.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Faeze Brahman, Alexandru Petrusca, and Snigdha Chaturvedi. 2020. Cue me in: Content-inducing approaches to interactive story generation. In *Proc AACL-IJCNLP*, pages 588–597.

André Philippus Brink and Diederik Johannes Opperman. 2000. *Groot verseboek 2000*. Tafelberg.

Alex Calderwood, Vivian Qiu, Katy Ilonka Gero, and Lydia B Chilton. 2020. How novelists use generative language models: An exploratory user study. In *Proc ACM IUI Workshop*.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proc ACM IUI*, pages 329–340.

Tim Van de Cruys. 2020. Automatic poetry generation from prosaic text. In *Proc ACL*, pages 2471–2480.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc NAACL*, pages 4171–4186.

Belén Díaz-Agudo, Pablo Gervás, and Pedro A González-Calero. 2002. Poetry generation in COLIBRI. In *Proc ECCBR*, pages 73–87.

Peter Dirix, Liesbeth Augustinus, Daniel Van Niekerk, and Frank Van Eynde. 2017. Universal dependencies for Afrikaans. In *Proc NoDaLiDa*, pages 38–47.

Roald Eiselen and Martin Puttkammer. 2014. Developing text resources for ten South African languages. In *Proc LREC*, pages 3698–3703.

∀, Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogueji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkadir Dangana, Herman Kamper, Hady Elsahar, Goodness Duru, Ghollah Kioko, Murhabazi Espoir, Elan van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Bassey, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020a. Participatory research for low-resourced machine translation: A case study in African languages. In *Proc EMNLP*, pages 2144–2160.

∀, Iroro Orife, Julia Kreutzer, Blessing Sibanda, Daniel Whitenack, Kathleen Siminyu, Laura Martinus, Jamiil Toure Ali, Jade Abbott, Vukosi Marivate, Salomon Kabongo, Musie Meressa, Espoir Murhabazi, Orevaoghene Ahia, Elan van Biljon, Arshath Ramkilowan, Adewale Akinfaderin, Alp Öktem, Wole Akin, Ghollah Kioko, Kevin Degila, Herman Kamper, Bonaventure Dossou, Chris Emezue, Kelechi Ogueji, and Abdallah Bashir. 2020b. Masakhane–machine translation for Africa. In *Proc ICLR Workshop*.

Pablo Gervás. 2001. An expert system for the composition of formal Spanish poetry. In *Proc SGES*, pages 19–32.

Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proc EMNLP*, pages 1183–1191.

Jing He, Ming Zhou, and Long Jiang. 2012. Generating Chinese classical poems with statistical machine translation models. In *Proc AAAI*, volume 26.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jack Hopkins and Douwe Kiela. 2017. Automatically generating rhythmic verse with neural networks. In *Proc ACL*, pages 168–178.

Ting-Yao Hsu, Yen-Chia Hsu, and Ting-Hao Huang. 2019. On how users edit computer-generated visual stories. In *Proc ACM CHI*, pages 1–6.

Glenn Hughes. 1972. *Imagism & the Imagists: A Study in Modern Poetry*. Biblo & Tannen Publishers.

Long Jiang and Ming Zhou. 2008. Generating Chinese couplets using a statistical MT approach. In *Proc COLING*, pages 377–384.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proc ACL*, pages 6282–6293.

Anna Kantosalo and Sirpa Riihiaho. 2019. Experience evaluations for human–computer co-creative processes–planning and conducting an evaluation in practice. *Connection Science*, 31(1):60–81.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proc ACL*, pages 1948–1958.

Pali Lehohla. 2012. *Census in brief 2011*. Statistics South Africa.

Robert P Levy. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proc ICCBR Workshop*.

Yusen Liu, Dayiheng Liu, and Jiancheng Lv. 2020. Deep poetry: A Chinese classical poetry generation system. In *Proc AAAI*, volume 34(09), pages 13626–13627.

Todd Lubart. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies*, 63(4-5):365–369.

Enrique Manjavacas, Folgert Karsdorp, Ben Burtenshaw, and Mike Kestemont. 2017. Synthetic literature: Writing science fiction in a co-creative process. In *Proc CCNLG*, pages 29–37.

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

Hugo Oliveira. 2009. Automatic generation of poetry: an overview. *Universidade de Coimbra*.

Hugo Gonçalo Oliveira. 2012. PoeTryMe: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence*, 1:21.

Hans Pienaar. 2020. *Prize-winning novel on #FeesMustFall is a campus of ideas*. Business Day. https://www.businesslive.co.za/bd/life/books/2020-04-09-book-review-prize-winning-novel-on-feesmustfall-is-a-campus-of-ideas/.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. Ghostwriter: Using an LSTM for automatic rap lyric generation. In *Proc EMNLP*, pages 1919–1924.

Martin Puttkammer, Martin Schlemmer, and Ruan Bekker. 2014. *NCHLT Afrikaans Annotated Text Corpora*. South African Centre for Digital Language Resource. https://hdl.handle.net/20.500.12185/296.

Sello Ralethe. 2020. Adaptation of deep bidirectional transformers for Afrikaans language. In *Proc LREC*, pages 2475–2478.

Melissa Roemmele and Andrew Gordon. 2018. Linguistic features of helpfulness in automated support for creative writing. In *Proc ICIDS*, pages 14–19.

Melissa Roemmele and Andrew S Gordon. 2015. Creative help: A story writing assistant. In *Proc ICIDS*, pages 81–92.

Justus Roux. 2016. South African National Centre for Digital Language Resources. In *Proc LREC*, pages 2467–2470.

Lauren Sanby, Ion Todd, and Maria C Keet. 2016. Comparing the template-based approach to GF: the case of Afrikaans. In *Proc WebNLG*, pages 50–53.

Ben Shneiderman. 2002. Creativity support tools. *Communications of the ACM*, 45(10):116–120.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Proc INTERSPEECH*.

Reid Swanson and Andrew S Gordon. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Interactive Intelligent Systems*, 2(3):1–35.

Aleksey Tikhonov and Ivan Yamshchikov. 2018. Sounds Wilde: Phonetically extended embeddings for author-stylized poetry generation. In *Proc SIGMORPHON*, pages 117–124.

Aaron Tucker. 2019. Machine co-authorship (s) via translative creative writing. *Journal of Creative Writing Studies*, 4(1):7.

Etienne Van Heerden. 2019. *Die Biblioteek aan die Einde van die Wêreld*. NB-Uitgewers.

Menno Van Zaanen and Gerhard Van Huyssteen. 2003. Improving a spelling checker for Afrikaans. In *Proc CLIN*, pages 143–156.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc NIPS*, pages 6000–6010.

Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. 2016a. Chinese song iambics generation with neural attention-based model. In *Proc IJCAI*, pages 2943–2949.

Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016b. Chinese poetry generation with planning based neural network. In *Proc COLING*, pages 1051–1060.

Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li. 2013. I, poet: automatic Chinese poetry composition through a generative summarization framework under constrained optimization. In *Proc IJCAI*.

Xiaopeng Yang, Xiaowen Lin, Shunda Suo, and Ming Li. 2018. Generating thematic Chinese poetry using conditional variational autoencoders with hybrid decoders. In *Proc IJCAI*, pages 4539–4545.

Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating Chinese classical poems with RNN encoder-decoder. In *Proc NLP-NABD*, pages 211–223.

Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic poetry generation with mutual reinforcement learning. In *Proc EMNLP*, pages 3143–3153.

Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proc EMNLP*, pages 670–680.

Patrick Ziering and Lonneke Van der Plas. 2016. Towards unsupervised and language-independent compound splitting using inflectional morphological transformations. In *Proc NAACL*, pages 644–653.

Andrea Zugarini, Stefano Melacci, and Marco Maggini. 2019. Neural poetry: Learning to generate poems using syllables. In *Proc ICANN*, pages 313–325.